

Z4P

Presentacions àgils des d'un navegador

Jofre Costa i Delgado

2n de Batxillerat A

Tutor: Joan Carol

Institut Pere Alsius i Torrent (Banyoles)

8 d'octubre de 2018

Agraïments

Vull donar gràcies a la meua família per la seva paciència, comprensió i suport incondicional en tot moment.

També vull agrair la col·laboració dels companys, amics i professors que han donat un cop d'ull al projecte i han compartit la seva valoració i opinió del programa final.

Gràcies també als professors Antonio Bueno, Teodor Jové i Pere Vilà de la Universitat de Girona per la seva ajuda en la tria del treball i per la seva valoració experta del projecte Z4P.

Índex

| | |
|---|----|
| 1. <u>Introducció</u> | 5 |
| 2. <u>Part teòrica</u> | 7 |
| 2.1. Què és la programació? | 7 |
| 2.2. Què és un llenguatge de programació? | 8 |
| 2.2.1. Elements sintàctics bàsic d'un llenguatge de programació ... | 10 |
| 2.2.2. Biblioteques | 13 |
| 2.2.3. Maneres de classificar els llenguatges de programació | 14 |
| 2.2.3.1. Segons el nivell d'abstracció | 15 |
| 2.2.3.2. Segons el mètode d'execució | 16 |
| 2.2.3.3. Segons el seu propòsit..... | 16 |
| 2.2.3.4. Segons el paradigma de programació | 17 |
| 2.3. La web i el seu funcionament | 18 |
| 2.4. Què és el desenvolupament web?..... | 21 |
| 2.4.1. Pàgines estàtiques i pàgines dinàmiques | 22 |
| 2.4.2. Desenvolupament de la part frontal d'una pàgina web..... | 23 |
| 2.4.2.1. Programació de la part frontal | 24 |
| 2.4.2.2. Accessibilitat i disseny responsiu..... | 26 |
| 2.4.2.3. SEO | 27 |
| 2.4.3. Desenvolupament de la part posterior d'una pàgina web | 28 |
| 3. <u>Part pràctica</u> | 30 |
| 3.1. Introducció al projecte Z4P | 30 |
| 3.2. Aprenentatge dels llenguatges necessaris..... | 31 |
| 3.3. Anàlisi de programes i aplicacions semblants..... | 32 |
| 3.3.1. S5 Slide Show | 33 |
| 3.3.2. S9 Slide Show i altres..... | 33 |
| 3.3.3. Microsoft PowerPoint..... | 33 |

| | | |
|--------|--|----|
| 3.3.4. | Google Slides..... | 34 |
| 3.3.5. | Prezi | 34 |
| 3.4. | Disseny i esbós inicial de les interfícies d'usuari | 35 |
| 3.5. | Funcionalitat inicial i llistat de característiques del projecte | 36 |
| 3.6. | Esriptura de l'HTML de Z4P..... | 37 |
| 3.7. | Esriptura del CSS de Z4P | 39 |
| 3.8. | Esriptura del JavaScript de Z4P..... | 42 |
| 3.9. | Llicències i drets d'autor | 47 |
| 3.10. | Elaboració d'un tutorial i d'una llista de característiques..... | 48 |
| 3.11. | Període de correcció d'errors i valoracions..... | 49 |
| 4. | <u>Conclusions</u> | 50 |
| 5. | <u>Bibliografia</u> | 52 |
| 6. | <u>Annexos</u> | 55 |
| 6.1. | Llicències i crèdits de Z4P | 55 |
| 6.2. | Tutorial de Z4P | 60 |
| 6.3. | Llistat de caraterístiques de Z4P | 64 |
| 6.4. | Codi de Z4P | 68 |

Índex d'il·lustracions

| | |
|--|----|
| Figura 1. Representació gràfica del funcionament de les variables..... | 10 |
| Figura 2. Diagrama del funcionament bàsic d'un bucle. | 11 |
| Figura 3. Diagrama del funcionament d'un condicional "if-then-else" | 12 |
| Figura 4. Representació gràfica del funcionament d'una funció | 13 |
| Figura 5. Representació gràfica del funcionament d'una funció.. | 21 |
| Figura 6. Funcionament de les pàgines dinàmiques | 23 |
| Figura 7. Funcionament de les pàgines dinàmiques | 23 |
| Figura 8. Comparació entre el funcionament d'Ajax i el tradicional | 26 |
| Figura 9. Esquema de l'interfície d'usuari en el mode normal. | 35 |
| Figura 10. Esquema de l'interfície d'usuari en el mode presentació..... | 36 |

1. Introducció

Fa uns pocs anys no tenia molt clar què volia ser quan fos gran ni quina carrera volia estudiar: al principi tirava més cap a les ciències socials, però també m'agradaven molts aspectes de les ciències i la tecnologia... I mai m'acabava de decidir. A l'últim any de l'ESO però, em vaig començar a interessar per la informàtica gràcies a una assignatura que cursàvem a l'institut i que em fascinava. Des de llavors m'he decantat cap a l'àmbit tecnològic, i he tingut força clar que vull estudiar la carrera d'enginyeria informàtica, o si més no alguna altra de semblant que involucri de manera abundant algun aspecte de la informàtica.

Per aquesta raó he decidit enfocar el meu treball de recerca en la informàtica, per tal de començar a adquirir coneixements per a la universitat i poder estar més preparat pel que pugui venir. Com que la informàtica és un tema molt ampli que inclou molts apartats diferents, he decidit centrar el treball sobretot en l'àmbit de la programació i la web.

Per què la programació? Bàsicament perquè et permet crear el programa que més t'interessi i que més s'adapti a les teves necessitats. Amb un llenguatge suficientment potent i amb el coneixement necessari, qualsevol programa que et puguis imaginar es pots crear. A més, la programació constitueix la base de molts altres àmbits de la informàtica, per no dir quasi tots, i per tant és un bon punt de partida per començar a endinsar-me en l'extens món de la informàtica. I per què la web? Doncs perquè és un camp de la informàtica molt habitual en el dia a dia de les persones, i no és exclusiu de les persones que hi entenen, sinó que qualsevol persona hi té accés i n'és un participant, per pocs coneixements que en tingui. A més, hi ha un àmbit de la web que em permet combinar els dos temes i iniciar-me en la programació i escriptura de codi: el desenvolupament web.

Els objectius principals del treball en són tres:

- Adquirir tots els coneixements possibles d'informàtica per ajudar-me en la futura carrera.

- Assolir un nivell elevat de programació amb tres llenguatges de desenvolupament de pàgines i aplicacions web com són HTML, CSS i JavaScript.
- Crear un programa complex que resolgui la necessitat de realitzar amb rapidesa presentacions de diapositives simples, però que a la vegada siguin útils per a qualsevol situació possible. Per fer-ho, s'hauran d'aplicar tots els coneixements adquirits durant el transcurs del treball.

El treball consta de dues parts principals: una part teòrica i una part pràctica. La part teòrica està centrada en aspectes teòrics relacionats amb la programació, la web i el desenvolupament web. La part pràctica es basa en la descripció del procés que s'ha utilitzat per crear el programa. La metodologia que he seguit en l'elaboració del treball comença amb una visita a la Universitat de Girona, on vaig parlar amb experts en diferents àmbits de la informàtica que em van ajudar moltíssim a triar el tema i a encarar el treball. Un cop escollit el tema, per a l'elaboració de la part teòrica m'he basat en la recerca d'informació en pàgines web i també en llibres relacionats amb el tema. L'elaboració de la part pràctica ha sigut més laboriosa, ja que he hagut d'aprendre els tres llenguatges mencionats anteriorment des de zero mitjançant guies extenses i completament en anglès. Per a l'escriptura del codi del programa he utilitzat un ordinador i un entorn de programació molt útil anomenat Visual Studio Code, on hi he escrit i editat tot el codi i m'ha ensenyat a entendre millor totes les accions. Tot i haver estudiat teòricament els llenguatges amb anterioritat, posar-los en pràctica no és tan fàcil, i en alguns moments he hagut d'utilitzar el mètode de prova i error per progressar en el codi del programa. Finalment, he descrit en un document de text el procés que he seguit en l'aprenentatge i l'escriptura del codi, i aquest document és el que constitueix la part pràctica.

2. Part teòrica

2.1. Què és la programació?

La programació es defineix com l'acció o efecte de programar, és a dir, establir i enunciar la seqüència de passos que es volen seguir per fer una determinada cosa. La paraula programació pot aparèixer en molts contextos diferents: en la televisió o la ràdio sobre la planificació prevista per l'ordre dels programes, en l'educació com el pla d'estudis que elabora un professor per a tot el curs... Però on probablement s'usa més és a l'àmbit de la informàtica.

La programació informàtica es defineix com l'elaboració de programes informàtics o aplicacions mitjançant el disseny, codificació i desenvolupament del seu codi font. Aquest codi font està escrit en un determinat llenguatge de programació i conté les instruccions que ha de seguir l'ordinador per executar un programa concret. De fet, l'objectiu de la programació és la creació de programes, mitjançant el desenvolupament i edició del seu codi font, que es comportin tal com el creador vol.

Tot i que per crear un programa només és necessari tenir una idea en ment i saber com es programa, per dissenyar i desenvolupar un programa o aplicació molt complexos es requereix de l'enginyeria de software. Segons l'IEEE (Institute of Electrical and Electronics Engineers), l'enginyeria de software és l'aplicació i estudi d'un enfocament sistemàtic, disciplinat i quantificable al desenvolupament, operació i manteniment del software o programari¹. Aquesta àrea de la informàtica segueix una determinada metodologia que va des de l'inici del projecte fins a l'aplicació o programa final, passant per l'anàlisi dels requisits, l'arquitectura del software, tests continus per trobar errors, manteniment... Entre tots els passos que se segueixen un d'ells és la programació, que únicament es basa en la implementació i desenvolupament del codi font del programa, i que en molts casos no acostuma a ser la part més costosa ni la més complicada de tot el procés.

Els coneixements necessaris per poder programar són relatius i depenen sobretot del llenguatge de programació que s'utilitzi. A vegades fins i tot no fa

1: IEEE-SA, 1993. IEEE Standard Glossary of Software Engineering Terminology. Nova York, 1993. Pàg. 67.

falta cap tipus de coneixement previ per poder programar, com és el cas dels llenguatges basats en blocs, tot i que en la gran majoria dels casos és essencial dominar a fons el llenguatge que s'utilitza. També en molts casos s'utilitzen processos i algorismes que ajuden a millorar i optimitzar el codi que s'escriu. Amb els coneixements suficients la programació es converteix en una eina amb un poder gairebé il·limitat: et permet crear qualsevol programa que puguis imaginar, amb la complexitat i totes les funcions que es vulguin i siguin necessàries.

2.2. Què és un llenguatge de programació?

Ara que ja se sap què és la programació, toca esbrinar què és un llenguatge de programació (tot i que ja es pot deduir una mica de l'apartat anterior). S'entén com a llenguatge un sistema de comunicació que serveix per expressar pensaments i sentiments a un receptor, amb el qual es comparteix el coneixement del codi lingüístic. Sabent això, ara ja podem definir un llenguatge de programació.

Un llenguatge de programació és aquell llenguatge o sistema formal, el qual presenta una estructura concreta i específica, que transmet una sèrie d'instruccions a una computadora per tal d'aconseguir controlar el seu comportament.

Generalment, els llenguatges de programació, de manera força semblant als llenguatges o llengües que tots coneixem, es divideixen en dues parts: la sintaxi i la semàntica. La sintaxi és la part que veiem del llenguatge, el codi en sí. La gran majoria dels llenguatges són textuals i contenen paraules, números i puntuació a l'igual que qualsevol altra llengua, però n'hi ha alguns que es basen en entorns més gràfics i visuals per establir relacions entre els diferents elements. La sintaxi descriu i especifica de quina manera s'han de combinar els diferents símbols o elements per obtenir línies o blocs correctament estructurats, i d'aquesta manera poder-se comunicar amb la computadora o ordinador. Si no es combinen correctament, l'ordinador no entén les ordres i com a conseqüència el programa no funcionarà correctament.

D'altra banda, la semàntica és la part no visible del llenguatge, i equival al significat del codi. I què és el significat del codi? Doncs es tracta del procés que segueix un ordinador quan executa un programa, és a dir, l'explicació del que fa el programa i del que s'aconsegueix quan aquest s'executa. Segons l'expert Jérôme Euzenat, la semàntica proporciona les normes per interpretar la sintaxi, la qual no ens dona el significat directament però limita les possibles interpretacions del que s'executa². Per veure més clar què són tant la sintaxi com la semàntica va bé afegir un exemple:

Sintaxi:

x = "Hola, " + "bon dia!"

Semàntica:

Se suma "Hola, " i "bon dia!" i s'emmagatzema a la variable x el valor "Hola, bon dia!".

En aquest exemple pot semblar que són molt semblants ja que es tracta d'un exemple molt simple, però es pot apreciar que en el cas de la sintaxi només hi ha codi escrit, i en canvi a la semàntica apareix el que succeeix a mesura que es llegeix i s'entén el codi. També podria passar que la sintaxi fos teòricament correcta però que la semàntica donés algun tipus d'error. És el cas de la frase "la patata escriu vídeos al mar": sintàcticament està bé i no té cap problema, però la frase no té ni cap ni peus i no queda gens clar el que vol dir. El mateix passa amb un llenguatge de programació, si la sintaxi està ben estructurada però hi ha alguna combinació amnigua o sense sentit entre elements, la semàntica falla.

A les dues parts del codi ja esmentades, la sintaxi i semàntica, hi ha qui hi afegeix una tercera part: els sistemes de tipus. Aquests el que fan és donar informació extra a un valor, a una variable o a un altre tipus de construcció, de manera que es pot saber de quin tipus es tracta i així estalviar temps i errors a l'hora de compilar o interpretar el programa. Per exemple, els valors d'una variable poden ser números, text, una llista de valors... I segons quin sigui se li assigna un determinat tipus que el compilador o intèrpret del llenguatge serà capaç d'identificar. Els sistemes de tipus també defineixen com es relacionen i

interactuen els diferents tipus, cosa molt rellevant quan es volen dur a terme operacions. Tots els llenguatges moderns presenten un sistema de tipus propi, però segons el llenguatge aquest actua de manera més flexible (permetent operacions entre diferents tipus) o de manera més rígida (prohibint o fent saltar un error en el procés de compilació quan es produeix una operació entre tipus no compatibles).

2.2.1. Elements sintàctics bàsic d'un llenguatge de programació

Una altra cosa molt important són els elements sintàctics comuns en molts llenguatges. Si ens fixem en tots els llenguatges moderns i que més s'utilitzen actualment, podem observar que en tots o en pràcticament tots apareixen cinc elements comuns: les variables, els bucles, els condicionals, les subrutines i funcions i per últim les "entrades i sortides".

Les **variables** són elements que tenen un nom o identificador associat i que contenen una determinada informació que es coneix com a valor, per tant es podria dir que la seva funció principal és l'emmagatzematge de dades o informació, ja sigui coneguda o no. El valor que poden contenir poden ser tant números, com text, com un llistat de valors de diferents tipus (arrays)... Per a què una variable sigui vàlida primer s'ha de declarar, que no és més que assignar-li un nom i un valor inicial. El que és molt important és que el valor que contenen no és rígid i invariable, sinó que pot canviar si se li assigna un nou valor. D'aquesta manera, pot ser que la variable tingui valors diferents en diferents parts del codi.



Figura 1. Representació gràfica del funcionament de les variables. Font: [w3resource](http://w3resource.com).

Seguidament ens trobem els **bucles**, que són una sèrie d'instruccions que són repetides constantment fins que una certa condició es deixa de complir (tot i que pot ser que la condició sempre es compleixi i llavors el bucle es repeteixi indefinidament). Els bucles poden tenir diferents formes, però la gran majoria contenen com a mínim una declaració de variable, una condició, un cos principal on inserir-hi tot el codi o instruccions i, finalment, l'assignació d'un nou valor a la variable inicial. El seu funcionament és força simple, i s'entén millor amb un petit exemple. Primer declarem una variable "i" i li assignem un valor de 1 ($i = 1$). Després afegim una condició que només permeti que el bucle funcioni quan "i" sigui més petita que 7 ($i < 7$). Tot seguit afegim unes instruccions dins del cos que facin que ens mengem una poma, i finalment fem que a la variable "i" se li sumi una unitat ($i = i + 1$). Amb això el que obtenim és quan $i=1$ ens mengem una poma, i llavors li sumem 1 a "i". Llavors es torna a l'inici i es comprova si la i segueix sent més petita que 7, i com que ho és ($2 < 7$) ens tornem a menjar una poma i tornem a sumar una unitat a "i". Aquest procés segueix fins que la condició no es compleix, és a dir, quan $i=7$.

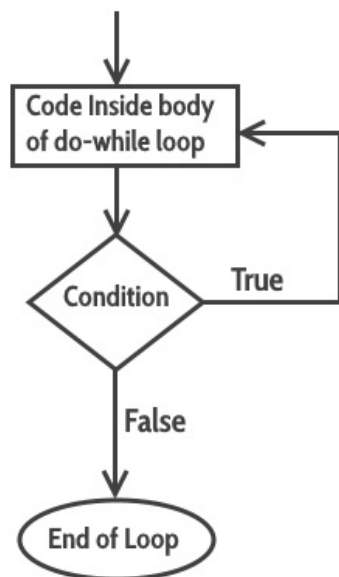


Figura 2. Diagrama del funcionament bàsic d'un bucle.
Font: [BeginnersBook](#)

Tot seguit hi ha els **condicionals**, que es tracta d'unes construccions que executen un bloc de codi o un altre dependent de si es compleix o no una condició. La seva estructura bàsica és la de "if-then-else", que conté dos blocs de codi diferents i una condició. Si la condició es compleix i per tant és certa, s'executa el codi de la primera branca ("if"). Per contra, si aquesta no és

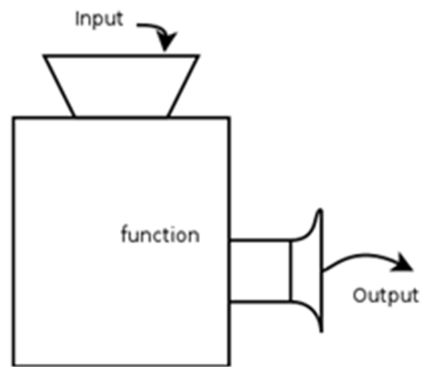


Figura 4. Representació gràfica del funcionament d'una funció.

Font: [Blogit](#).

Ja per acabar tenim les “**entrades i sortides**” (inputs/outputs), que consisteixen en elements que permeten la interacció entre el programa i objectes o entitats del món exterior, com per exemple persones o altres sistemes informàtics. Les “entrades” són les senyals o informació que rep el sistema de l’exterior, i en canvi les “sortides” són les senyals o informació que el sistema envia a l’exterior. Exemples d’entrades serien accions amb el teclat o el ratolí, com clicar sobre la pantalla, prémer una determinada tecla o escriure sobre un espai de text. Exemples de sortides serien pràcticament qualsevol element que apareix al monitor o pantalla, que en la majoria de casos sorgeixen com a resposta d’una entrada.

2.2.2. Biblioteques

Una de les paraules que apareixen cada cop més a mesura que vas aprenent un llenguatge és biblioteca. Quan es parla de biblioteques en programació no s’està fent referència a les biblioteques tradicionals on s’emmagatzemen llibres que es poden consultar, però sí que és cert que ambdós sentits presenten certa analogia. Una biblioteca és una gran font d’informació i coneixement, i és on acudeix molta gent a buscar llibres quan no sap alguna cosa o simplement vol adquirir més coneixement. Doncs el mateix passa amb les biblioteques dels llenguatges de programació: contenen una gran quantitat de codi, i quan un usuari no aconsegueix o no vol programar un determinat bloc perquè és massa complex o requereix molt de temps, aquest accedeix a la biblioteca, agafa el bloc de codi que compleixi els requisits i l’implementa en el seu programa.

Les biblioteques normalment contenen subrutines o funcions, variables, valors, classes... que poden ser utilitzades per un programa, i d'aquesta manera es pot estalviar una gran quantitat de codi perquè no és necessari reescriure un determinat fragment o bloc cada vegada que es vol utilitzar. Fins aquí podria semblar que no fa falta enllaçar una biblioteca a un programa, sinó que simplement el que cal fer és copiar el bloc de codi que es vol i enganxar-lo al programa. Aquesta metodologia, però, és molt poc eficient perquè probablement es perdrà una quantitat de temps considerable buscant el bloc de codi que es vol, però sobretot perquè les biblioteques poden ser utilitzades per múltiples programes alhora, i si s'ha de copiar i enganxar el bloc de codi a cada programa es perd eficiència i rendiment. A diferència dels programes, les biblioteques no funcionen de manera autònoma, és a dir, no funcionen per sí mateixes, sinó que la seva finalitat és ser utilitzades per programes o també per altres biblioteques.

Qualsevol programador pot crear les seves pròpies biblioteques personalitzades segons els seus interessos, però n'existeixen unes anomenades biblioteques estàndard que es comporten d'una manera una mica diferent. Es tracta de biblioteques, normalment de llenguatges compilats (com per exemple C o C++), que es troben disponibles en qualsevol versió d'un determinat llenguatge i que inclouen algorismes, mecanismes i estructures que s'utilitzen usualment en aquest. Es podrien considerar com una espècie d'extensions oficials del llenguatge, tot i que hi ha molts usuaris que no les consideren com a biblioteques sinó que formen part del mateix llenguatge. Ja per acabar, existeixen biblioteques molt potents i molt conegudes que són famoses entre els programadors d'un determinat llenguatge. Aquestes són iguals que les personalitzades de les que es parlava abans, però en aquest cas hi contribueixen i hi participen un gran nombre de persones en la seva creació, manteniment i actualització. Algunes d'aquestes grans biblioteques són jQuery de JavaScript, Pillow de Python, Guava de Java... I moltes més.

2.2.3. Maneres de classificar els llenguatges de programació

Actualment existeixen una gran quantitat de llenguatges de programació, alguns força semblants entre ells i d'altres que no tenen res a veure. Per aquesta raó, al llarg dels anys s'han anat creant diferents classificacions per

veure quins llenguatges posseïen característiques semblants, i així poder triar-ne un o un altre segons les característiques que més convenients siguin pel programa que es vol crear. De classificacions n'hi ha moltes, però aquí en veurem les quatre principals: segons el nivell d'abstracció del llenguatge, segons el seu mètode d'execució, segons el seu propòsit i segons el paradigma de programació.

2.2.3.1. Segons el nivell d'abstracció

El nivell d'abstracció d'un llenguatge fa referència, en termes generals, a la seva proximitat i dependència d'una determinada màquina o "hardware" i a la seva capacitat de generalitzar problemes i processos. Segons el seu nivell d'abstracció tenim els llenguatges de baix nivell i els llenguatges d'alt nivell. De baix nivell en diferenciem dos de diferents: el llenguatge màquina o codi binari i els llenguatges ensambladors. El primer és el més pròxim al "hardware" o màquina ja que és l'únic llenguatge que entenen els processadors directament, sense haver de passar per cap mena traducció. La seva sintaxi (si se li pot dir així) consta únicament de cadenes de zeros i uns per donar les instruccions al "hardware". Els segons, els llenguatges ensambladors (que són diferents segons el processador), tenen un nivell d'abstracció superior al del codi binari perquè, de fet, es van crear per simplificar la tasca de programar, i es va passar de programar amb cadenes de 0 i 1 a fer-ho amb abreviatures de lletres i números que escurçaven i facilitaven el codi. Ambdós tipus de llenguatges, però, són molt propers i dependents de la màquina (la qual cosa fa que en una altra màquina pugui passar una cosa diferent) i s'allunyen molt del llenguatge natural humà. Això impossibilita que es puguin enfrontar problemes de manera general i abstracte.

Els llenguatges que ens falten són els llenguatges d'alt nivell. Es tracta de llenguatges que són totalment independents de l'estructura interna de l'ordinador i que es poden utilitzar en pràcticament qualsevol màquina o computadora. Són llenguatges que s'apropen molt al llenguatge humà, el qual en facilita la comprensió i ajuda a l'abstracció de processos i problemes. Alguns exemples d'aquest tipus de llenguatge són Java, JavaScript, C, Python...

2.2.3.2. Segons el mètode d'execució

Existeixen bàsicament dos tipus de llenguatges segons el mètode d'execució d'un programa: els llenguatges compilats i els llenguatges interpretats. Els llenguatges compilats són aquells que s'han de traduir a llenguatge màquina abans de ser executats. D'aquest procés de traducció se'n diu compilar i és efectuat per compiladors. Un cop compilat es pot executar tantes vegades com es vulgui i sense haver de repetir el procés, cosa que permet que el temps d'execució sigui mínim. Exemple de llenguatges compilats en són Java, C, C++...

D'altra banda, els llenguatges interpretats són aquells que tradueixen el codi a llenguatge màquina pas per pas, línia per línia. Aquest procés és efectuat per un programa intèrpret emmagatzemat dins l'ordinador, i s'ha de repetir cada vegada que s'executa el programa, cosa que fa que el temps d'execució pugui ser més elevat. Alguns exemples de llenguatges interpretats són JavaScript, Python, Ruby...

2.2.3.3. Segons el seu propòsit

Depenent del propòsit i objectiu del nostre programa tenim dos tipus de llenguatges: els llenguatges de propòsit general i els de propòsit específic. Els primers, els llenguatges de propòsit general, són els que són aptes per a o per a fer qualsevol tipus de tasca. Permeten la implementació de quasi qualsevol algoritme i ofereixen un rendiment més que decent en qualsevol tipus de programa que es vulgui crear. És cert que alguns llenguatges de propòsit general són més vàlids per algun tipus de feines, com per exemple la creació de sistemes operatius, però serveixen igualment per dissenyar qualsevol altre tipus de programa. Alguns exemples d'aquest tipus de llenguatge són C++, Java, Python...

Els llenguatges de propòsit específic són els que presenten unes característiques especials i més restringides amb l'objectiu de realitzar tasques més específiques com processar textos, gràfics, dades, efectuar operacions

matemàtiques complexes... Alguns exemples de llenguatges de propòsit específic són SQL, Matlab, etc.

2.2.3.4. Segons el paradigma de programació

Un paradigma de programació és un model de disseny i desenvolupament de programes que segueix unes normes específiques, les quals formen el nucli central del paradigma i són inqüestionables. Aquests paradigmes permeten classificar els llenguatges de programació segons les seves característiques, i cada llenguatge pot incloure's en més d'un paradigma. N'existeixen molts de diferents, però els principals en són sis: l'imperatiu, 'orientat a objectes, el processal, el declaratiu, el lògic i el funcional. El primer de tots, el paradigma imperatiu, és el que indica a l'ordinador com s'ha d'executar el programa, pas per pas, tal com si se n'indiqués l'algoritme. Exemple de llenguatges que segueixen aquest paradigma són Fortran, C, Pascal...

Tot seguit apareix la programació orientada a objectes, que es considera un descendent del paradigma de programació imperatiu. Aquest model es basa en el concepte de objectes i classes, que són entitats indivisibles que contenen estats (dades o informació), comportament (procediments o també anomenats mètodes) i identitat pròpia (cada objecte o classe es diferencia dels altres). Aquests tres elements permeten expressar els programes com a sistemes d'objectes i classes, semblants a una esquema del món real, que defineixen en defineixen la seva interacció, comunicació i diferència. Alguns llenguatges que segueixen aquest paradigma són C++, Java, JavaScript...

A continuació trobem el paradigma processal, un altre descendent del paradigma de programació imperatiu. Aquest tipus d'estructura consisteix en la divisió d'un problema en parts més petites, les quals seran realitzades per subrutines o funcions, que es comunicaran entre elles per a ser executades a mesura que el problema avança. En un programa, les subrutines o funcions contenen una sèrie d'instruccions que s'han de dur a terme, i s'executen cada vegada que el programa les crida. Exemples de llenguatges que deriven d'aquest paradigma en són Pascal, BASIC, ALGOL...

El següent es tracta del paradigma declaratiu, que és el contrari de l'imperatiu. En aquest cas no s'indica a l'ordinador o computadora com s'ha de fer, sinó que simplement declara les coses que s'han de realitzar i complir, sense dir-li es fa. Alguns llenguatges amb aquest paradigma són SQL, HTML (tot i que no és ben bé un llenguatge de programació)...

Seguidament tenim el paradigma lògic, un descendent del paradigma de programació declaratiu. La seva característica principal és que expressa quines tasques s'han de realitzar mitjançant la lògica formal matemàtica. Qualsevol programa escrit amb un llenguatge de programació de paradigma lògic està format per un conjunt de regles lògiques, les quals indiquen quines condicions i normes s'han de seguir i complir. Exemples d'aquest tipus de llenguatges en són Prolog, Datalog...

Ja per acabar hi ha el paradigma funcional, un altre descendent del paradigma de programació declaratiu. El seu funcionament es basa en l'ús i avaluació de funcions matemàtiques per saber quines tasques s'han de realitzar. En aquest tipus de paradigma, el resultat d'una funció depèn únicament dels arguments i informació que se'ls introdueix, de manera que no permet els canvis d'estat ni la mutació de dades, és a dir, els valors aconseguits prèviament mitjançant funcions o processos són fixes i immutables, no es poden reassignar. Alguns llenguatges que incorporen aquest paradigma són Lisp, Scheme, Haskell...

2.3. La web i el seu funcionament

La paraula web és la forma simplificada que s'utilitza per denominar la WWW (World Wide Web). En el dia a dia les persones utilitzen Internet i web sense cap mena de distinció, com si es tractessin de termes intercanviables o equivalents, però això és un greu error. Internet és un sistema global descentralitzat de xarxes (privades, públiques, d'abast mundial o local...) d'ordinadors interconnectades, en les quals aquests ordinadors i computadores poden compartir i enviar-se informació entre sí. Aquesta compartició es pot realitzar mitjançant diferents mètodes: correus electrònics, missatgeria

instantània, telefonia, transferència de fitxers mitjançant el protocol FTP (File Transfer Protocol) i, també, la World Wide Web.

La WWW és una xarxa global de distribució de recursos (normalment en format d'hipertext com HTML) en línia on els documents o recursos web estan identificats per URLs (Uniform Resource Locators) estan connectats entre sí mitjançant enllaços i són accessibles via Internet. Per aconseguir que això es compleixi, es requereix d'una sèrie d'elements bàsics que en permeten el bon funcionament: els servidors web, els clients, el TCP i el IP, els DNS, l'HTTP i, evidentment, els fitxers que componen el recurs web, tant els que fan referència al codi com els de multimèdia.

Els servidor web són ordinadors que alberguen els fitxers d'un recurs web, i normalment tenen una gran capacitat per tal d'emmagatzemar molts fitxers de molts llocs web diferents. Un ordinador domèstic també pot fer de servidor, però sol ser una opció molt problemàtic i ineficient.

Els clients són els usuaris amb dispositius connectats a Internet que disposen de navegadors o altre "software" que permeti navegar i mostrar recursos web.

El TCP i el IP (Transmission Control Protocol i Internet Protocol) són els dos primers protocols d'Internet que apareixen, i es complementen un a l'altre. Cada ordinador disposa d'una direcció IP única (a l'igual que una casa té la seva adreça), i el que fa el protocol IP és enviar informació en forma de petits paquets, cada un dels quals porta també la direcció IP d'origen i la de destinació per saber on ha d'anar. El problema d'aquest protocol és que no assegura que els paquets hagin arribat, que estiguin en bones condicions o estiguin en l'ordre correcte. Això ho soluciona el protocol TCP, que estableix una connexió entre el destí i l'origen fins que s'acaba l'intercanvi d'informació. El TCP també s'encarrega d'evitar qualsevol tipus d'error en aquest intercanvi o transmissió d'informació, ja es tracti de pèrdues de paquets o un ordre incorrecte d'aquests.

El DNS (Domain Name System) és un sistema de nomenclatura descentralitzat per a dispositius i serveis amb adreça IP que es troben connectats a Internet o a una xarxa privada. Els servidors DNS és relacionar l'adreça IP d'un recurs

web (l'adreça real) amb el seu nom de domini. D'aquesta manera, quan un usuari busca un domini concret, el DNS permet trobar el recurs web i que aquest estigui disponible per a l'usuari.

L'HTTP és un altre protocol d'Internet i es tracta d'una sèrie de normes que defineixen la manera de transmetre documents d'hipermèdia a la web. Aquesta és la base de la transmissió d'informació a la WWW, ja que permet accedir a aquests documents mitjançant el domini o fent clic en un enllaç o hipervincle. L'HTTP també defineix una sèrie de mètodes de petició (*Request Methods*) per indicar l'acció que s'ha d'efectuar sobre el recurs demanat. N'existeixen molts i contínuament se'n van afegint més, però els principals i els més bàsics són el GET (demana una representació o còpia del recurs) i el POST (envia informació al recurs, que normalment provocarà que aquest canviï d'alguna manera). Actualment existeix una extensió molt popular de l'HTTP que porta per nom HTTPS. La funció d'aquest últim és exactament la mateixa que l'original, amb l'afegit que estableix una connexió segura entre l'origen i el destí. S'utilitza sobretot en la transmissió d'informació molt rellevant, com per exemple dades bancàries o contrasenyes.

Ara ja es coneixen els elements bàsics que formen la WWW, però com s'interrelacionen per aconseguir que tot funcioni i un usuari pugui accedir a una pàgina o recurs web? És un procés força simple, i que comença evidentment quan un usuari escriu una adreça o domini vàlid al navegador, com per exemple el domini inventat "http://www.hola.cat/". El navegador contacta amb el servidor DNS i li envia l'URL de la pàgina, i aquest li retorna l'adreça IP del servidor on s'allotja la pàgina, és a dir, la seva adreça real. Llavors el navegador envia un missatge de petició HTTP a aquesta adreça IP utilitzant el mètode GET, és a dir, demanant-li al servidor que enviï una còpia de la pàgina web al client o usuari (aquest missatge amb tota la informació que conté és enviat a través de la connexió a internet i utilitzant els protocols TCP/IP). Seguidament, si el servidor aprova la petició, aquest envia un missatge al client dient-li que accepta la petició, i tot seguit li envia els fitxers de la pàgina web en forma de petits paquets (tal com indiquen els protocols TCP/IP). Finalment, el navegador ajunta tots els petits paquets i construeix la pàgina web, que just després mostra a l'usuari. Tot i que sembli un procés molt llarg, en realitat la

seva durada no excedeix les dècimes de segon. Cal esmentar que existeixen altres sistemes de comunicació que no pas el de client i servidor, com per exemple el P2P (Peer-to-peer), però aquest és el principal i més utilitzat en tota la WWW.

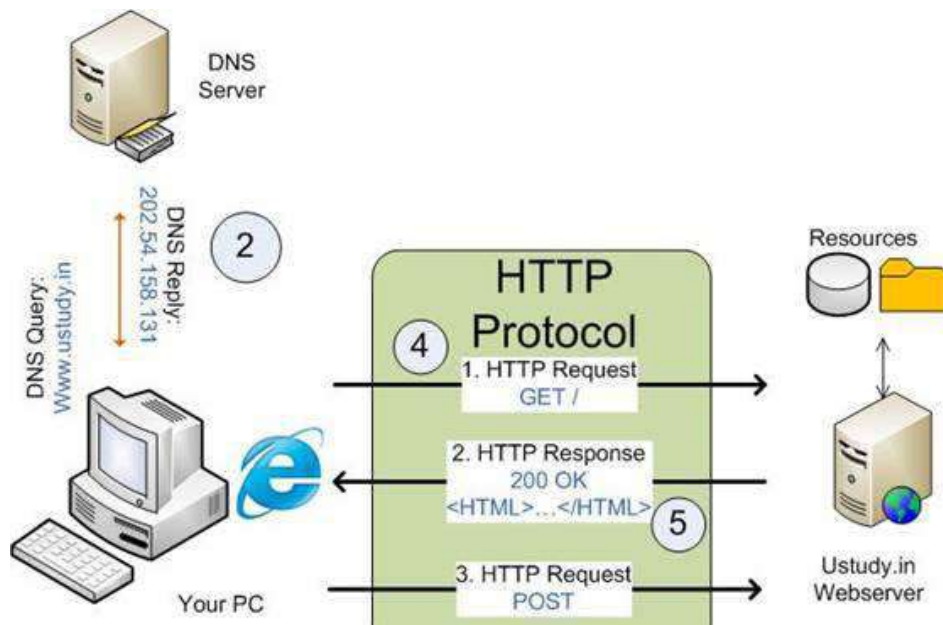


Figura 5. Representació gràfica del funcionament d'una funció. Font: [Universidad de Alicante](http://www.universidaddealicante.es).

2.4. Què és el desenvolupament web?

El desenvolupament web consisteix en el desenvolupament i manteniment de pàgines i aplicacions web, tant si són destinades per a la WWW o per a una xarxa local. Aquest desenvolupament pot anar des de la creació de simples pàgines web que només continguin text fins a complexes aplicacions web com xarxes socials o comerços en línia. El desenvolupament web es pot dividir en dues parts: la part frontal ("front-end") i la part posterior ("back-end"). La part frontal és la que s'encarrega principalment del que l'usuari veu quan accedeix a una pàgina web. En canvi, la part posterior és la que es fa càrrec del que passa darrere de la pàgina o aplicació web, és a dir, s'encarrega del que passa dins del servidor i com això afecta a la pàgina o aplicació. Per l'interès i l'objectiu del treball s'explicarà més detalladament el desenvolupament de la part frontal.

2.4.1. Pàgines estàtiques i pàgines dinàmiques

Abans d'explicar el desenvolupament de la part frontal i el de la part posterior, cal comentar i aclarir dos conceptes molt relacionats amb el tema com són les pàgines estàtiques i les pàgines dinàmiques. Les pàgines estàtiques són aquelles que sempre que es fa una petició al servidor per accedir-hi, responen amb el mateix codi rígid, és a dir, cada vegada que s'hi accedeix el codi es manté "constant", igual, no canvia ni es modifica quan l'usuari es troba a la pàgina. Això no significa un gran problema per a les pàgines petites, però en cas de projectes més grossos aquesta invariabilitat representa un greu problema, ja que si per exemple tenim una pàgina web amb una apartat principal que tracta d'animals i volem afegir un apartat diferent que tracti de la flora, s'haurà de crear una nova pàgina o document per aconseguir-ho. I així anar fent, de manera que si es volguessin afegir cent apartats nous s'haurien de crear cent pàgines diferents. Això resulta molt poc eficient, i més si tenim en compte que a cada pàgina es repeteix l'estructura bàsica. Com a resultat obtindríem que es tarda molt de temps a carregar les pàgines i l'estància en aquell domini se'ns faria molt avorrida. Per tant, si es repeteixen tants elements comuns d'una pàgina a una altra, no seria més fàcil i eficient que el contingut dels apartats s'actualitzés automàticament en comptes d'haver de crear més pàgines? Doncs sí, i això s'aconsegueix a les pàgines dinàmiques.

Les pàgines dinàmiques són aquelles que quan es fa una petició al servidor per accedir-hi, responen amb un codi i un contingut diferents segons el mètode de la petició i la informació que aquesta porti. En les pàgines dinàmiques s'utilitza normalment una base de dades dins d'un servidor per emmagatzemar tota la informació, en comptes de fer-ho als mateixos documents o pàgines com passava amb les estàtiques (si utilitzem l'exemple d'abans, tota la informació continguda en els diferents apartats es trobaria en una base de dades, en comptes d'haver de crear altres pàgines per poder mostrar-la). Quan el servidor rep una petició HTTP amb el mètode GET que demana mostrar un determinat apartat o pàgina, el servidor el localitza, extreu la seva informació de la base de dades i llavors construeix un document d'hipertext (Normalment en HTML) inserint aquesta informació dins d'una plantilla. Aquest document serà, si no es produeix cap error, la resposta que doni el servidor a la petició. Tal com es pot

observar, aquest procés ofereix moltes avantatges respecte al de les pàgines estàtiques, i això és possible gràcies a l'ús d'una base de dades que és accessible fàcilment i que permet emmagatzemar informació de manera òptima. També hi col·labora molt a l'ús d'una plantilla, ja que només varia una part de la seva estructura quan es canvia d'apartat, i d'aquesta manera s'evita que canviï tota l'estructura del document d'hipertext cada vegada que es vol canviar d'apartat o pàgina.

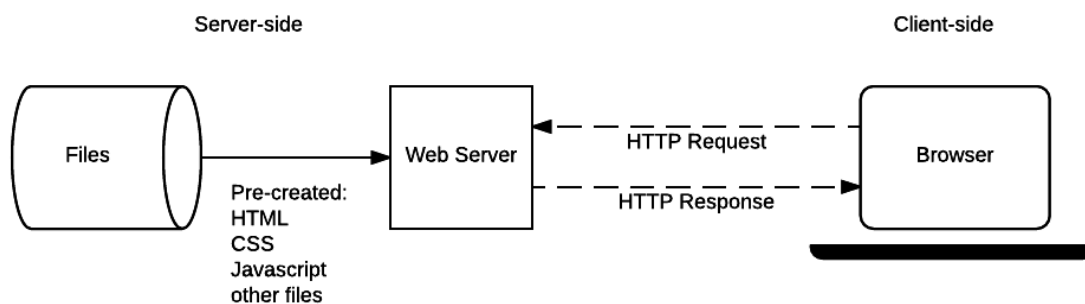


Figura 6. Funcionament de les pàgines dinàmiques. Font: [MDN Web Docs](#).

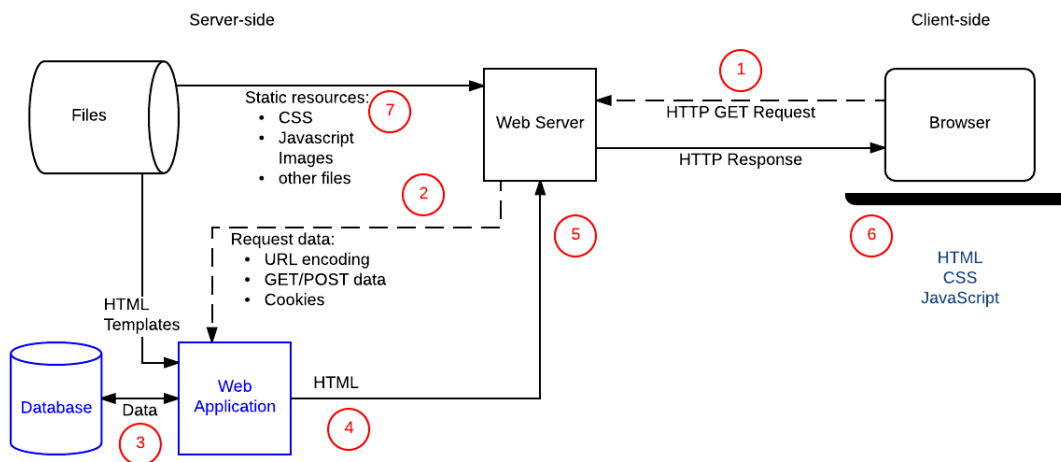


Figura 7. Funcionament de les pàgines dinàmiques. Font: [MDN Web Docs](#).

2.4.2. Desenvolupament de la part frontal d'una pàgina web

El desenvolupament de la part frontal, també anomenada del costat del client (“client-side”), se centra sobretot en el que és visible d'una pàgina o aplicació web per a l'usuari, tot i que també fa referència al seu comportament o funcionalitat. Aconseguir un bon disseny per a la pàgina que es vol crear passa per un procés de programació adequat a les seves característiques. A més de la programació, que és l'element principal del desenvolupament de la part

frontal, n'existeixen d'altres que també són molt importants i s'han de tenir en compte, com són el SEO i l'accessibilitat i disseny responsiu.

2.4.2.1. Programació de la part frontal

La programació de la part frontal, també anomenada del costat del client ("client-side programming"), en termes generals és força simple ja que involucra llenguatges més fàcils que no són considerats llenguatges de programació com a tal. La programació en el costat del client, per si sola, només permet crear pàgines estàtiques, ja que no existeix cap mètode per programar el servidor des de la part frontal. Això no vol dir que no s'hi pugui comunicar (ja que si no hi hagués una connexió amb el servidor no s'hauria pogut ni visualitzar la pàgina), sinó que no es pot alterar el funcionament intern del servidor des de la part frontal. Existeixen principalment dues combinacions de llenguatges vàlides que permeten aquest tipus de programació: la combinació d'HTML, CSS i JavaScript i Ajax.

La combinació triple d'HTML, CSS i JavaScript és una opció molt comuna, i tal com es pot observar consta de tres llenguatges. El primer és l'HTML (Hypertext Markup Language), que és un llenguatge de marcat, i per tant no és un llenguatge de programació com a tal. Un llenguatge de marcat permet codificar un document de manera que incorpora una sèrie d'etiquetes o marques que donen informació de com s'ha d'estructurar el text. Una altra característica important és, tal com el seu nom indica, l'hipertext o enllaços, que permeten la comunicació entre pàgines. Tot i que no té una sintaxi molt complicada, l'HTML representa una part molt important d'una pàgina web. És el que fa tot s'aguanti i funcioni. Sense aquest la implementació de CSS i JavaScript en un entorn web no tindria cap mena de sentit. Si fem un símil amb el cos humà, l'HTML representa l'esquelet d'una pàgina web, el que fa que tot es mantingui dempeus. L'HTML es podria substituir per un cosí seu, l'XHTML (Extensible Hypertext Markup Language). En termes generals són exactament iguals, però l'últim és més modern i és més estricte en el seu llenguatge perquè segueix les característiques estrictes de l'XML (Extensible Markup Language).

El segon llenguatge dels tres és el CSS (Cascading Style Sheets), un llenguatge de fulls d'estil, i que per tant la seva funció bàsica és expressar l'estil i la presentació que han de tenir els documents. El CSS tampoc és un llenguatge de programació com a tal, i per això presenta una sintaxi fàcil i intuïtiva. Tot i que no és absolutament imprescindible com l'HTML a l'hora de crear recurs web, qualsevol pàgina o aplicació que busqui tenir un mínim de futur i èxit necessita CSS. Si l'HTML era l'esquelet, el CSS és el que es veu, l'exterior d'una pàgina web, i per tant si volem tenir una pàgina cuidada, polida i que entri directament pel ulls dels usuaris és primordial dedicar-hi un esforç important al CSS.

El tercer llenguatge dels tres és el JavaScript, que en aquest cas sí que és un llenguatge de programació, i per tant és el que té la sintaxi més complexa dels tres, però en conseqüència també és el més potent i amb el que es poden fer més coses dels tres. No es tracta d'un llenguatge absolutament necessari, però, tal com passa amb el CSS, si es vol que una pàgina o aplicació web tingui un mínim de futur i èxit és necessari el JavaScript. Si comparem una pàgina web amb una persona, ja s'ha dit que l'HTML era l'esquelet i que el CSS era l'exterior, el que es pot veure. Doncs el JavaScript es tractaria dels òrgans interns de la pàgina web, els que s'encarreguen de la funcionalitat i que ens permeten realitzar tot tipus d'accions.

L'altra combinació possible és Ajax. Ajax (Asynchronous JavaScript and XML), tal com el seu nom indica, no es tracta d'un sol llenguatge, sinó que es tracta d'un conjunt de tecnologies i tècniques de programació molt conegudes, però que se'n fa un ús especial i poc comú que permet millorar l'experiència dels usuaris en una pàgina. La característica principal d'Ajax és la seva fluïdesa. A les pàgines web, a diferència de la gran majoria d'aplicacions o programes d'escriptori, quan es clica un botó, s'envia un formulari o alguna altra acció semblant, l'usuari s'ha d'esperar que la pàgina torni a carregar i mostri els resultats de la petició, sense poder interactuar amb la pàgina durant aquest període. Ajax soluciona aquest problema, ja que gestiona en segon pla l'enviament de peticions al servidor, el processament de la informació retornada i tot el que fa referència a la comunicació amb el servidor. D'aquesta manera, els resultats enviats pel servidor són afegits a la pàgina sense haver-la de

refrescar i el usuari pot seguir interactuant amb la pàgina en tot moment. Però com s'aconsegueix exactament això? Doncs això ho fa possible una capa de procés que s'afegeix entre la pàgina web i el servidor, anomenada motor d'Ajax. Aquesta capa intercepta totes les peticions de l'usuari i fa la gestió de la comunicació amb el servidor d'amagat, sense entorpir l'experiència de l'usuari. Les peticions al servidor i les resposta que dona no necessàriament han de produir-se per l'acció de l'usuari, sinó que poden ser asincròniques (D'aquí la primera sigla d'Ajax) i produir-se en el moment que resulti més convenient per l'usuari i la pàgina.

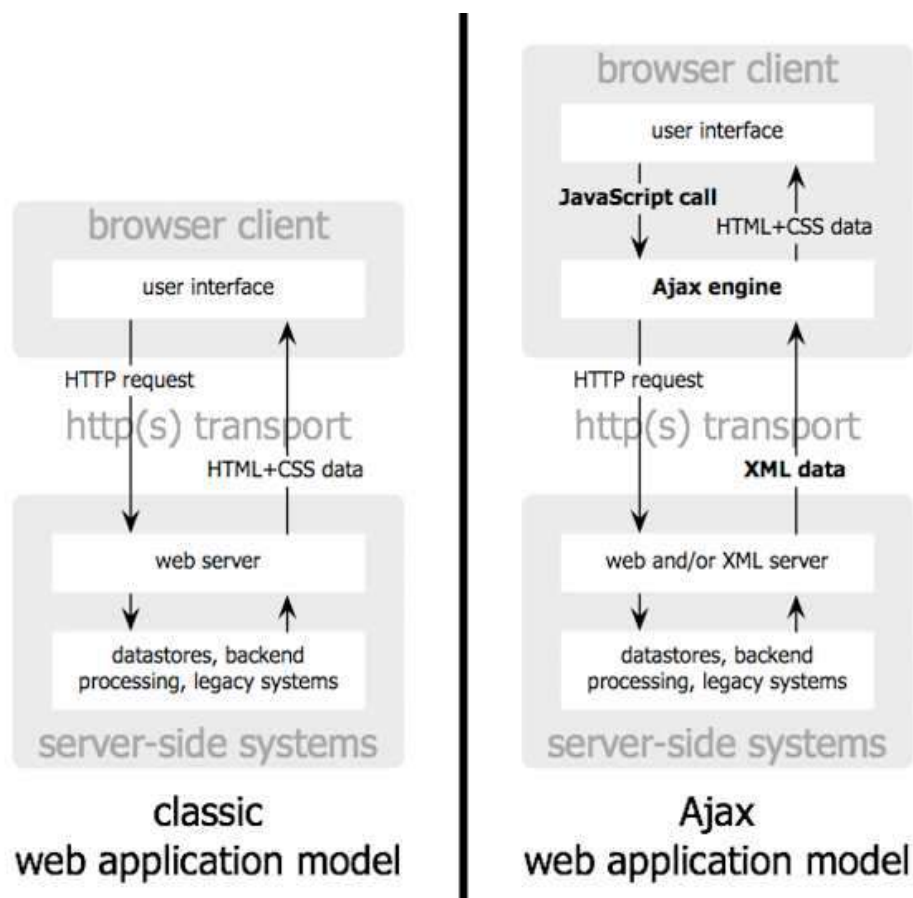


Figura 8. Comparació entre el funcionament d'Ajax i el tradicional. Font: [Adaptive Path](#).

2.4.2.2. Accessibilitat i disseny responsiu

L'accessibilitat i el disseny responsiu en el disseny web tenen com a objectiu fer pàgines web que puguin ser utilitzades per al màxim nombre d'usuaris, i que tots i cada un d'ells puguin gaudir d'una bona experiència a la pàgina. L'accessibilitat fa referència a la primera part, és a dir, permetre que el màxim

nombre de persones puguin utilitzar la pàgina. Hi ha molts usuaris que es troben condicionats d'alguna manera que els dificulta l'accés a les pàgines web, ja sigui per una minusvalidesa, perquè tenen una connexió a Internet limitada, perquè naveguen amb teclat... I l'accessibilitat busca que tots aquests usuaris d'alguna manera siguin tractats iguals i se'ls doni les mateixes oportunitats que a la resta. Per aconseguir-ho s'han de seguir una sèrie de bones pràctiques en el desenvolupament web, des d'algunes més bàsiques a altres que no ho són tant. Les més conegudes i les principals són utilitzar els elements d'HTML segons la funció que els correspon, fer que tots elements d'una pàgina web puguin ser accedits mitjançant el teclat o qualsevol aparell d'assistència (tant si s'encarrega de substituir l'oïda, la vista o qualsevol altre sentit), proporcionar "plans B" per si alguna característica principal falla...

D'altra banda, el disseny responsiu té com a principal objectiu que les pàgines web es vegin i es comportin de la mateixa manera, independentment del dispositiu que s'estigui utilitzant per accedir-hi. Les pàgines han de ser flexibles i s'han d'adaptar a les diferents característiques i factors del dispositiu que s'està usant. D'aquestes característiques a que és probablement la principal és la mida de la pantalla, tot i que també n'hi ha d'altres com la resolució, el navegador que s'utilitza i la versió d'aquest que s'estiguin utilitzant... Les pàgines amb un disseny responsiu requereixen de més esforç que no pas la creació d'una pàgina diferent per a cada tipus de dispositiu, però són molt més fiables, fàcils de mantenir, més eficients... També hi ha una sèrie de bones pràctiques que s'han de seguir per aconseguir un disseny responsiu. Les més conegudes són ajustar la mida dels elements d'una pàgina web en funció de la mida de la pantalla (ús de percentatges), proporcionar imatges i altres arxius en múltiples formats per si el principal falla, proporcionar versions de diferents mides d'una mateixa imatge per adaptar-la a la mida de la pantalla...

2.4.2.3. SEO

El SEO (Search Engine Optimization) tracta d'un conjunt d'estratègies que s'utilitzen per fer més visible una pàgina en els resultats d'un cercador o, dit d'una altra manera, millorar la posició d'una pàgina en el cercador. Això si ho

parem a pensar és molt útil, ja que com millor posicionada estigui una pàgina en el cercador, més visites rebrà, el qual moltes vegades es tradueix en ingressos. Per posar en pràctica aquestes estratègies i poder millorar la posició d'una pàgina en el cercador, primer de tot s'ha d'entendre com funcionen aquests. Els cercadors no tenen un funcionament extremadament complicat, però sí que tenen les seves coses. En termes molt generals, els cercadors contenen una sèrie de petits programes anomenats aranyes ("spiders"), rastrejadors ("crawlers") i bots que s'encarreguen d'analitzar i recollir informació de tots els llocs web, i llavors emmagatzemen en una base de dades la informació recollida, la qual s'assigna a l'URL d'on prové la informació. Així, quan un usuari cerca alguna cosa per internet, es busquen semblances i coincidències a la base de dades, que finalment es mostraran com a resultats de la cerca. Algunes de les tècniques més simples però útils per millorar el posicionament d'una pàgina web són un ús correcte dels elements de l'HTML, ús de paraules clau, tria d'un domini curt i fàcil per a la pàgina...

2.4.3. Desenvolupament de la part posterior d'una pàgina web

El desenvolupament de la part posterior, també anomenat del costat del servidor ("server-side"), es basa en controlar el que passa dins d'un servidor i com això afecta a la pàgina o aplicació. La funció principal és rebre les peticions de la pàgina i triar quin codi o contingut retornar en base aquestes peticions. Si ens fixem en l'apartat de les pàgines estàtiques i dinàmiques, podem observar que aquest funcionament és molt semblant al de les pàgines dinàmiques, i de fet és així perquè el desenvolupament i programació en el costat del servidor és el que permet l'existència de pàgines dinàmiques. La programació en el costat del servidor també ofereix una sèrie de característiques i possibilitats molt interessants per a la creació de qualsevol tipus de pàgina web. Algunes de les més importants són emmagatzematge i transmissió d'informació eficients; experiència adaptada a l'usuari gràcies a la informació que recullen d'aquest; restringir o autoritzar l'accés a un determinat contingut (com és el cas de moltes xarxes socials); guardar dades de sessions (com passa a tots els llocs web que et permeten crear un compte); anàlisi de dades...

Per escriure codi en el costat del servidor es poden utilitzar molts llenguatges diferents, i qualsevol d'ells anirà perfectament bé i tindrà accés total al sistema operatiu del servidor. Els més populars són Python, PHP, JavaScript (amb Node.js), JSP, Ruby... Normalment la programació en el costat del servidor inclou l'ús d'entorns de treball ("frameworks"), els quals ajuden a solucionar problemes i a agilitzar el procés de desenvolupament, ja que proporcionen una gran quantitat de funcionalitat que s'usa sovint en aquest entorn. D'aquesta manera s'evita que els usuaris perdin temps creant-la i implementant-la pel seu propi compte. També existeixen "frameworks" en el costat del client que simplifiquen les tasques de disseny i presentació, però estalvien molta més feina i són més rellevants en el desenvolupament del costat del servidor.

3. Part pràctica

3.1. Introducció al projecte Z4P

Z4P es tracta d'una eina o programa per elaborar i reproduir presentacions de diapositives. El nom prové de “zap”, que en anglès vol dir “produir-se alguna cosa molt ràpidament i amb força”. Aquesta és la idea de Z4P: crear fàcilment i amb rapidesa presentacions de qualsevol tipus i que poden servir per a qualsevol situació, ja sigui un treball de classe o un projecte que s'ha de exposar als dirigents d'una empresa. La substitució de l'A pel 4 és deu bàsicament a qüestions estilístiques i d'originalitat, però també perquè aquest número té una importància personal.

El programa està escrit en HTML, CSS i JavaScript, i com que són llenguatges que funcionen en entorns web requereixen d'un navegador per poder funcionar. La idea inicial pel projecte Z4P era basar-me en el programa S5 (Simple Standards-Based Slide Show System) creat per Eric A. Meyer, el qual va comptar amb la col·laboració i ajuda d'un grup de persones més que van fer que el programa millorés molt. Z4P segueix una línia força semblant al S5 en alguns aspectes, però se'n desmarca de molts altres, i per tant això el fa únic. Z4P no utilitza de moment cap servidor ni tecnologia del costat del servidor, i per tant es tracta d'una pàgina estàtica.

Un dels principals objectius de Z4P, apart de crear presentacions de manera fàcil i ràpida, és introduir l'usuari en el programa i que hi pugui tenir intervenció. La gran majoria d'aquests tipus d'aplicacions tenen moltes opcions de disseny avançades i variades, però molt difícilment veuràs el codi font amb què es troba escrit el programa. Amb Z4P és un dels seus atractius principals, i fent això és molt més probable que l'usuari se senti més rellevant i poderós que utilitzant una eina de presentacions qualsevol. A més, Z4P es pot utilitzar com a recurs per a l'aprenentatge dels tres llenguatges implicats, sobretot d'HTML i CSS. No només observant l'estructura de les diapositives i els botons es pot aprendre, sinó que l'usuari també pot editar al seu gust el codi per provar les seves habilitats o perquè vol canviar algun aspecte del disseny.

Per poder aconseguir acabar Z4P i obtenir el resultat buscat he hagut de seguir un procés com tot bon programador o enginyer de software faria: primer de tot he fet un anàlisi de programes i aplicacions que tenen relació amb el que es vol aconseguir, seguidament he dissenyat i dibuixat un esbós inicial de les diferents interfícies d'usuari possibles en base a l'anàlisi inicial, a les meves possibilitats i als meus gustos personals, i després he pensat en la funcionalitat bàsica del programa, també fixant-me sobretot en l'anàlisi i en els meus interessos. Tot seguit he dividit l'etapa de programació en tres parts en funció dels llenguatges: primer la programació de l'HTML, després la del CSS i finalment la del JavaScript. Ja quasi al final, he creat dos documents HTML mínimament estilitzats amb CSS, un que ensenya com s'ha d'utilitzar i què s'ha de fer per crear la teva primera presentació amb Z4P, i l'altre que conté una llista de totes les característiques del projecte i la seva funcionalitat. Finalment, he dedicat un petit marge de temps a solucionar errors petits i no tant petits del programa per aconseguir una millora notable en el seu funcionament i rendiment. Totes aquestes etapes mencionades han format part del procés, però no han sigut les úniques. Les que queden podran ser vistes més endavant.

3.2. Aprenentatge dels llenguatges necessaris

Com ja he esmentat anteriorment, per aconseguir finalitzar el projecte Z4P he sigut tot un procés premeditat que ha constatat d'un nombre determinat d'etapes. La primera de totes, com no podia ser d'altra manera, ha sigut l'aprenentatge dels tres llenguatges. Els tres professors de la Universitat de Girona amb els quals vaig aconseguir contactar em van ensenyar una pàgina web que contenia unes guies boníssimes per aprendre els tres llenguatges: MDN Web Docs. És una pàgina que forma part de la Fundació Mozilla que ajuda a nous o futurs desenvolupadors de codi a aprendre a programar amb tecnologies web de codi obert. També lidera tota una sèrie de projectes per millorar la WWW i compta amb tota una comunitat que fa aportacions contínues a aquests projectes i també en proposta de nous. També, com és obvi, tenen articles i documents que expliquen amb detall diferents productes de Mozilla i com fer-ne un bon ús.

Doncs bé, gràcies a les guies d'aquesta pàgina he aconseguit aprendre a programar amb HTML i CSS a un nivell elevat, i amb JavaScript a un nivell basic tirant cap a entremig. Tot i ser els dos llenguatges més fàcils, he dedicat molt de temps a l'aprenentatge de l'HTML i el CSS i he llegit amb profunditat tots els articles de la pàgina sobre aquests dos llenguatges. De fet, segurament els he dedicat més temps del que tocava i no hauria d'haver profunditzat tant en alguns aspectes, ja que aquests no m'interessaven pel projecte i resultaven ser força especialitzats. Al JavaScript en canvi, tot i ser el llenguatge més difícil, complet i tenir una sintaxi més complicada; li he dedicat menys temps al seu aprenentatge, però tot i així he aconseguit el que volia acabar fent. Això és perquè ja havia anat veient petits fragments de JavaScript quan aprenia els altres dos llenguatges, i també perquè no necessitava d'un nivell tant elevat de domini del llenguatge com en el cas de l'HTML i CSS. Cal esmentar que les guies les he hagut de seguir totes en anglès ja que no totes es trobaven disponibles en català o castellà, i les que sí que ho estaven tenien una versió molt millor en el seu idioma original.

3.3. Anàlisi de programes i aplicacions semblants

Per tenir una idea de com havia ser el programa que volia crear, la seva interfície d'usuari i una mica la funcionalitat que havia de tenir he analitzat una sèrie d'eines per a fer presentacions. En vaig escollir cinc, que van des de programes als quals Z4P s'hi assemblaria en un futur, fins als programes més utilitzats per fer presentacions arreu del món. Els seleccionats són les següents:

- S5 Slide Show
- S9 Slide Show i variants semblants
- Prezi
- PowerPoint
- Google Slides

De cada una d'aquestes eines n'he fet un anàlisi força aprofundit de les seves característiques:

3.3.1. S5 Slide Show

És un programa per fer presentacions de diapositives creat per Eric A. Meyer. No disposa d'un editor simple i visual com la resta de programes, sinó que mitjançant la utilització d'una sintaxi força simple es poden crear i editar presentacions. Tot i que no hi ha gaire material disponible, és totalment editable i pots crear el que tu vulguis segons les teves necessitats gràcies que està basat únicament en XHTML, CSS i JavaScript. A més, la presentació de diapositives es pot imprimir i els arxius tenen poc pes i són fàcilment compressibles. Per utilitzar-lo no cal tenir connexió a Internet, no s'ha de tenir el programa descarregat per poder editar o visualitzar qualsevol arxiu (només cal la plantilla inicial de 26 KB i tenir un navegador), no necessita registrar-se enlloc i és compatible amb tots els cercadors. El problema és que no és molt visual i pot costar una mica treballar amb la sintaxi si no hi estàs acostumat (tot i que és força intuïtiu).

3.3.2. S9 Slide Show i altres

Existeix el S9, el S6... Són variants del programa original S5 que utilitzen "versions diferents" de CSS, HTML i JavaScript. Tenen més o menys el mateix funcionament que l'original, però arreglen alguns problemes com l'escalament d'imatges. El problema que tenen respecte l'original és que la sintaxi és més complicada i no tant intuïtiva, cosa que no permet el seu ús per part de la majoria de la població.

3.3.3. Microsoft PowerPoint

És un programa per a fer presentacions de diapositives creat per Microsoft. És molt professional, té moltíssimes funcions i disposa d'una gran quantitat de material ja creat que pots utilitzar (sobretot plantilles, detalls estilístics únics...). Es pot utilitzar sense connexió a Internet, però s'ha de tenir descarregat el programa per poder-lo utilitzar. Aquest problema se soluciona amb el *PowerPoint Online*, inclòs dins l'*Office Online*. Amb la versió web pots treballar sense connexió a Internet, no cal descarregar-se res, poden treballar dues persones o més simultàniament, es pot guardar al núvol. El principal problema

és que has de registrar-te per poder utilitzar-lo (només en el cas de la versió web) i haver d'iniciar sessió cada vegada i, el que és més important, és de pagament.

3.3.4. Google Slides

És una eina de Google per a fer presentacions de diapositives que està inclosa dins del servei Google Drive. No és gaire professional i tampoc té una gran quantitat de funcionalitats, tot i que sí que hi ha força material creat que pot ser utilitzat. No s'ha de descarregar res per poder-lo utilitzar, però sí que has d'estar connectat a Internet i has de tenir una compte de Google (que t'obliga a iniciar sessió cada vegada que el vols usar). La part bona és que és gratuït, hi poden treballar simultàniament dues persones o més (de les primeres eines que ho van fer possible), es guarda al núvol, és compatible amb *PowerPoint* i, finalment, amb la compta de Google pots gaudir de moltíssims serveis més, molt útils tots ells.

3.3.5. Prezi

És un programa per a fer presentacions dinàmiques i totalment diferents de les usuals presentacions de diapositives. És professional, té una gran varietat de funcions i també disposa molt de material ja creat que es pot utilitzar. Hi ha dues versions diferents: la gratuïta i la de pagament. Per utilitzar la gratuïta has d'estar connectat a Internet, tot el que facis serà públic i s'ha de guardar al núvol, però el límit d'emmagatzematge és baix. La de pagament inclou una versió d'escriptori que no necessites connexió a Internet per utilitzar-la, pots no fer públic tot el que facis i el límit d'emmagatzematge és superior. A més, pots compartir presentacions offline a altres persones, sense necessitat que tinguin compte de Prezi. La funció de col·laborar simultàniament dues o més persones en un mateix projecte només està permesa en la versió de pagament. Finalment, en les dues versions es requereix d'una compta Prezi, però a la gratuïta has d'iniciar sessió cada vegada i a la de pagament no és necessari.

3.4. Disseny i esbós inicial de les interfícies d'usuari

Basant-me en el que he pogut analitzar de les cinc eines, he fet un disseny a mà de les dues interfícies d'usuari principals que es troben disponibles a Z4P. La primera és la del seu mode normal i inicial, on podem trobar els controls visibles tota l'estona, les diapositives no ocupen tota la pantalla, hi ha un apartat on deixar els comentaris específics per a cada diapositiva, una sèrie de botons desplegable que permeten a l'usuari canviar l'estil de la presentació i, finalment, un botó per guardar i un per activar el mode presentació.

L'altra interfície és pel mode presentació de Z4P, on les diapositives en aquest cas ocupen tota la pantalla, apareix una barra de progrés a la part superior de la pantalla, els controls es troben disponibles per a l'usuari i hi ha un botó que quan es clica, fa aparèixer un menú desplegable que conté uns altres botons que controlen les eines de dibuix, les quals només es poden utilitzar en aquest mode.

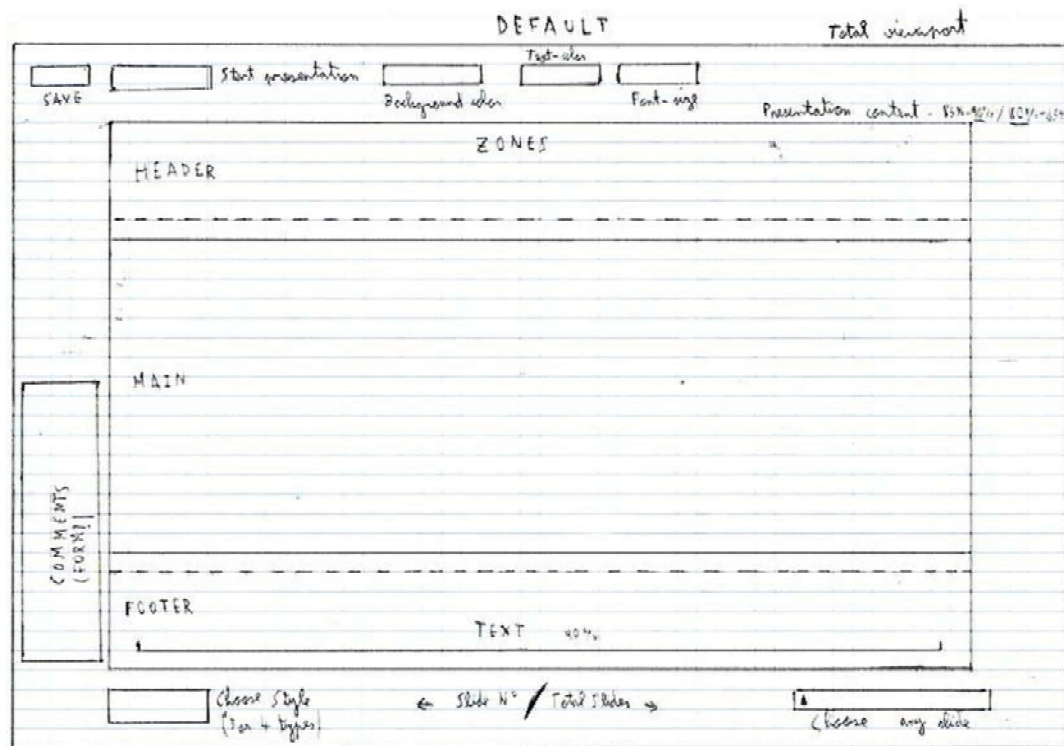


Figura 9. Esquema de l'interfície d'usuari en el mode normal. Font: pròpia.

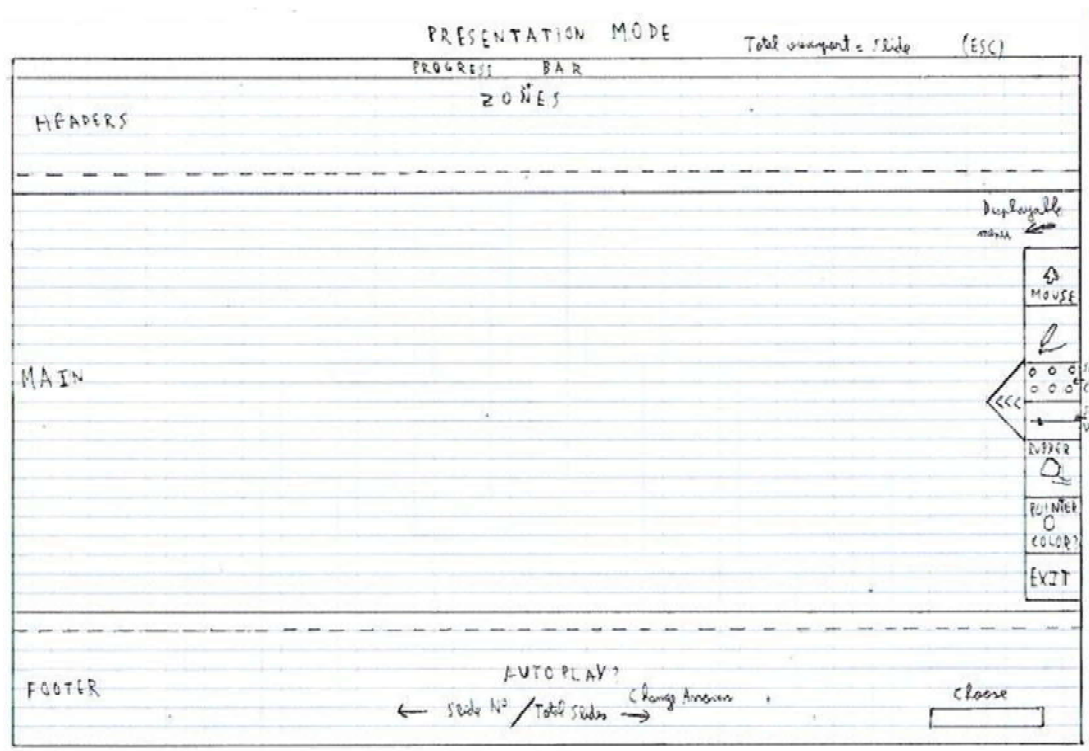


Figura 10. Esquema de l'interfície d'usuari en el mode presentació. **Font:** pròpia.

3.5. Funcionalitat inicial i llistat de característiques del projecte

També fixant-me en les característiques de les cinc eines anteriorment mencionades i analitzades, he elaborat un llistat inicial de les característiques que hauria o podria tenir el resultat final del projecte. Evidentment no són totes característiques definitives, però m'han sigut molt útils com a guia al llarg del projecte. El llistat de característiques el podeu trobar a continuació:

El programa que es vol crear és semblant al ja existent S5. No es partiria totalment de 0 sinó que s'utilitzaria material ja creat com a base de certes parts del projecte. S'utilitzarien "versions" diferents dels llenguatges utilitzats en la creació del S5 (que eren XHTML, CSS i JavaScript). Seria totalment editable i es podria crear el que l'usuari volgués segons les seves necessitats, la lletra escalaria segons la mida i la proporció de la pantalla, els arxius tindrien poc pes i serien fàcilment compressibles, no necessitaria connexió a Internet ni cap tipus de programa descarregat (només un navegador i la plantilla inicial), no seria necessari registrar-se enlloc i seria compatible amb la majoria de

cercadors. El programa tindria també presentaria característiques noves: transicions més dinàmiques entre diapositives (però no tan exagerades com Prezi), sintaxi més intuïtiva i més fàcil d'utilitzar, es podria fer ús de la major part d'arxius multimèdia (fotos, àudios, vídeos...), etc. El que no tindria seria un editor tant visual i fàcil d'utilitzar com el del PowerPoint o el de Google Drive, no hi hauria escalat d'imatge amb la variació de la mida de la pantalla, no podria ser editat simultàniament per dues persones o més i tampoc tindria cap servei propi de guardat al núvol (tot i que es podria guardar perfectament a qualsevol servei d'emmagatzematge de fitxers al núvol com Dropbox o Google Drive).

3.6. Escripció de l'HTML de Z4P

Aquí ja comença la primera etapa de les tres que hi ha de programació, i tal com indica el títol es tracta de la part d'escripció del codi amb HTML. En el cas de Z4P, el seu HTML presenta una estructura força entenedora i accessible. Al principi, per tenir una idea de l'HTML que havia d'afegir per representar les diapositives, em vaig fixar en el codi de S5, però me'n vaig diferenciar ràpidament quan vaig dividir cada diapositiva en tres regions, i també quan vaig afegir un apartat per a deixar comentaris que l'usuari trobés oportuns a cada diapositiva. La resta de codi no s'assembla en gairebé res al del S5 ja que Z4P incorpora molts elements nous.

Si s'observa el codi es pot veure que s'utilitza en moltes ocasions un element anomenat "<div>". Aquest element s'utilitza com a contenidor per albergar altres elements, i el que és més important és que funciona com un bloc que pot situar-se a qualsevol lloc de la pantalla, i no com una simple línia de text. No té una funció tan determinada com la pot tenir un botó, que quan és clicat efectua una acció, sinó que pot ser utilitzat en molts contextos i se sol tenir una tendència per part dels desenvolupadors web d'abusar-ne. Com ja he dit, dins l'HTML de Z4P n'apareixen una bona quantitat, i molts elements que haurien de ser botons han sigut substituïts per <div>s amb aspecte de botons. Això és perquè, estilísticament parlant, els <div>s són més editables que no pas els botons normals i corrents, i també se'ls pot afegir funcionalitat personalitzada. Fins aquí pot no semblar no hi ha cap problema, sinó més aviat que els <div>s

ens aporten més beneficis que no pas inconvenients, però això és perquè no es té en compte l'accessibilitat. Per un usuari corrent que no tingui cap discapacitat això no li representa cap problema, però per tots aquells usuaris que tenen una discapacitat, que utilitzen una versió antiga del navegador, que naveguen amb teclat... Sí que pot representar un problema, i gros, ja que segurament no tenen cap manera de saber que un determinat `<div>` funciona com a botó, i això fa que l'experiència de l'usuari dins la pàgina no sigui bona. Aquest problema, però, ja es troba solucionat a Z4P. Tots els `<div>`s que funcionen com a botó o d'alguna altra cosa que no els correspon, dins del codi van acompanyats d'un atribut que indica quin rol tenen dins la pàgina. Així, els `<div>`s problemàtics són tractats segons el seu rol verdader i s'eviten problemes d'accessibilitat. Per a què els `<div>`s que fan de botons siguin totalment accessibles, s'ha d'afegir un altre atribut que permet als usuaris de teclat viatjar d'un `<div>` a un altre sense cap tipus de problema.

Un dels nous elements que incorpora Z4P respecte S5 (i que segur que haureu notat) són els botons desplegable. Dins HTML existeix un element que crea un botó desplegable (l'element `<select>`), però a l'igual que passava amb els botons normals té un funcionament restringit i gairebé no es pot personalitzar el seu estil. Per poder crear un eina o element que s'ajustés a les meves necessitats, vaig haver-me de llegir un article de MDN Web Docs que explicava com crear botons desplegables totalment personalitzats i accessibles. Incorporar-los dins l'HTML no va ser excessivament complicat, però vaig haver de retocar i adaptar el seu codi per tal que funcionessin dins de Z4P. El seu HTML es basa en la creació d'una llista que conté totes les opcions desplegables, i que quan se selecciona una d'aquestes opcions s'ensenya la que ha estat elegida en l'espai interior de l'element, tal i com ho faria un botó desplegable normal. El que és molt interessant d'aquests tipus de botons personalitzats és que incorporen un "pla b" per si l'element falla. Aquest segon pla es tracta d'un botó desplegable normal i corrent que, com he dit, només apareix si sorgeix algun error amb el "pla a". Aquests elements evidentment que incorporen codi escrit en CSS i JavaScript per poder funcionar correctament, però això es tractarà en els respectius apartats.

Dins del codi també es pot veure que la gran majoria d'elements contenen algun atribut anomenat "class" o "id". Aquests, conjuntament amb algun altre, són atributs molt aconsellables d'afegir als elements perquè permeten diferenciar i seleccionar-los des d'altres llenguatges sense cap tipus de confusió possible.

Finalment, he d'esmentar que el botó de guardar és una mica especial. No és un botó normal i corrent, ni tampoc es tracta d'un <div>. Es tracta d'un enllaç que se li ha canviat el rol a botó i que en aquest context funciona com a bloc, en comptes de funcionar com un simple text com ho faria fer normalment. Perquè un enllaç i no un <div> o un botó? Perquè els enllaços tenen un atribut que els permet descarregar documents, i que en aquest cas és l'opció més simple i que m'interessa més per a Z4P de moment. D'aquest botó se'n tornarà a parlar-ne més endavant.

3.7. Escriptura del CSS de Z4P

La segona etapa de la part de la programació és la que correspon a l'escriptura de codi amb CSS. El CSS de Z4P té moltes hores a darrere per tal de fer el programa visual i atractiu als ulls de l'usuari, però també per assegurar-me que les regles d'estil són dirigides als elements que toquen i no hi ha cap confusió d'objectius per part del codi. Si s'analitza atentament es pot veure que és força complet i que utilitza un munt d'elements diferents per estilitzar l'HTML. Si ens fixem en l'esbós inicial i el comparem amb el resultat final es pot observar que aquest s'ha mantingut fidel al disseny inicial. Les úniques coses que no són iguals són la disposició dels botons superiors, que s'estenen al llarg de tota la pantalla; la posició on s'allotja l'apartat de comentaris, que evidentment és molt millor la seva posició al resultat final ja que l'usuari disposa de més espai i la forma del bloc és més òptima, i els components del quadre de controls, ja que al projecte final s'hi afegeix el botó desplegable per viatjar a qualsevol diapositiva.

Fixant-nos amb el disseny final es pot observar que s'han cuidat molt els petits detalls de disseny de Z4P (que en són uns quants), que al cap i a la fi són els

que acaben marcant les diferències entre programes semblants. Per petits detalls ens podríem referir als degradats de fons que apareixen en tots els botons i en la barra de progrés, així com els canvis de color dels botons quan són clicats. També ens podríem referir als diferents estils de diapositives disponibles per a Z4P. En trobem sis de diferents: “plain”, “classic”, “frame”, “wavy”, “vertical abstraction” i “horitzontal abstraction”. El “plain” és l'estil predeterminat de la presentació, i no té gaire res d'especial. No té cap mena d'estampat i els colors de fons són sempre plans. El “classic” ja presenta un estil més original i no tant monòton com el “plain”. La capçalera i el peu de la presentació són els que canvien de color, mentre que la zona principal sempre manté el mateix. També es pot observar que la capçalera és lleugerament més clara que el peu per diferenciar encara més les diferents zones de la diapositiva. El “frame” és força semblant al “plain”, però a diferència d'aquest presenta un estampat de fons. Els colors de fons segueixen essent plans, però incorpora quatre triangles negres a cada cantonada de la diapositiva que fan que la diapositiva sigui més original i atractiva. El “wavy” és diferència força de la resta d'estils, ja que sí que se'n pot canviar el color de fons, però presenta un patró que produeix un efecte força curiós i que és entretingut i agradable a la vista. Finalment tenim el “vertical abstraction” i el “horitzontal abstraction”, que són molt semblants entre sí. L'única diferència és que la imatge de fons que incorporen els dos estils està situada a diferents llocs de la pantalla, i els usuaris poden triar quina de les dues disposicions els agrada més. Respecte els estils de diapositiva s'ha de comentar una última cosa, i és que cada estil incorpora un tipus de lletra pel títol i un altre pel peu i pel contingut principal de la presentació, i que quan es canvia d'estil s'ha de tenir en compte que també es canvia el tipus de lletra.

Un altre part important del CSS de Z4P és que permet que aquest sigui molt més accessible i responsiu en múltiples aspectes. El primer de tots és que permet que Z4P s'adapti a qualsevol mida de pantalla, ja sigui petita com en el cas d'un mòbil o gran com en el cas d'un monitor o d'un projector. Això és gràcies que tots els elements de componen el programa no tenen una mida fixa (com per exemple 12 mm o 800 px), sinó que tenen mides relatives a la grandària de la pantalla. Això no passa només amb els elements que es

comporten com blocs, sinó que també passa amb totes les lletres i textos, que també escalen la seva mida de manera automàtica si les característiques de la pantalla varien. Una altra de les característiques que fan que Z4P sigui més accessible és que la gran majoria de fitxers que pertanyen al programa tenen una solució alternativa disponible. Això passa amb totes les imatges, botons i tipus de lletra. Amb els botons, si el degradat de fons falla per algun problema de connexió o perquè l'usuari utilitza un navegador antic, apareix un color de fons pla. D'aquesta manera s'evita que l'element torni al seu color de fons al predeterminat i el programa perdi el seu estil bàsic. Amb les imatges i tipus de lletra passa una cosa una mica diferent. En aquest cas, si la primera opció falla perquè el navegador no suporta el seu format, hi ha una imatge alternativa que és exactament igual que la primera, però en un format diferent. D'aquesta manera m'asseguro que les imatges i les lletres es puguin visualitzar la major part de les vegades, ja que és estrany que un navegador no suporti dos o més dels formats més usats per aquell tipus de fitxer. En el cas de Z4P, els formats suportats per les imatges són .png i .svg, i els suportats pels tipus de lletres són .woff, .woff2, .ttf i .otf. Un altre aspecte que fa del Z4P un programa un accessible és el contrast de color. Tots aquells usuaris que tenen alguna discapacitat visual moltes vegades tenen problemes visitant pàgines web perquè no poden distingir si tenen el ratolí sobre un botó o no. Per aquesta raó, quan el ratolí està sobre un botó es nota un clar canvi de color. A més, cada element o apartat de Z4P busca tenir un color que el distingeixi respecte els altres per ajudar a aquests usuaris a trobar el que busquen amb més facilitat.

El CSS de Z4P també conté una sèrie d'ordres que s'executen en cas que un determinat element d'HTML tingui un atribut amb un valor especial. Pràcticament tot el programa es basa en aquesta dinàmica, que és possible gràcies a una mica de JavaScript. Però no totes aquestes ordres s'executen gràcies al JavaScript, sinó que n'hi ha que s'executen només si l'usuari n'edita el valor de l'atribut a mà en el codi del programa. Aquest és el cas de l'alineació del text i de les imatges respecte la diapositiva. Quan l'usuari modifica l'atribut, el text o les imatges (segons l'atribut que hagi canviat) aniran cap a un costat o cap a l'altre segons el valor que hi hagi introduït.

Ja per acabar amb el CSS, tenim els botons desplegable personalitzats. Ja havia parlat d'aquests botons a l'apartat de la programació de l'HTML, i ara toca parlar-ne des del punt de vista de CSS. Quan fas un cop d'ull a un botó d'aquest tipus, ja d'inici es nota una característica especial: la fletxa que indica que es tracta d'un botó desplegable. Aquesta s'aconsegueix aplicant una mica de CSS: s'afegeix just darrere del text del botó un caràcter que simbolitza una fletxa apuntant cap avall, i llavors s'edita de manera que queda una part del botó de diferent color amb la "fletxa" ben diferenciada al seu interior. Pel que fa a les opcions del botó, el codi les col·loca una a sota de l'altra i els dona estils perquè semblin cel·les. Quan s'activa el botó desplegable les cel·les es fan visibles, però quan es deixa d'enfocar aquestes es fan invisibles i desapareixen. Dins les cel·les hi ha una altra ordre que, amb ajuda d'una mica de JavaScript, fa que el color de fons de les opcions es torni negre i el color del seu text blanc quan el ratolí hi passa per sobre. Això és molt útil quan vas navegant pels botons i a més fa pujar el nivell estilístic i de detalls de Z4P. Finalment, hi ha uns petits detalls d'estil que es poden apreciar en alguns botons desplegable, i és que cada opció incorpora una imatge de fons. En el cas dels botons per seleccionar colors, a part de tenir el text de l'opció pintat del color que es vol triar, a la dreta d'aquest apareix un puntet de color que indica encara amb més claredat el color que s'ha triat. En el cas del botó per seleccionar gruix del retolador, les opcions tenen com a imatge de fons una línia que indica esquemàticament el gruix del retolador, i quan hi passes el ratolí per sobre apareix un text que t'indica el "valor" del gruix de la línia.

3.8. Escriptura del JavaScript de Z4P

La tercera i última etapa de la part de la programació és la que fa referència a l'escriptura de codi amb JavaScript. El JavaScript de Z4P també té moltes hores a darrere, segurament més i tot que el CSS, i això es deu que el llenguatge és més complicat i no el tenia tan ben après, tot i que he anat aprenent i agafant destresa amb el llenguatge a mesura que escrivia el programa. Z4P comprèn la funcionalitat bàsica d'un programa per realitzar presentacions, com els canvis de diapositiva mitjançant el ratolí o tecles, el

canvi de fons... I se li afegixen algunes característiques afegides com les eines de dibuix, que li donen un toc original i propi. Hi ha algunes parts del codi (bàsicament les eines de dibuix i els botons desplegable) que estan basades, no copiades, d'alguns blocs de codi extrets d'internet, però la resta es tracta de codi propi.

Si se li fa una ullada al codi es pot veure que té diferents parts (unes millor diferenciades que d'altres): un conjunt de declaracions de variables que senyalen determinats elements de l'HTML, definicions de funcions, bucles, codi corresponent a diferents accions... Un dels blocs de JavaScript que més importància té en tot el codi és probablement un condicional "if else if" que conté dos bucles, ja que d'aquest bloc en depenen moltes altres parts del codi. Aquest condicional el que fa és assignar un número a cada diapositiva de manera ascendent, començant amb el número u i acabant amb el número que li pertoca a l'última diapositiva. També afegix una cel·la al botó desplegable de la secció de controls per cada diapositiva de què disposa la presentació, posant en el text el número de diapositiva que es tracta i el seu títol. A més, hi ha una part del seu codi que permet que quan s'obri un document guardat amb el programa, aquest no contingui més cel·les del compte en el botó o que els números de les diapositives estiguin mal assignats. Finalment, hi ha un bloc de codi (independent al condicional) que dona el valor del número de diapositiva actual i el número total de diapositives a la secció de controls.

Molt relacionades amb el condicional "if else if" del paràgraf anterior hi ha tres funcions que permeten el canvi de diapositiva. La primera és la que permet anar una diapositiva enrere, tant si és amb les fletxes com amb les tecles corresponents. El que fa és seleccionar la diapositiva anterior com a activa, i llavors actualitza correctament el número de diapositiva actual dins el quadre de controls. Si no hi ha cap diapositiva anterior perquè es tracta de la primera, s'inhabilita la fletxa i la navegació amb tecles cap a l'esquerra, i també canvia el color de fons de la fletxa per indicar que no es pot anar més enllà. La segona funció és molt semblant a aquesta, ja que es tracta de la que permet anar a la diapositiva següent. Fa exactament el mateix que la funció anterior, tot i que en aquest cas en el sentit contrari. Quan no hi ha cap diapositiva següent perquè estem a l'última, la fletxa i la navegació amb tecles també s'inhabiliten, i la

fletxa també canvia de color de fons indicant que no es pot avançar més. La tercera funció és la que permet canviar a qualsevol diapositiva mitjançant el botó desplegable de l'apartat de controls. Quan una de les cel·les és seleccionada, la funció busca quina de les diapositives coincideix amb el número de cel·la, i llavors transforma en activa aquella diapositiva.

Ja he parlat dels botons desplegables tant en l'apartat de l'HTML com en el de CSS, i per tant l'únic que falta és parlar de la seva funcionalitat i JavaScript. En aquest cas, el codi és gairebé igual al de l'article de MDN Web Docs mencionat en l'apartat de l'HTML, ja que està molt ben fet i s'adapta perfectament a la funcionalitat que necessitava el programa. Aquest bloc de JavaScript el que fa primer de tot és que si el cercador suporta la llista de cel·les personalitzades, amaga l'opció alternativa. Si no és així, dona pas a l'alternativa i amaga la llista de cel·les. Quan un botó desplegable no es troba ni actiu ni "enfocat" la seva llista de cel·les no es troba visible. Però quan sí que es troba en un dels dos estats, aquestes es fan totalment visibles, i a més a més desactiva i desenfoca la resta de botons. Això evita que hi hagi més d'un d'aquests botons "funcionant" alhora. Quan la llista es troba desplegada i passem el ratolí per sobre les cel·les, aquestes canvien el seu color gràcies a una determinada funció. El canvi d'una cel·la a una altra també es pot efectuar mitjançant la fletxa superior i inferior del teclat. Finalment, en el moment que es clica una de les opcions el text del botó s'actualitza i la llista s'amaga. Aquests tipus de botons també inclouen una sèrie de funcionalitat afegida que permet que siguin accessibles pel màxim nombre d'usuaris, ja tinguin alguna incapacitat, naveguin amb teclat... Tot i que aquesta funcionalitat està molt bé, no fa gran cosa. Actualitza el text del botó i fins aquí arriba. Per tal que canviés el color del text, el color de fons de la diapositiva, l'estil... he hagut de crear una funció que fes que tots aquests botons actuïn tal com ho han de fer. Aquesta funció el que fa és que, quan se selecciona una opció d'un botó, i s'emmagatzema el valor de l'opció escollida. Després, mitjançant un condicional del tipus "if else if" s'identifica el botó que s'ha clicat i es realitza l'acció que pertoca tenint en compte l'opció escollida. Per exemple, si es tracta del botó que tria el color de fons, s'agafa el valor emmagatzemat quan s'ha triat l'opció i, a través d'una funció interna, l'assigna com a color de fons.

A part d'aquestes funcions també n'hi ha moltes altres. Una altra de les més importants que n'activa moltes i en desactiva d'altres és la del mode presentació. Quan s'entra al mode presentació mitjançant el botó o la tecla que pertoca, les diapositives passen a ocupar tota la pantalla i s'inhabiliten els comentaris i la majoria de botons. Per contra, s'activen una sèrie de funcionalitats i elements totalment nous. La primera de totes és la barra de progrés que es troba a dalt de les diapositives. A la diapositiva inicial mai apareix, però a la resta sí, i indica quan es porta i quan falta per acabar la presentació. El seu codi és molt simple, i només es basa en una divisió del número de diapositiva actual entre el nombre total de diapositives, es passa aquest valor a tant per cent i llavors s'assigna aquest valor (relatiu a la mida de la pantalla) a la seva amplada. Una altra funcionalitat que apareix amb la pantalla completa és el control d'accions mitjançant una sèrie de tecles i el ratolí. En el mode normal ja apareixien una sèrie d'accions que es podien realitzar amb el teclat, però quan el mode presentació es troba actiu n'apareixen unes quantes més que permeten avançar o retrocedir diapositives, tornar al mode normal, fer aparèixer el menú lateral... Per canviar diapositives també es pot fer segons el número de clics sobre la pantalla, i per adaptar això a plataformes tàctils m'he basat en un bloc de codi escrit per l'usuari "brettwp" de JSFiddle, i l'he modificat de manera que primer conti el nombre de clics i després actuï en funció d'aquests.

Una altra funcionalitat que també apareix únicament amb el mode presentació és el menú lateral, que incorpora les eines de dibuix. Aquestes es componen d'un llapis o retolador, una goma i un punter. El menú també inclou un botó perquè el ratolí torni al seu funcionament habitual, un botó per sortir del mode presentació i dos botons desplegable: un per seleccionar l'amplada del traç i l'altre per seleccionar-ne el color (aquests botons desplegable funcionen exactament igual que els altres). El llapis i el punter agafen el seu color i amplada dels botons desplegable, i en canvi la goma només n'agafa l'amplada ja que té un "color" fix. Aquesta última funciona exactament igual que el retolador, però pinta amb un color totalment transparent (l'opacitat és 0) que se superposa a tota la resta de traços que es troba. Tot i que totes les opcions del menú lateral són interessants, n'hi ha una que destaca, la del punter. Quan es

clica el seu botó, primer de tot s'esborra tot el que hi havia pintat prèviament (si és que hi havia alguna cosa). Llavors, per aconseguir el seu traç tan curiós m'he fixat en un bloc molt ben dissenyat per l'usuari "@chrisdoble" de CodePen, on el que fa és guardar una sèrie de punts per on ha passat el ratolí recentment, i els uneix mitjançant línies rectes del color desitjat. Aquests punts s'esborren al cap d'un curt període de temps, i això permet l'estela fugaç del punter.

Una cosa que cal fer-ne esment és la visibilitat de l'apartat de controls en mode presentació, ja que aquesta sí que depèn d'una petita part de JavaScript, però el que verdaderament fa que es torni visible o invisible segons la posició del ratolí ho fan una sèrie de blocs de CSS.

Una altra funció molt interessant que inclou Z4P és el Commonmark.js. El Commonmark ([enllaç](#)) és una espècie de "subllenguatge" que permet donar format i estilitzar mitjançant una sèrie d'instruccions simples i sense haver de saber res ni d'HTML ni de CSS. De fet, el seu codi el que fa és que transforma tot el que està escrit amb aquest "subllenguatge" en HTML, de manera que el cercador ho pugui entendre. Aquesta eina tan avançada no n'he tocat res ni n'he volgut tocar res, ja que s'adapta a les necessitats de Z4P, és prou potent, requeriria molt de temps i se surt de les meves capacitats i coneixements actuals de JavaScript. La biblioteca Commonmark.js requereix d'altres biblioteques per poder funcionar. Aquests són la famosa jQuery, Bootstrap i també Lodash. No en vull parlar gaire d'aquestes ja que no són molt rellevants pel codi de Z4P, però sí que es tracta de biblioteques extenses i completíssimes que ofereixen un munt de funcionalitats molt útils i que tenen un gran nombre de persones a darrere que les van millorant i actualitzant contínuament, sobretot en el cas de jQuery i Bootstrap.

Ja per acabar, hi ha una última funció que encara no n'he fet esment, i es tracta del botó de guardar. La seva funcionalitat actualment és força simple, i és que quan es clica, aquest descarrega el document actual d'HTML que l'usuari està desenvolupant (Per tornar a obrir el fitxer correctament, s'ha de guardar a la carpeta amb la resta de fitxers). Aquesta característica està bé perquè funciona a la majoria de cercadors (els d'altres plataformes no està assegurat), però és

una opció força precària. Per resoldre el problema momentàniament no és una mala solució, però en un futur s'hauria de canviar. Per poder emmagatzemar informació sobre el color i el text de la presentació sense haver-la de descarregar i de manera que perdurés en tots els cercadors i plataformes, fan falta coneixements avançats referents a APIs i a tecnologia de servidors per tal de guardar aquesta informació. En aquest moment (per falta de temps i possibilitats) es tracta d'una cosa impossible. Tot i així, baixar-ho i poder-ho tornar a obrir en un altre dispositiu o cercador i que tot el que s'ha avançat o modificat s'hagi mantingut és una opció acceptable.

3.9. Llicències i drets d'autor

Durant la programació de Z4P (sobretot durant el disseny mitjançant CSS) m'he enfrontat en múltiples ocasions als drets d'autor i llicències de codi obert, ja fossin d'imatges, de tipus de lletra o, fins i tot, llibreries de JavaScript externes. En el moment de buscar els fitxers necessaris per a Z4P ja vaig buscar específicament aquells que eren de codi obert o de lliure utilització, i d'aquesta manera m'estalvio un gran nombre de problemes.

De les primeres que em vaig encarregar van ser de les imatges ja que era el més ràpid de fer. La gran majoria d'imatges les vaig treure de la pàgina web Iconfinder, i la gran majoria d'elles permetien utilitzar-les en projectes amb fins comercials. N'hi havia algunes que estaven sotmeses sota llicències Creative Commons, que permeten utilitzar i gaudir d'altres treballs imposant unes condicions mínimes (com per exemple, donar crèdits a l'autor).

Pel que fa al tipus de lletra, les he extret totes de Font Squirrel o de pàgines annexes a aquesta. Totes les lletres es troben sotmeses a alguna llicència de codi obert. La gran majoria segueixen la llicència SIL Open Font License, la qual engloba molts altres tipus de lletres diferents.

Finalment, pel que fa a biblioteques de codi només m'he fixat si estaven sotmeses a algun tipus de llicència especial. Lodash, Bootstrap i jQuery segueixen la llicència del MIT, que permet compartir i utilitzar aquestes llibreries sense gairebé cap tipus de restricció. Dins aquesta llicència també s'indica que

els drets d'autor i semblants estan sotmesos a una versió antiga de Creative Commons. Pel que fa a Commonmark, aquest està sotmès únicament a una llicència Creative Commons, més actual que no pas l'esmentada anteriorment.

Totes les llicències i crèdits dels respectius fitxers apareixen en forma de comentari dins l'element <head> de l'HTML de Z4P, de manera que qualsevol usuari pot accedir-hi i fer una ullada als creadors i llicències. Llegir i entendre totes les llicències ha sigut tediós i cansat, però es tracta d'un part imprescindible per evitar problemes legals de plagi en el futur.

3.10. Elaboració d'un tutorial i d'una llista de característiques

Un cop acabat d'escriure tot el codi de Z4P, encara no se li pot posar punt i final al projecte. Amb l'objectiu d'orientar i guiar els nous usuaris de Z4P en la creació de presentacions de diapositives, he creat dos documents escrits en HTML que ajuden i ensenyen als usuaris a donar els seus primers passos amb el programa. Ambdós documents estan mínimament estilitzats amb CSS per fer més fàcil la seva comprensió i lectura.

El primer es tracta d'un tutorial que ensenya detalladament, pas a pas, el procés que s'ha de seguir per crear la primera presentació de l'usuari. Encara que no es tingui cap coneixement d'informàtica o els ordinadors no siguin el punt fort de l'usuari, això no representa cap problema perquè els passos expliquen molt clarament el procés que s'ha de seguir i, en molts d'ells, hi ha imatges que il·lustren el que s'ha de fer.

El segon es tracta d'un llistat de les característiques que presenta Z4P, on explica què són, en què consisteixen i com funcionen cada una d'elles. També hi podem trobar una taula que ensenya el "subllenguatge" de Commonmark del qual havia parlat anteriorment. En les dues primeres columnes s'explica el que s'ha d'escriure per obtenir el resultat de la tercera columna. Ja per acabar, al final apareix un llistat de totes tecles que tenen un acció assignada dins de Z4P, cosa que ajuda als usuaris a orientar-se millor en el programa.

3.11. Període de correcció d'errors i valoracions

Després d'acabar el programa he deixat un període de temps per revisar i corregir errors del programa, i de fet ha estat molt útil ja que he canviat petites coses que no funcionaven tal com ho havien de fer. Aquest període es tancarà temporalment amb la presentació del projecte, però es tornarà a obrir quan tingui més temps per reprendre'l, i probablement es quedarà obert indefinidament ja que amb el temps vull anar-lo millorant, i també estaré atent a les possibles queixes que els usuaris en facin.

Després d'acabar el programa també he enviat i ensenyat el projecte a una sèrie d'amics, companys de classe i professors per veure què els semblava i quina era la seva opinió. A tots els companys que els ho he ensenyat els ha agradat el projecte i els ha semblat molt interessant. Aquestes interaccions també han sigut molt útils perquè m'han permès solucionar uns quants errors i també m'han donat idees per a futures implementacions o actualitzacions del projecte.

També he contactat amb els professors de l'UdG perquè em donessin una opinió més professional de Z4P, però en el moment de la impressió del treball encara no me l'han pogut fer arribar perquè tenen un volum de feina per realitzar molt gran, i és normal essent principi de curs.

4. Conclusions

Un cop finalitzat el treball ha arribat el moment de valorar els resultats obtinguts en la part pràctica.

El resultat principal de la part pràctica és el codi del programa que he dissenyat. Aquest codi està compost d'una part d'HTML, una de CSS i una altra de JavaScript. La combinació dels tres dona lloc a Z4P, un programa per a realitzar presentacions de diapositives.

Si ens fixem en els objectius inicials i els comparem amb el resultat final del treball, es podria dir que s'han complert amb un grau d'èxit força elevat. El primer objectiu, que fa referència a l'aprenentatge d'informàtica per a un futur proper, s'ha complert correctament ja que he après molts conceptes que abans no coneixia i que segur que em seran d'utilitat en un futur. Potser en podria haver après més, però amb tot el coneixement adquirit i consolidat efectuant el treball ja he fet un bon pas inicial en el món de la informàtica.

El segon objectiu és probablement el que s'ha complert menys satisfactòriament. Sí que he après l'escriptura de codi amb HTML i CSS amb un nivell considerablement elevat, ja que són llenguatges més simples i entenedors i he dedicat molt de temps al seu aprenentatge. En canvi, també he après a programar amb JavaScript, però només amb un nivell bàsic tirant cap a entremig. Això s'explica bàsicament per dues raons: la primera és que JavaScript és un llenguatge complex i conté tots els elements que conformen un llenguatge de programació (a diferència de l'HTML i el CSS). La segona i segurament principal raó és la falta de temps, combinada també amb el fet que Z4P no necessitava de JavaScript extremadament complex per poder funcionar.

Finalment, el tercer objectiu s'ha complert correctament ja que Z4P satisfà la necessitat de crear presentacions de diapositives àgils. Z4P també es tracta d'un programa complex, i més si ens considera que és el primer projecte de programació que es realitza. Per últim, s'han utilitzat de manera apropiada els coneixements adquirits al llarg del projecte per escriure el codi font del programa.

Si parlem de les meves expectatives inicials que tenia del programa, crec que s'han complert totes i cada una d'elles. El resultat final encaixa amb molta exactitud amb el programa que volia realitzar i tenia en ment a l'inici, i per tant n'estic molt satisfet.

Durant la programació de Z4P vaig tenir més d'un problema (sobretot amb el JavaScript) que em va costar de solucionar, i que al final moltes vegades vaig haver de resoldre amb el mètode de prova i error. Tot i això, em va ser molt útil per millorar i aprendre més ja que analitzava el codi després de solucionar l'error, i això em permetia veure on m'havia equivocat i evitar futurs errors semblants.

El principal problema que he tingut al llarg de la realització del treball ha sigut la gestió del temps, ja que potser no hauria d'haver dedicat tant de temps a l'aprenentatge de l'HTML i el CSS i hauria d'haver-me posat a programar abans, tal com vaig fer amb JavaScript. També he d'esmentar que hi he dedicat moltes hores al treball, segurament més de les que tocarien. D'això en tinc la culpa bàsicament jo, ja que vaig triar un tema que sobresortia completament del currículum de batxillerat i que suposava una càrrega de feina molt important.

Finalment, evidentment totes les hores dedicades al treball no seran en va, sinó que, com ja he comentat, en un futur probablement em serviran per quan cursi els estudis d'enginyeria informàtica i també per altres futurs projectes a nivell web que tinc pensats. A més, en el futur també voldria continuar treballant amb Z4P i anar-lo millorant de mica en mica fins a aconseguir un programa totalment capacitat, el qual no tingui res a envejar als programes professionals per a fer presentacions que més s'utilitzen en l'actualitat.

5. Bibliografia

Llibres

Ballard, Phil; Moncur, Micheal. *Ajax, JavaScript y PHP*. Ediciones Anaya Multimedia (Grupo Anaya, S.A.). Madrid, 2009. Pàg. 20-71.

L. Ledford, Jerri. *SEO. Optimización del posicionamiento en buscadores*. Ediciones Anaya Multimedia (Grupo Anaya, S.A.). Madrid, 2009. Pàg. 23-54.

MacDonald, Matthew. *Creación y diseño Web*. Ediciones Anaya Multimedia (Grupo Anaya, S.A.). Madrid, 2010. Pàg. 25-115.

Mateu, Carles. *Desarrollo de aplicaciones web*. Fundació per a la Universitat Oberta de Catalunya. Barcelona, 2004. Pàg. 11-20, 51-104.

Parsons, David. *Desarrollo de Aplicaciones Web Dinámicas con XML y Java*. Ediciones Anaya Multimedia (Grupo Anaya, S.A.). Madrid, 2009. Pàg. 28-88.

van der Vlist, Eric et al. *Programación Web 2.0*. Ediciones Anaya Multimedia (Grupo Anaya, S.A.). Madrid, 2008. Pàg. 69-114, 231-248.

Recursos Web

Anònim. *5 basic elements of programming* [en línia]. Assignment Help. <http://www.assignmenthelp.net/assignment_help/elements-of-programming>. [Consulta: 4 de setembre de 2018].

Anònim. *Clasificación de los lenguajes de programación* [en línia]. La Revista Informática. <<http://www.larevistainformatica.com/clasificacion-de-los-lenguajes-de-programacion.html>>. [Consulta: 6 de setembre de 2018].

Anònim. *Definición de lenguaje de programación*. [en línia]. Definición.de. <<https://definicion.de/lenguaje-de-programacion/>>. [Consulta: 31 d'agost de 2018].

Anònim. *Introduction to Web Development* [en línia]. Codeconquest. <<https://www.codeconquest.com/what-is-coding/web-programming/>>. [Consulta: 25 de setembre del 2018].

Anònim. *Lenguajes de programación: Compilados vs Interpretados* [en línia]. WordPress. <<https://otroespacioblog.wordpress.com/2012/09/02/lenguajes-de-programacion-compilados-vs-interpretados/>>. [Consulta: 10 de setembre de 2018].

Anònim. *What is a type System?* [en línia]. Stack Exchange Network. <<https://softwareengineering.stackexchange.com/questions/333643/what-is-a-type-system>>. [Consulta: 4 de setembre de 2018].

Cavsi. *¿Cuál es la clasificación de los Lenguajes de Programación?* [en línia]. Cavsi. <<http://www.cavsi.com/preguntasrespuestas/cual-es-la-clasificacion-los-lenguajes-de-programacion/>>. [Consulta: 5 de setembre de 2018].

Hernández Yáñez, Luis. *Fundamentos de la programación* [en línia]. Facultad de Informática Universidad Complutense de Madrid. Madrid. <<https://www.fdi.ucm.es/profesor/luis/fp/FP.pdf>>. [Consulta: 30 de juny de 2018].

Mozilla Foundation. *Learn Web Development* [en línia]. Mozilla. <<https://developer.mozilla.org/en-US/docs/Learn>>. [Consulta: del 28 de juny fins al 28 de setembre del 2018]

Pino Reyes, Juan José. *¿Qué es HTML?* [en línia]. DevCode. <<https://devcode.la/blog/que-es-html/>>. [Consulta: 20 de setembre de 2018].

Rouse, Margaret. *HTTP (Hypertext Transfer Protocol)* [en línia]. TechTarget. <<https://searchwindevelopment.techtarget.com/definition/HTTP>>. [Consulta: 16 de setembre de 2018].

Rouse, Margaret. *Internet Protocol* [en línia]. TechTarget. <<https://searchunifiedcommunications.techtarget.com/definition/Internet-Protocol>>. [Consulta: 15 de setembre de 2018].

Rouse, Margaret. *TCP (Transmission Control Protocol)* [en línia]. TechTarget. <<https://searchnetworking.techtarget.com/definition/TCP>>. [Consulta: 15 de setembre de 2018].

Techopedia. *Domain Name System (DNS)* [en línia]. Techopedia. <<https://www.techopedia.com/definition/24201/domain-name-system-dns>>. [Consulta: 17 de setembre de 2018].

Techopedia. *Programming Language* [en línia]. Techopedia. <<https://www.techopedia.com/definition/24815/programming-language>>. [Consulta: 30 d'agost de 2018].

Techopedia. *Semantics* [en línia]. Techopedia. <<https://www.techopedia.com/definition/687/semantics-computing>>. [Consulta: 2 de setembre de 2018].

Techopedia. *Syntax* [en línia]. Techopedia. <<https://www.techopedia.com/definition/3959/syntax>>. [Consulta: 2 de setembre de 2018].

Udacity. *Syntax vs Semantics – Programming Languages* [en línia]. YouTube. <<https://www.youtube.com/watch?v=vP-mn62EF0o>>. [Consulta: 3 de setembre de 2018].

Wikipedia. *Computer Programming* [en línia]. Wikimedia Foundation. <<https://es.wikipedia.org/wiki/Programaci%C3%B3n>>. [Consulta: 29 de juny de 2018].

6. Annexos

6.1. Llicències i crèdits de Z4P

Com ja s'ha esmentat a l'apartat 3.9, he hagut de fer front a una sèrie de temes relacionats amb les llicències i els drets d'autors per evitar qualsevol problema legal de Copyright o semblants. Per aquesta raó, aquí insereixo totes les llicències i crèdits d'autor necessaris per complir tots els requisits i impedir qualsevol tipus de compliació. Totes les llicències i crèdits es troben en anglès perquè pugui entendre-ho un nombre més gran de persones i perquè totes les pàgines i companyies que apareixen treballen en aquest idioma.

<!--Credits and licenses

Images

Thanks to Aleksandr Reva (<https://www.iconfinder.com/Revicon>), Ivan Boyko (<https://www.iconfinder.com/visualpharm>), Timothy Miller (<https://www.iconfinder.com/tmthymllr>), Danilo Demarco (<http://www.danilodemarco.com/>), Rawpixel.com (<https://www.pexels.com/photo/black-and-gray-abstract-graphic-908283/>), and Toptal Subtle Patterns (<https://www.toptal.com/designers/subtlepatterns/>) for their images.

Link to all the images:

https://www.iconfinder.com/icons/309088/arrows_cursor_move_pointer_icon By Ivan Boyko Under Creative Commons 3.0 license (<https://creativecommons.org/licenses/by-sa/3.0/>)

https://www.iconfinder.com/icons/326602/create_edit_pencil_write_icon Free for commercial use.

https://www.iconfinder.com/icons/1021016/eraser_icon Free for commercial use.

https://www.iconfinder.com/icons/216972/jedi_light_saber_red_sith_sword_icon Free for non commercial use.

https://www.iconfinder.com/icons/3005766/account_door_exit_logout_icon By Aleksandr Reva. Free for commercial use.

https://www.iconfinder.com/icons/171417/backwards_rewind_icon By Danilo Demarco. Free for commercial use .

https://www.iconfinder.com/icons/126585/arrow_back_previous_icon,
https://www.iconfinder.com/icons/126569/arrow_forward_next_icon Both by Timothy Miller.

https://www.iconfinder.com/icons/126569/arrow_forward_next_icon

<https://www.pexels.com/photo/black-and-gray-abstract-graphic-908283/> By rawpixel.com Free for commercial use

<https://www.toptal.com/designers/subtlepatterns/vertical-waves/> By Toptal Subtle Patterns Free for commercial use

Fonts

Gentium Basic:

(https://www.fontsquirrel.com/fonts/Gentium-Basic?q%5Bterm%5D=gentium+basic&q%5Bsearch_check%5D=Y) Copyright (c) 2003-2008 SIL International (<http://www.sil.org/>), with Reserved Font Names "Gentium" and "SIL". This Font Software is licensed under the SIL Open Font License, Version 1.1. This license is available with a FAQ at: <http://scripts.sil.org/OFL>

Playfair Display:

(https://www.fontsquirrel.com/fonts/playfair-display?q%5Bterm%5D=playfair&q%5Bsearch_check%5D=Y) Copyright (c) 2010-2012 by Claus Eggert Sørensen (es@forthehearts.net), with Reserved Font Name 'Playfair'.

This Font Software is licensed under the SIL Open Font License, Version 1.1. This license is available with a FAQ at: <http://scripts.sil.org/OFL>

Life Savers:

(https://www.fontsquirrel.com/fonts/life-savers?q%5Bterm%5D=life+s&q%5Bsearch_check%5D=Y) Copyright (c) 2012, Pablo Impallari (www.impallari.com|impallari@gmail.com), Copyright (c) 2012, Rodrigo Fuenzalida (www.rfuenzalida.com|hello@rfuenzalida.com), Copyright (c) 2012, Brenda Gallo (gbrenda1987@gmail.com), Copyright (c) 2012, Iginio Marini. (www.ikern.com|mail@iginomarini.com), with Reserved Font Name "Life Savers". This Font Software is licensed under the SIL Open Font License, Version 1.1. This license is also available with a FAQ at: <http://scripts.sil.org/OFL>

Butler:

(https://www.fontsquirrel.com/fonts/butler?q%5Bterm%5D=butler&q%5Bsearch_check%5D=Y) By <http://fabian-desmet.com/portfolio/butler-font/> Under Attribution-ShareAlike 4.0 International license (CC BY-SA 4.0) (Creative Commons 4.0).

Objectivity:

(https://www.fontsquirrel.com/fonts/objectivity?q%5Bterm%5D=objec&q%5Bsearch_check%5D=Y) Copyright 2017 Alexander Slobzheninov (www.behance.net/slobzheninov). This Font Software is licensed under the SIL Open Font License, Version 1.1. This license is available with a FAQ at: <http://scripts.sil.org/OFL>

Timbra Sans:

(https://www.fontsquirrel.com/fonts/timbra-sans?q%5Bterm%5D=timbra&q%5Bsearch_check%5D=Y) Timbra Sans

(Regular, Bold). Version: 1.00 (Timbra_Sans_Regular.otf, Timbra_Sans_Bold.otf) Copyright (c) 2018, Christian Schaarschmidt (www.illunatic.de). Complete license can be find here: <https://www.fontsquirrel.com/license/timbra-sans> Contact: chris@illunatic.de

Cooper Hewitt:

(https://www.fontsquirrel.com/fonts/cooper-hewitt?q%5Bterm%5D=cooper&q%5Bsearch_check%5D=Y) Copyright (c) 2014, Cooper Hewitt Smithsonian Design Museum (cooperhewitt.org), with Reserved Font Name "Cooper Hewitt". Font Software is licensed under the SIL Open Font License, Version 1.1. This license is available with a FAQ at: <http://scripts.sil.org/OFL>

CommonMark

Copyright (c) 2014, John MacFarlane All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

`lib/normalize-reference.js` is a slightly modified version of <https://github.com/dmoscrop/fold-case>: Copyright Mathias Bynens <<https://mathiasbynens.be/>> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED

TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

lib/from-code-point.js is derived from a polyfill Copyright Mathias Bynens <<http://mathiasbynens.be/>> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

bench/samples/*.md: With the exception of `bench/samples/README.md`, the samples in `bench/samples` are taken from <https://github.com/markdown-it/markdown-it>. Copyright (c) 2014 Vitaly Puzrin, Alex Kocharin. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (C) 2014-15 John MacFarlane

Released under the Creative Commons CC-BY-SA 4.0 license:
<<http://creativecommons.org/licenses/by-sa/4.0/>>.

Bootstrap

The MIT License (MIT) Copyright (c) 2011-2018 Twitter, Inc. Copyright (c) 2011-2018 The Bootstrap Authors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Lodash

Copyright JS Foundation and other contributors <<https://js.foundation/>> Based on Underscore.js, copyright Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors <<http://underscorejs.org/>>

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at <https://github.com/lodash/lodash>. The following license applies to all parts of this software except as documented below:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR

ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright and related rights for sample code are waived via CC0. Sample code is defined as all source code displayed within the prose of the documentation. CC0: <http://creativecommons.org/publicdomain/zero/1.0/>

Files located in the `node_modules` and `vendor` directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

jQuery

The multiple licenses and more information can be found in the following page: <https://jquery.org/license/> -->

Thanks to MDN Web Docs (<https://developer.mozilla.org/en-US/>) for providing their code about custom form widgets, which let me to implement the custom drop-down buttons from Z4P. Under license Creative Commons 2.5.

Also I'm very thankful to user "brettwp" from JSFiddle for the code he wrote (<http://jsfiddle.net/brettwp/J4djY/>), and also to user "@chrisdoble" from CodePen for the code he wrote (<https://codepen.io/chrisdoble/pen/WQLLVp>).

6.2. Tutorial de Z4P

Tal com s'havia esmentat a l'apartat 3.10, s'ha elaborat un petit tutorial per iniciar els nous usuaris de Z4P. Està totalment en català, i és el següent:

Tutorial basic per a l'ús de Z4P

Si ets un nou usuari i no tens cap tipus de coneixement de programació amb HTML o CSS, dissenyar presentacions amb Z4P pot semblar complicat i és possible que et faci tirar enrere, però no t'espantis. Crear les teves pròpies presentacions és molt més fàcil del que sembla, i per iniciar-te i ajudar-te a elaborar la teva primera "Z4Presentation" he preparat un petit tutorial que indica quines parts del codi es poden editar i quines no, i també com donar la forma que més t'interessa per les teves diapositives.

Primers passos

Primer de tot, com és obvi, s'ha de tenir descarregat el paquet d'arxius de Z4P. Si estàs llegint aquest tutorial és molt probable que ja el tinguis descarregat. El paquet d'arxius conté tot el que necessites per crear la teva presentació: una sèrie de carpetes amb arxius que fan funcionar correctament Z4P, un arxiu CSS que dona estil a la presentació, un arxiu JavaScript que aporta la funcionalitat i tres arxius HTML. Aquests tres arxius són el tutorial que estàs llegint ara mateix, un llistat de característiques de Z4P i, finalment, l'arxiu de la

Per canviar el títol, substitueix [Presentation's title here] per el títol corresponent. Per afegir el nom de l'autor i altra informació d'aquest tipus substitueix [Write the author name and related here] per la informació que hi vulguis posar.

Editar i afegir una diapositiva

Per editar una diapositiva es fa de manera molt semblant al que s'ha fet a l'apartat anterior. L'àrea que s'ha d'editar s'indica en la imatge següent.

```
<div class="slide">
  <div class="header"><div>&nbsp;</div><h1>[write a title for the slide]</h1><div>&nbsp;</div></div>
  <div class="main"><img class="i" src=""><p class="content"> <!--write the URL of the image you want to insert (if you have one) inside src=""-->
  [write your main content here]
</p></div>
  <div class="footer"><div>&nbsp;</div><p>[write the footer of the slide]</p><div>&nbsp;</div></div>
  <textarea class="comments" name="selfnotes" maxlength="300" wrap="hard">Add any notes or comments.</textarea>
</div>
```

Per canviar el títol de la diapositiva, substitueix [Write a title for the slide] per el títol corresponent. Per afegir un peu a la diapositiva, substitueix [Write the footer of the slide] pel peu que hi vulguis posar. Finalment, per editar el contingut principal de la diapositiva, substitueix [Write your main content here] per el text principal que hi vulguis posar. Si vols saber com editar correctament l'apartat del contingut principal, fixa't en l'apartat "Commonmark.js" de l'arxiu de característiques. Si no necessites tantes diapositives i en vols suprimir, ho pots fer-ho, i per tant és important que sàpigues que per eliminar-ne una has d'esborrar tots els apartats que apareixen a la imatge de sobre. Si no tens suficients diapositives i en necessites més és molt fàcil. Només cal que vagis al final de l'arxiu i copiïs l'apartat assenyalat en la següent diapositiva.

```
<!-- If you want to have more slide, paste and copy the code below:
<div class="slide">
  <div class="header"><div>&nbsp;</div><h1>[write a title for the slide]</h1><div>&nbsp;</div></div>
  <div class="main"><img class="i" src=""><p class="content">
  [write your main content here]
  </p></div>
  <div class="footer"><div>&nbsp;</div><p>[write the footer of the slide]</p><div>&nbsp;</div></div>
  <textarea class="comments" name="selfnotes" maxlength="300" wrap="hard">Add any notes or comments.</textarea>
</div>
-->
</div>
</div>
</div>
<script src="practical_part.js" type="text/javascript"></script>
</body>
</html>
```

Només cal que copiïs el que es troba assenyalat en la imatge anterior i l'enganxis a sota l'última diapositiva o a sota una diapositiva on t'interessi afegir-li.

Afegir imatges

Afegir una imatge és força simple amb Z4P. Només has de enganxar l'URL de la imatge dins la zona marcada en la imatge següent.

```
<div class="slide">
  <div class="header"><div>&nbsp;</div><h1>[write a title for the slide]</h1><div>&nbsp;</div></div>
  <div class="main"><img class="i" src=""><p class="content"> <!--write the URL of the image you want to insert (if you have one) inside src=""-->
[write your main content here]
</p></div>
<div class="footer"><div>&nbsp;</div><p>[write the footer of the slide]</p><div>&nbsp;</div></div>
<textarea class="comments" name="selfnotes" maxlength="300" wrap="hard">Add any notes or comments.</textarea>
</div>
```

Enganxa l'URL de la imatge dins les cometes de la zona marcada. L'apartat d'inserció d'imatges encara es troba força limitada a Z4P, i de moment no és molt recomanable inserir-ne cap a la diapositiva inicial.

Alineació del text i de les imatges

Per alinear el text i les imatges correctament dins d'una diapositiva, ens hem de fixar en l'atribut `class=""`, que es troba present en la majoria d'elements del text. En la imatge següent s'indica on el trobem dins d'una diapositiva qualsevol.

```
<div class="slide">
  <div class="header center"><div>&nbsp;</div><h1>[write a title for the slide]</h1><div>&nbsp;</div></div>
  <div class="main"><img class="i right" src=""><p class="content justify"> <!--write the URL of the image you want to insert (if you have one) inside src=""-->
[write your main content here]
</p></div>
  <div class="footer left"><div>&nbsp;</div><p>[write the footer of the slide]</p><div>&nbsp;</div></div>
  <textarea class="comments" name="selfnotes" maxlength="300" wrap="hard">Add any notes or comments.</textarea>
</div>
```

Tots els textos i imatges de les diapositives estan alineats cap el costat esquerre de manera predeterminada. Si vols canviar-lo, l'únic que has de fer és anar dins l'atribut `class=""`, afegir un espai (per separar el que es trobi ja dins de l'atribut) i a continuació escriure *left*, *right*, *center* o *justify* en cas que d'estar editant un text, o escriure *left* o *right* en cas d'estar editant una imatge. Com ja he dit, la inserció d'imatges encara es troba força limitada a Z4P i de moment només es pot inserir una imatge a l'esquerra o a la dreta de la diapositiva. A la imatge següent es pot veure un exemple de codi.

```
<div class="slide">
  <div class="header"><div>&nbsp;</div><h1>[write a title for the slide]</h1><div>&nbsp;</div></div>
  <div class="main"><img class="i" src=""><p class="content"> <!--write the URL of the image you want to insert (if you have one) inside src=""-->
[write your main content here]
</p></div>
  <div class="footer"><div>&nbsp;</div><p>[write the footer of the slide]</p><div>&nbsp;</div></div>
  <textarea class="comments" name="selfnotes" maxlength="300" wrap="hard">Add any notes or comments.</textarea>
</div>
```

Tecles útils

La majoria d'elles són intuïtives i només funcionen en el mode presentació. Per veure el llistat complet de tecles que s'utilitzen amb Z4P, visita l'apartat "Tecles útils" de l'arxiu de característiques.

Consells i advertències finals

Es poden incloure fins a un màxim de dues imatges per diapositiva, i perquè es col·loquin de manera correctament una hauria d'estar alienada cap a l'esquerra i l'altra cap a la dreta. També cal fer esment que les imatges que s'insereixin ja han de tenir la mida que haurien de tenir en la presentació final perquè Z4P no disposa de cap eina per variar-ne la mida. També cal tenir en compte que en cas de canviar de plataforma les imatges molt probablement es veuran d'una manera no desitjada perquè aquestes no escalen de manera automàtica en funció de la mida de la pantalla.

Tots els títols i peus de diapositiva estan pensats i dissenyats per contenir només una línia de text. Per tant, si s'excedeix aquest límit Z4P molt probablement no ensenyarà aquests textos de manera adequada. En el cas del contingut principal sí que es permet escriure múltiples línies de text.

Z4P inclou un botó que et descarrega i guarda el projecte amb tot el progrés que haves aconseguit fins aquell moment. Per tal que la presentació funcioni correctament quan tornis a obrir aquest document és molt important guardar-lo en la mateixa carpeta que la resta de documents dels que es parlava en el primer apartat. Si es tenen coneixements d'HTML i CSS poden guardar-los on ells vulguin, però han de tenir present que han d'editar els links del document de la presentació.

6.3. Llistat de característiques de Z4P

Tal com s'havia esmentat a l'apartat 3.10, s'ha elaborat una llista de totes les característiques de Z4P per si els usuaris les volen conèixer. A part, també inclou una explicació detallada de les tecles disponibles a Z4P i de com fer servir Commonmark. Està totalment en català, i és el següent:

Característiques de Z4P

Z4P ofereix múltiples característiques que permeten una millor experiència a l'usuari. Aquestes característiques són les següents:

Disseny responsiu (*Responsive design*)

Z4P es troba disponible per a tots els cercadors (O si més no la gran majoria) moderns i ofereix un servei digne en tots ells. També es tracte d'una eina multiplataforma: tan es pot utilitzar en ordinadors com en mòbils i tauletes gràcies que el seu disseny fa que Z4P s'adapti a la mida de la pantalla. L'únic que no s'adapta en funció de la pantalla són les imatges, i per tant si es canvia de plataforma poden representar un problema.

Accessibilitat (*Accessibility*)

Z4P ha estat pensat i dissenyat per poder oferir el seu servei al màxim nombre de persones possible. L'estructura de l'HTML que compon el programa és organitzada i entenedora, la qual cosa permet als usuaris que usen "Screenreaders" entendre el que està succeint. També està pensat perquè els usuaris que naveguen amb el teclat gaudeixin d'una bona experiència.

Opcions de disseny

Z4P Permet als usuaris canviar l'estil de la presentació al seu gust mitjançant quatre botons desplegable que canvien el color del text, la mida del contingut principal, el color de fons i el "tema" de la presentació. També es poden efectuar altres canvis com l'alineació del text o les imatges editant la plantilla. Si es posseeixin coneixements d'HTML i CSS es pot canviar el disseny de Z4P al gust de l'usuari.

Barra de progrés i comentaris

Z4P ofereix una barra de progrés que indica quan es porta de presentació i quan falta per la seva finalització. Apareix en totes les diapositives menys en la del títol. També, en el mode normal es pot trobar un apartat de comentaris a la part de baix a la dreta de la pantalla on pots anotar els comentaris que vulguis i necessitis de cada diapositiva en concret. Aquesta secció desapareix quan s'activa el mode presentació.

Eines de dibuix

Quan s'activa el mode presentació, l'usuari disposa d'un panell desplegable que li ofereix unes eines simples de dibuix: un retolador, una goma i un punter. Es pot triar el color i el gruix de les tres eines (evidentment, de la goma no es pot variar-ne el color). El menú també disposa d'un botó per tornar al mode normal del ratolí i un altre per sortir de la presentació. S'ha de tenir en compte que quan s'activa el botó del punter s'esborra tot el que s'ha dibuixat, i que quan es canvia de diapositiva tot el que s'ha dibuixat a l'anterior diapositiva s'esborra automàticament. Finalment, quan se surt del mode pantalla completa també s'esborra tot el que s'ha dibuixat en totes les diapositives.

Controls

Z4P disposa d'un apartat per tenir control de les diapositives. Aquest està compost per dues fletxes a ambdós costats per canviar de diapositiva, un botó desplegable per saltar a qualsevol diapositiva i un espai que indica el número de diapositiva actual i el nombre total de diapositives. Quan s'activa el mode presentació els botons es troben ocults, però si es passa el ratolí per sobre la part inferior central de la pantalla els controls es fan visibles. Aquesta última funció probablement no estarà disponible per mòbils i tauletes.

Guardar el projecte

Z4P incorpora un botó per guardar el projecte en desenvolupament. Quan es clica es baixa un document en format HTML que conté el progrés actual del teu projecte. Es pot editar igualment que la plantilla inicial, però perquè funcioni correctament s'ha guardat aquest document en la carpeta original on es trobava la plantilla i la resta de documents. Aquesta funció de guardat pot ser que no es trobi disponible en algun navegador i en dispositius mòbils. A més, és una funció una mica precària i que en futur es millorarà perquè sigui més intuïtiva i més senzilla d'utilitzar per a l'usuari.

Commonmark.js

Z4P inclou dins la seva funcionalitat el Markdown/Commonmark, un "llenguatge de marcat" (Markup language) lleuger que permet als usuaris editar i afegir l'estil que vulguin als seus textos sense haver de tenir coneixement d'HTML ni CSS. La seva "sintaxi" és simple i intuïtiva, i a continuació us deixo les funcions més bàsiques i comunes que permeten editar un text. Si voleu aprendre'n de més específiques o voleu saber-ne més sobre Commonmark visiteu la seva pàgina web.

| S'escriu això | O això | Per aconseguir aquest resultat |
|--|--|--|
| <code>_Cursiva_</code> | <code>*Cursiva*</code> | <i>Cursiva</i> |
| <code>__Negreta__</code> | <code>**Negreta**</code> | Negreta |
| Paràgraf\ Paràgraf | Paràgraf (dos espais) Paràgraf | Paràgraf Paràgraf |
| Paràgraf (un o més espais intercalats) Paràgraf | - | Paràgraf Paràgraf |
| # Capçalera 1 ## Capçalera 2 (Com més # s'utilitzin, més petita serà la capçalera) | Capçalera 1 ===== (1 o més = a la següent línia) Capçalera 2 ----- (1 o més - a la següent línia) (Només disponibles per les capçaleres 1 i 2) | Capçalera 1 Capçalera 2 |

| | | |
|--|--|--|
| - Llista - Llista | + Llista + Llista O bé * Llista * Llista | • Llista • Llista |
| 1. U 2. Dos 3. Tres | 1) U 2) Dos 3) Tres | 1. U 2. Dos 3. Tres |
| [Link](http://www.exemple.cat) O bé <http://www.exemple.cat> | [Link][1] ... [1]:http://www.exemple.cat (normalment quan vols enllaçar múltiples vegades una | Link http://www.exemple.cat |

Tecles útils

Z4P Conté unes quantes accions que poden ser controlades per tecles. A la taula següent les podeu veure totes:

| Acció | Tecles |
|--|---|
| Anar a la següent diapositiva (Només en mode presentació) | <ul style="list-style-type: none"> • Clic sobre la diapositiva • Tocar o pitjar la diapositiva (en cas dels mòbils) • Espai • Fletxa dreta • Fletxa amunt • W • D • + • 8 • 6 |

| | |
|--|--|
| Anar a l'anterior diapositiva (Només en mode presentació) | <ul style="list-style-type: none"> • Doble clic sobre la diapositiva • Tocar o pitjar dues vegades la diapositiva (en cas dels mòbils) • Fletxa esquerra • Fletxa avall • S • A • - • 2 • 4 |
| Sortir del mode presentació | <ul style="list-style-type: none"> • Esc |
| Entrar en mode presentació | <ul style="list-style-type: none"> • F4 |
| Desplegar el menú lateral (Només en mode presentació) | <ul style="list-style-type: none"> • M |
| Esborrar tot el que s'ha dibuixat (Només en mode presentació) | <ul style="list-style-type: none"> • Shift |

6.4. Codi de Z4P

Per motius d'extensió s'ha hagut d'incloure el codi de Z4P als annexos, però en realitat és la part principal del treball i la que té més pes, ja que la part pràctica descriu tot el seu procés d'escriptura.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>[Insert title]</title>
  <!--Metadata-->
    <meta charset="utf-8">
    <meta name="author" content="Jofre Costa i Delgado">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-
scalable=0">
    <meta name="version" content="Z4P 1.0">
  <!--Links to CSS and JS files-->
  <style>
    @font-face {
    font-family: 'butlermedium';
    src: url('fonts/head_abstraction.woff') format('woff'),
      url('fonts/head_abstraction.woff2') format('woff2'),
      url('fonts/head_abstraction.ttf') format('truetype'),
      url('fonts/head_abstraction.otf') format('opentype');
    font-weight: normal;
  
```

```

    font-style: normal;
}
@font-face {
    font-family: 'gentium_book_basicregular';
    src: url('fonts/head_classic.woff') format('woff'),
        url('fonts/head_classic.woff2') format('woff2'),
        url('fonts/head_classic.ttf') format('truetype'),
        url('fonts/head_classic.otf') format('opentype');
    font-weight: normal;
    font-style: normal;
}
@font-face {
    font-family: 'playfair_displayregular';
    src: url('fonts/head_frame.woff') format('woff'),
        url('fonts/head_frame.woff2') format('woff2'),
        url('fonts/head_frame.ttf') format('truetype'),
        url('fonts/head_frame.otf') format('opentype');
    font-weight: normal;
    font-style: normal;
}
@font-face {
    font-family: 'life_saversbold';
    src: url('fonts/head_wavy.woff') format('woff'),
        url('fonts/head_wavy.woff2') format('woff2'),
        url('fonts/head_wavy.ttf') format('truetype'),
        url('fonts/head_wavy.otf') format('opentype');
    font-weight: normal;
    font-style: normal;
}
@font-face {
    font-family: 'cooper_hewittbook';
    src: url('fonts/main_abstraction.woff') format('woff'),
        url('fonts/main_abstraction.woff2') format('woff2'),
        url('fonts/main_abstraction.ttf') format('truetype'),
        url('fonts/main_abstraction.otf') format('opentype');
    font-weight: normal;
    font-style: normal;
}
@font-face {
    font-family: 'objectivitymedium';
    src: url('fonts/main_frame.woff') format('woff'),
        url('fonts/main_frame.woff2') format('woff2'),
        url('fonts/main_frame.ttf') format('truetype'),
        url('fonts/main_frame.otf') format('opentype');
    font-weight: normal;
    font-style: normal;
}
@font-face {
    font-family: 'timbra_sansbold';
    src: url('fonts/main_wavy.woff') format('woff'),

```

```

        url('fonts/main_wavy.woff2') format('woff2'),
        url('fonts/main_wavy.ttf') format('truetype'),
        url('fonts/main_wavy.otf') format('opentype');
font-weight: bold;
font-style: normal;
}
/*Main style of custom select widgets*/
#full_layout .select {
width: 100%;
position: relative;
display: inline-block;
box-sizing: border-box;
padding: .2em 0 .2em 0.7em;
margin: 0;
border-radius: 10px;
box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
background: #8a8a8a;
background: linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
}
#full_layout form {
margin: 0;
}
#full_layout #sel_width .select, #full_layout #sel_color .select {
border-radius: 0;
height: 100%;
padding: .2em 0 .2em 0.2em;
font-size: 0.95em;
padding: 2em 0 0 0.4em;
}
.select:hover, .select:focus, .select.active {
outline: none;
box-shadow: 0 0 3px 1px #22b85b;
}
.select:active {
outline: none;
box-shadow: 0 0 3px 1px #d32525;
}
.select .optlist {
position: absolute;
top: 100%;
left: 0;
}
#ch_style .select .optlist {
position: absolute;
top: -600%;
left: 0;
border-top-width : .1em;
border-radius: 5px 5px 5px 5px;
}

```

```

#chSlide .select .optlist {
  position: absolute;
  top: -500%;
  left: 0;
  border-top-width : .1em;
  border-radius: 5px 5px 5px 5px;
}
#sel_color .select .optlist {
  grid-column: 2;
  grid-row: 5/6;
}
#sel_width .select .optlist {
  grid-column: 2;
  grid-row: 6/7;
}
.select .optlist.hidden {
  max-height: 0;
  visibility: hidden;
}
.select .value {
  display: inline-block;
  width: 100%;
  overflow: hidden;
  vertical-align: top;
  white-space: nowrap;
  text-overflow: ellipsis;
}
.select:after {
  content: "▼";
  position: absolute;
  z-index: 1;
  top: 0;
  right: 0;
  box-sizing: border-box;
  height: 100%;
  width: 10%; /* 20px */
  padding-top: 0.4em;
  padding-right: .3em; /* 1px */
  border-left: .1em solid #000; /* 2px */
  border-radius: 0 9px 8px 0; /* 0 1px 1px 0 */
  background-color: #000000;
  color: #ffffff;
  text-align: center;
}
#sel_color .select:after {
  height: 100%;
  grid-column: 2;
  grid-row: 5/6;
  padding-top: 2em;
  border-radius: 0;
}

```



```

}
#sel_width .select:after {
  height: 100%;
  grid-column: 2;
  grid-row: 6/7;
  padding-top: 2em;
  border-radius: 0;
}
.select .optlist {
  z-index : 2;
  list-style: none;
  margin : 0;
  padding: 0;
  box-sizing : border-box;
  min-width : 100%;
  max-height: 10em;
  overflow-y: auto;
  overflow-x: hidden;
  border: .1em solid #000;
  border-top-width : .05em;
  border-radius: 0 0 5px 5px;
  box-shadow: 0 .2em .4em rgba(0,0,0,.4);
}
.select .option:nth-of-type(2n) {
  background: #ffd2d2;
}
.select .option:nth-of-type(2n+1) {
  background: #ffb3b3;
}
#sel_width .select .option:nth-of-type(2n), #sel_width .select .option:nth-of-
type(2n+1) {
  color: #ffffff00;
}
#full_layout #choose_slide {
  font-size: 1em;
  font-weight: 600;
}
#full_layout #choose_slide option {
  background: none;
  font-size: 1em;
  text-align: center;
}
#full_layout #controls #chSlide {
  font-weight: 600;
  font-size: 1em;
}
#full_layout #controls #chSlide li {
  font-size: 1em;
  font-weight: 600;
}

```

```

.select .option {
  padding: .2em .3em; /* 2px 3px */
}
#full_layout .select .optlist .highlight {
  background: #000000;
  color: #ffffff;
}
#full_layout .select .optlist .highlight[value="black"] {
  background: #000000;
  color: #ffffff;
}
.widget select, .no-widget .select {
  position : absolute;
  left   : -5000em;
  height : 0;
  overflow : hidden;
}
/*Style of the rest of the page*/
body {
  font-size: 0.9vw;
  margin: 0;
  background-color: #ddf7fd;
  overflow: hidden;
  color: black;
  font-family: Arial, Helvetica, sans-serif;
  height: 100%;
}
div #slideshow {
  position: absolute;
  width: 90%;
  height: 80%;
  margin: 0 auto;
  top: 10%;
  left: 5%;
}
.slide {
  position: absolute;
  width: 100%;
  height: 100%;
  box-shadow: 3px 1px 2px #0000004f;
  background-color: #ffffff;
  z-index: 0;
  visibility: hidden;
}
#useful {
  width: 100%;
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: space-around;
}

```

```

    position: absolute;
    top: 3%;
    left: 0;
}
#useful div, #useful a {
    display: inline-block;
    padding: 5px;
    background: #7eedfc;
    background: linear-gradient(130deg, #7eedfc, #0099ff 20%, #7eedfc);
    border-radius: 10px;
    font-weight: 700;
    box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
    text-decoration: none;
    font-size: 1.15em;
    text-align: left;
}
#useful #save {
    outline: none;
    width: 8%;
    text-align: center;
    font-size: 1.4em;
    color: black;
}
#useful #start {
    width: 14%;
    text-align: center;
    font-size: 1.4em;
}
#useful div:hover, #useful div:focus {
    background-color: #0099ff;
    background-image: linear-gradient(130deg, #b8f2fa, #00ff95 50%, #b8f2fa);
    outline: none;
}
#useful div:active {
    box-shadow: inset 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -
2px -2px 5px #ffffffbf;
    outline: none;
}
#useful #blank {
    width: 1%;
    border: none;
    background: none;
    box-shadow: none;
}
#useful select {
    display: inline-block;
    font-size: 1.2em;
    padding: 5px;
    background: #8a8a8a;
}

```

```

background: linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
font-weight: 700;
border-radius: 10px;
box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
text-decoration: none;
}
#useful .select:hover, #useful .select:focus, #useful select:hover, #useful
select:focus {
outline: none;
background: #b9b9b9;
background: linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#useful select:active {
box-shadow: inset 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -
2px -2px 5px #ffffffbf;
}
#font_size {
width: 17%;
}
#t_color {
width: 14%;
}
}
#background_color {
width: 18.25%;
}
}
#lateralmenu {
visibility: hidden;
width: 10%;
height: 60%;
position: absolute;
top: 20%;
right: -7.5%;
display: grid;
grid-template-columns: minmax(15px,1fr) minmax(45px,3fr);
grid-gap: 0;
grid-auto-rows: 8,3333333%;
background-color: none;
z-index: 100;
}
}
#mouse {
background: #8a8a8a url('photos/mouse.png') no-repeat center center/50%;
background: #8a8a8a url('photos/mouse.svg') no-repeat center center/50%;
background: url('photos/mouse.png') no-repeat center center/50%,
linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
background: url('photos/mouse.svg') no-repeat center center/50%,
linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
border: 1.5px solid black;
border-width: 3px 3px 1.5px 3px;
grid-column: 2;
}

```

```

    grid-row: 1/3;
    border-top-left-radius: 10px;
    box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
}
#mouse: hover, #mouse: focus {
    outline: none;
    background: #b9b9b9 url('photos/mouse.png') no-repeat center center/50%;
    background: #b9b9b9 url('photos/mouse.svg') no-repeat center center/50%;
    background: url('photos/mouse.png') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
    background: url('photos/mouse.svg') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#marker {
    background: #8a8a8a url('photos/marker.png') no-repeat center center/50%;
    background: #8a8a8a url('photos/marker.svg') no-repeat center center/50%;
    background: url('photos/marker.png') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
    background: url('photos/marker.svg') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
    border: 1.5px solid black;
    border-width: 1.5px 3px 1.5px 3px;
    grid-column: 2;
    grid-row: 3/5;
    box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
}
#marker: hover, #marker: focus {
    outline: none;
    background: #b9b9b9 url('photos/marker.png') no-repeat center center/50%;
    background: #b9b9b9 url('photos/marker.svg') no-repeat center center/50%;
    background: url('photos/marker.png') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
    background: url('photos/marker.svg') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#selcolor {
    background: #8a8a8a;
    background: linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
    border: 1.5px solid black;
    border-width: 1.5px 3px 1.5px 1.5px;
    grid-column: 2;
    grid-row: 5/6;
    font-size: 0.75em;
    font-weight: 800;
    box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
    margin: 0;
    text-align: center;
}

```

```

}
#full_layout #sel_color {
  background: #8a8a8a;
  background: linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
  border: 1.5px solid black;
  border-width: 1.5px 3px 1.5px 1.5px;
  grid-column: 2;
  grid-row: 5/6;
  font-size: 0.75em;
  font-weight: 800;
  box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
}

#selwidth {
  background: #8a8a8a ;
  background: linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
  border: 1.5px solid black;
  border-width: 1.5px 3px 1.5px 1.5px;
  grid-column: 2;
  grid-row: 6/7;
  font-size: 0.75em;
  font-weight: 800;
  box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
}

#full_layout #sel_width {
  background: #8a8a8a;
  background: linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
  border: 1.5px solid black;
  border-width: 1.5px 3px 1.5px 1.5px;
  grid-column: 2;
  grid-row: 6/7;
  font-size: 0.75em;
  font-weight: 800;
  box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
  margin: 0;
}

#full_layout #sel_width .select:hover, #full_layout #sel_width .select:focus,
#full_layout #sel_color .select:hover, #full_layout #sel_color .select:focus {
  outline: none;
  background: #b9b9b9;
  background: linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}

#full_layout #sel_width .select:active, #full_layout #sel_color .select:active {
  box-shadow: inset 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -
2px -2px 5px #ffffffbf;
}

#rubber {

```

```

background: #8a8a8a url('photos/rubber.png') no-repeat center center/50%;
background: #8a8a8a url('photos/rubber.svg') no-repeat center center/50%;
background: url('photos/rubber.png') no-repeat center center/50%,
    linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
background: url('photos/rubber.svg') no-repeat center center/50%,
    linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
border: 1.5px solid black;
border-width: 1.5px 3px 1.5px 3px;
grid-column: 2;
grid-row: 7/9;
box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
}
#rubber:hover, #rubber:focus {
    outline: none;
    background: #b9b9b9 url('photos/rubber.png') no-repeat center center/50%;
    background: #b9b9b9 url('photos/rubber.svg') no-repeat center center/50%;
    background: url('photos/rubber.png') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
    background: url('photos/rubber.svg') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#pointer {
    background: #8a8a8a url('photos/pointer.png') no-repeat center center/50%;
    background: #8a8a8a url('photos/pointer.svg') no-repeat center center/50%;
    background: url('photos/pointer.png') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
    background: url('photos/pointer.svg') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
    border: 1.5px solid black;
    border-width: 1.5px 3px 1.5px 3px;
    grid-column: 2;
    grid-row: 9/11;
    box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
}
#pointer:hover, #pointer:focus {
    outline: none;
    background: #b9b9b9 url('photos/pointer.png') no-repeat center center/50%;
    background: #b9b9b9 url('photos/pointer.svg') no-repeat center center/50%;
    background: url('photos/pointer.png') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
    background: url('photos/pointer.svg') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#exit {
    background: #8a8a8a url('photos/exit.png') no-repeat center center/50%;
    background: #8a8a8a url('photos/exit.svg') no-repeat center center/50%;
    background: url('photos/exit.png') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
}

```

```

background: url('photos/exit.svg') no-repeat center center/50%,
            linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
border: 1.5px solid black;
border-width: 1.5px 3px 3px 3px;
grid-column: 2;
grid-row: 11/13;
box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
}
#exit:hover, #exit:focus {
outline: none;
background: #b9b9b9 url('photos/exit.png') no-repeat center center/50%;
background: #b9b9b9 url('photos/exit.svg') no-repeat center center/50%;
background: url('photos/exit.png') no-repeat center center/50%,
            linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
background: url('photos/exit.svg') no-repeat center center/50%,
            linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#displayable {
background: #8a8a8a url('photos/displayable.png') no-repeat center
center/50%;
background: #8a8a8a url('photos/displayable.svg') no-repeat center
center/50%;
background: url('photos/displayable.png') no-repeat center center/50%,
            linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
background: url('photos/displayable.svg') no-repeat center center/50%,
            linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
border: 1.5px solid black;
border-width: 3px 1.5px 3px 3px;
grid-column: 1;
grid-row: 5/7;
border-top-left-radius: 15px;
border-bottom-left-radius: 15px;
box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
}
#displayable:hover, #displayable:focus {
outline: none;
background: #b9b9b9 url('photos/displayable.png') no-repeat center
center/50%;
background: #b9b9b9 url('photos/displayable.svg') no-repeat center
center/50%;
background: url('photos/displayable.png') no-repeat center center/50%,
            linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
background: url('photos/displayable.svg') no-repeat center center/50%,
            linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#selcolor:hover, #selcolor:focus, #selwidth:hover, #selwidth:focus {
outline: none;
background: #b9b9b9;
}

```



```

    background: linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#mouse:active, #marker:active, #selcolor:active, #selwidth:active,
#rubber:active, #pointer:active, #exit:active, #displayable:active, #larrow:active,
#rarrow:active {
    box-shadow: inset 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -
2px -2px 5px #ffffffbf;
}
#full_layout .select [value="black"], #full_layout .select .black {
    color: #000000;
    background-image: url('photos/colors/black.png');
    background-image: url('photos/colors/black.svg');
    background-repeat: no-repeat;
    background-position: 80% center;
    background-size: 8%;
}
#full_layout #sel_color .select .optlist .option:first-child, #full_layout
#background_color .select .optlist .option:first-child, #full_layout #t_color .select
.optlist .option:first-child {
    background-image: none;
}
#full_layout .select [value="white"], #full_layout .select .white {
    color: #ffffff;
    background-image: url('photos/colors/white.png');
    background-image: url('photos/colors/white.svg');
    background-repeat: no-repeat;
    background-position: 80% center;
    background-size: 8%;
}
#full_layout .select [value="darkgrey"], #full_layout .select .darkgrey {
    color: #4c4c4c;
    background-image: url('photos/colors/darkgrey.png');
    background-image: url('photos/colors/darkgrey.svg');
    background-repeat: no-repeat;
    background-position: 80% center;
    background-size: 8%;
}
#full_layout .select [value="lightgrey"], #full_layout .select .lightgrey {
    color: #999999;
    background-image: url('photos/colors/lightgrey.png');
    background-image: url('photos/colors/lightgrey.svg');
    background-repeat: no-repeat;
    background-position: 80% center;
    background-size: 8%;
}
#full_layout .select [value="yellow"], #full_layout .select .yellow {
    color: #ffff00;
    background-image: url('photos/colors/yellow.png');
    background-image: url('photos/colors/yellow.svg');
    background-repeat: no-repeat;
}

```

```

        background-position: 80% center;
        background-size: 8%;
    }
    #full_layout .select [value="orange"], #full_layout .select .orange {
        color: #ff7f00;
        background-image: url('photos/colors/orange.png');
        background-image: url('photos/colors/orange.svg');
        background-repeat: no-repeat;
        background-position: 80% center;
        background-size: 8%;
    }
    #full_layout .select [value="red"], #full_layout .select .red {
        color: #ff0000;
        background-image: url('photos/colors/red.png');
        background-image: url('photos/colors/red.svg');
        background-repeat: no-repeat;
        background-position: 80% center;
        background-size: 8%;
    }
    #full_layout .select [value="lightgreen"], #full_layout .select .lightgreen {
        color: #00ff00;
        background-image: url('photos/colors/lightgreen.png');
        background-image: url('photos/colors/lightgreen.svg');
        background-repeat: no-repeat;
        background-position: 80% center;
        background-size: 8%;
    }
    #full_layout .select [value="darkgreen"], #full_layout .select .darkgreen {
        color: #004c00;
        background-image: url('photos/colors/darkgreen.png');
        background-image: url('photos/colors/darkgreen.svg');
        background-repeat: no-repeat;
        background-position: 80% center;
        background-size: 8%;
    }
    #full_layout .select [value="lightblue"], #full_layout .select .lightblue {
        color: #00ffff;
        background-image: url('photos/colors/lightblue.png');
        background-image: url('photos/colors/lightblue.svg');
        background-repeat: no-repeat;
        background-position: 80% center;
        background-size: 8%;
    }
    #full_layout .select [value="darkblue"], #full_layout .select .darkblue {
        color: #0000ff;
        background-image: url('photos/colors/darkblue.png');
        background-image: url('photos/colors/darkblue.svg');
        background-repeat: no-repeat;
        background-position: 80% center;
        background-size: 8%;
    }

```

```

}
#full_layout .select [value="brown"], #full_layout .select .brown {
  color: #5e2400;
  background-image: url('photos/colors/brown.png');
  background-image: url('photos/colors/brown.svg');
  background-repeat: no-repeat;
  background-position: 80% center;
  background-size: 8%;
}
#full_layout .select [value="pink"], #full_layout .select .pink {
  color: #ff00ff;
  background-image: url('photos/colors/pink.png');
  background-image: url('photos/colors/pink.svg');
  background-repeat: no-repeat;
  background-position: 80% center;
  background-size: 8%;
}
#full_layout .select [value="purple"], #full_layout .select .purple {
  color: #7f00ff;
  background-image: url('photos/colors/purple.png');
  background-image: url('photos/colors/purple.svg');
  background-repeat: no-repeat;
  background-position: 80% center;
  background-size: 8%;
}
#full_layout .select [value="vthin"], #full_layout .select .vthin {
  background-image: url('photos/vthin.png');
  background-image: url('photos/vthin.svg');
  background-repeat: no-repeat;
  background-position: 5% center;
  background-size: 90%;
  padding: 1.5em;
  box-sizing: content-box;
}
#sel_width .select .optlist .highlight[value="vthin"], #sel_width .select .optlist
.highlight .vthin {
  background-color: #fff6f6;
  background-image: url('photos/vthin.png');
  background-image: url('photos/vthin.svg');
  background-repeat: no-repeat;
  background-position: 5% center;
  background-size: 90%;
}
#full_layout .select [value="thin"], #full_layout .select .thin {
  background-image: url('photos/thin.png');
  background-image: url('photos/thin.svg');
  background-repeat: no-repeat;
  background-position: 5% center;
  background-size: 90%;
  padding: 1.5em;
}

```

```

    box-sizing: content-box;
}
#sel_width .select .optlist .highlight[value="thin"], #sel_width .select .optlist
.highlight .thin {
    background-color: #fff6f6;
    background-image: url('photos/thin.png');
    background-image: url('photos/thin.svg');
    background-repeat: no-repeat;
    background-position: 5% center;
    background-size: 90%;
}
#full_layout .select [value="medium"], #full_layout .select .medium {
    background-image: url('photos/medium.png');
    background-image: url('photos/medium.svg');
    background-repeat: no-repeat;
    background-position: 5% center;
    background-size: 90%;
    padding: 1.5em;
    box-sizing: content-box;
}
#sel_width .select .optlist .highlight[value="medium"], #sel_width .select .optlist
.highlight .medium {
    background-color: #fff6f6;
    background-image: url('photos/medium.png');
    background-image: url('photos/medium.svg');
    background-repeat: no-repeat;
    background-position: 5% center;
    background-size: 90%;
}
#full_layout .select [value="thick"], #full_layout .select .thick {
    background-image: url('photos/thick.png');
    background-image: url('photos/thick.svg');
    background-repeat: no-repeat;
    background-position: 5% center;
    background-size: 90%;
    padding: 1.5em;
    box-sizing: content-box;
}
#sel_width .select .optlist .highlight[value="thick"], #sel_width .select .optlist
.highlight .thick {
    background-color: #fff6f6;
    background-image: url('photos/thick.png');
    background-image: url('photos/thick.svg');
    background-repeat: no-repeat;
    background-position: 5% center;
    background-size: 90%;
}
#full_layout .select [value="vthick"], #full_layout .select .vthick {
    background-image: url('photos/vthick.png');
    background-image: url('photos/vthick.svg');
}

```

```

background-repeat: no-repeat;
background-position: 5% center;
background-size: 90%;
padding: 1.5em;
box-sizing: content-box;
}
#sel_width .select .optlist .highlight[value="vthick"], #sel_width .select .optlist
.highlight .vthick {
background-color: #fff6f6;
background-image: url('photos/vthick.png');
background-image: url('photos/vthick.svg');
background-repeat: no-repeat;
background-position: 5% center;
background-size: 90%;
}
option {
font-size: 1.2em;
font-weight: 700;
}
#full_layout option:nth-of-type(2n) {
background: #ffd2d2;
}
#full_layout option:nth-of-type(2n+1) {
background: #ffb3b3;
}
#controls {
box-sizing: border-box;
display: grid;
width: 30%;
height: 9.5%;
position: absolute;
bottom: 0;
left: 35%;
grid-template-columns: 1fr repeat(3, 3fr) 1fr repeat(5, 5fr) 1fr repeat(3, 3fr)
1fr;
grid-gap: 0;
grid-auto-rows: 10%;
background-color: #ff8282;
background-image: linear-gradient(130deg, #ff8282, #ff4b4b 50%, #ff8282);
border: 4.5px double black;
z-index: 100;
}
#larrow {
grid-column: 2/5;
grid-row: 2/10;
border: 2px solid black;
background: #ff4b4b url('photos/arrow_left.png') no-repeat center
center/50%;
background: #ff4b4b url('photos/arrow_left.svg') no-repeat center center/50%;
background: url('photos/arrow_left.png') no-repeat center center/50%,

```

```

        linear-gradient(130deg, #ff4b4b, #fc2929 50%, #ff4b4b);
background: url('photos/arrow_left.svg') no-repeat center center/50%,
        linear-gradient(130deg, #ff4b4b, #fc2929 50%, #ff4b4b);
        box-shadow: 1px 1px 1px black, inset 2px 2px 4px #000000a9, inset -2px -
2px 3px #ffffff6f;
    }
#larrow:hover, #larrow:focus {
    outline: none;
    background: #ff8d4b url('photos/arrow_left.png') no-repeat center
center/50%;
    background: #ff8d4b url('photos/arrow_left.svg') no-repeat center center/50%;
    background: url('photos/arrow_left.png') no-repeat center center/50%,
        linear-gradient(130deg, #ff8036, #fcff4b 50%, #ff8036);
    background: url('photos/arrow_left.svg') no-repeat center center/50%,
        linear-gradient(130deg, #ff8036, #fcff4b 50%, #ff8036);
}
#rarrow {
    grid-column: 12/15;
    grid-row: 2/10;
    border: 2px solid black;
    background: #ff4b4b url('photos/arrow_right.png') no-repeat center
center/50%;
    background: #ff4b4b url('photos/arrow_right.svg') no-repeat center
center/50%;
    background: url('photos/arrow_right.png') no-repeat center center/50%,
        linear-gradient(130deg, #ff4b4b, #fc2929 50%, #ff4b4b);
    background: url('photos/arrow_right.svg') no-repeat center center/50%,
        linear-gradient(130deg, #ff4b4b, #fc2929 50%, #ff4b4b);
    box-shadow: 1px 1px 1px black, inset 2px 2px 4px #000000a9, inset -2px -
2px 3px #ffffff6f;
}
#rarrow:hover, #rarrow:focus {
    outline: none;
    background: #ff8d4b url('photos/arrow_right.png') no-repeat center
center/50%;
    background: #ff8d4b url('photos/arrow_right.svg') no-repeat center
center/50%;
    background: url('photos/arrow_right.png') no-repeat center center/50%,
        linear-gradient(130deg, #ff8036, #fcff4b 50%, #ff8036);
    background: url('photos/arrow_right.svg') no-repeat center center/50%,
        linear-gradient(130deg, #ff8036, #fcff4b 50%, #ff8036);
}
#curr_slide {
    grid-column: 6/11;
    grid-row: 2/6;
}
#choose_slide {
    grid-column: 6/11;
    grid-row: 7/10;
}

```

```

#full_layout #chSlide {
  grid-column: 6/11;
  grid-row: 7/10;
}
#controls div {
  display: flex;
  align-items: center;
  justify-content: center;
}
#controls div span {
  font-size: 2.5em;
  font-weight: 900;
}
#controls #chSlide span{
  font-size: 1em;
  font-weight: 600;
}
#progress_bar {
  visibility: hidden;
  position: sticky;
  top: 0;
  left: 0;
  width: 100%;
  height: 0.9vw;
  background: #00ffbf;
  background: linear-gradient(110deg, #00ffbf, #34ff45, #00ffbf);
  z-index: 100;
}
#choose_style {
  position: absolute;
  width: 17%;
  bottom: 3.5%;
  left: 10%;
  display: inline-block;
  padding: 0;
  font-size: 1.2em;
  background: #8a8a8a;
  background: linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
  font-weight: 700;
  border-radius: 10px;
  box-shadow: 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -2px -
2px 5px #ffffffbf;
  text-decoration: none;
}
#ch_style {
  position: absolute;
  width: 17%;
  bottom: 3.5%;
  left: 10%;
  display: inline-block;

```

```

font-size: 1.2em;
font-weight: 700;
text-decoration: none;
}
#choose_style:hover, #ch_style .select:hover, #ch_style .select:focus,
#choose_style:focus, #full_layout .slide .comments:hover, #full_layout .slide
.comments:focus {
    outline: none;
    background: #b9b9b9;
    background: linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#choose_style:active, #ch_style .select:active {
    box-shadow: inset 2px 2px 1px black, inset 2px 2px 4px #0000008e, inset -
2px -2px 5px #ffffffbf;
}
#full_layout .slide .comments {
    position: absolute;
    top: 101%;
    left: 68%;
    width: 31%;
    height: 10%;
    font-size: 1.3em;
    font-weight: 550;
    border: 3px dashed black;
    background: #8a8a8a;
    background: linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
}
#full_layout .slide[name='plain'] {
    background-color: #ffffffda; /* Very important -- da*/
}
#full_layout .slide[name='plain'] .header {
    font-family: Georgia, 'Times New Roman', Times, serif;
}
#full_layout .slide[name='classic'] {
    background: #ffffffda; /* Very important -- da*/
    background: linear-gradient(200deg, #c2c2c2, whitesmoke, #c2c2c2);
    border: 1px solid black;
    font-family: Verdana, Geneva, Tahoma, sans-serif;
}
#full_layout .slide[name='classic'] .header {
    background-color: #5e2400cd; /* Very important -- cd*/
    font-family: 'gentium_book_basicregular', 'Times New Roman', Times, serif;
}
#full_layout .slide[name='classic'] .footer {
    background-color: #5e2400;
}
#full_layout .slide[name='frame'] {
    background-color: #ffffffe6; /* Very important -- e6*/
    background-image: url('photos/top_left.png'), url('photos/top_right.png'),
url('photos/bottom_left.png'), url('photos/bottom_right.png');
}

```



```

    background-image: url('photos/top_left.svg'), url('photos/top_right.svg'),
    url('photos/bottom_left.svg'), url('photos/bottom_right.svg');
    background-position: 1.5% 3%, 98.5% 3%, 1.5% 97%, 98.5% 97%;
    background-size: 6%, 6%, 6%, 6%;
    background-repeat: no-repeat, no-repeat, no-repeat, no-repeat;
    font-family: 'objectivitymedium', Arial, Helvetica, sans-serif;
}
#full_layout .slide[name='frame'] .header {
    font-family: 'playfair_displayregular', 'Times New Roman', Times, serif
}
#full_layout .slide[name='frame'] .header div, #full_layout .slide[name='frame']
.footer div {
    width: 7.5%;
}

#full_layout .slide[name='wavy'] {
    background-color: #ffffffe6; /* Very important -- e6*/
    background-image: url('photos/wavy.png');
    background-image: url('photos/wavy.svg');
    background-repeat: repeat;
    font-family: 'timbra_sansbold', Arial, Helvetica, sans-serif;
}
#full_layout .slide[name='wavy'] .header {
    font-family: 'life_saversbold', 'Times New Roman', Times, serif
}
#full_layout .slide[name='vabstraction'] {
    background-color: #ffffffe6; /* Very important -- e6*/
    background-image: url('photos/abstraction.jpeg');
    background-image: url('photos/abstraction.png');
    background-size: 40% 100%;
    background-position: 140% center;
    background-repeat: no-repeat;
    font-family: 'cooper_hewittbook', Arial, Helvetica, sans-serif;
}
#full_layout .slide[name='habstraction'] {
    background-color: #ffffffe6; /* Very important -- e6*/
    background-image: url('photos/abstraction.jpeg');
    background-image: url('photos/abstraction.png');
    background-size: 100% 80%;
    background-position: center 340%;
    background-repeat: no-repeat;
    font-family: 'cooper_hewittbook', Arial, Helvetica, sans-serif;
}
#full_layout .slide[name='vabstraction'] .header, #full_layout
.slide[name='habstraction'] .header {
    font-family: 'butlermedium', 'Times New Roman', Times, serif;
}
.header {
    text-align: left;
    position: absolute;

```

```

    top: 0;
    left: 0;
    width: 100%;
    height: 20%;
    font-weight: 600;
}
.header div {
    display: inline-block;
    vertical-align: middle;
    height: 100%;
    margin: 0;
    width: 1%;
}
.header h1 {
    text-align: left;
    display: inline-block;
    margin: 1%;
    vertical-align: middle;
    font-size: 5em;
}
.main {
    text-align: left;
    box-sizing: border-box;
    padding: 0 1em 0 1em;
    position: absolute;
    top: 20%;
    left: 0;
    width: 100%;
    height: 65%;
    font-size: 1.5em;
    word-break: break-all;
    margin: 0;
}
.main div {
    display: inline-block;
    position: relative;
    height: 100%;
    margin: 0;
    width: 1%;
}
#full_layout .display #slideshow .main div.l {
    float: left;
    left: 0;
}
#full_layout #slideshow .main #title {
    display: inline-block;
    margin: 1%;
    vertical-align: middle;
    font-size: 4.5em;
    text-align: left;
}

```

```

    word-break: break-all;
}
#full_layout #slideshow .main.title div {
    display: inline-block;
    vertical-align: middle;
    height: 100%;
    margin: 0;
    width: 1%;
}
#full_layout .main .content {
    display: inline;
    word-break: break-all;
    text-align: left;
}

.footer {
    text-align: left;
    position: absolute;
    top: 85%;
    left: 0;
    width: 100%;
    height: 15%;
    font-size: 2em;
    font-weight: 600;
}
.footer div {
    display: inline-block;
    vertical-align: middle;
    height: 100%;
    margin: 0;
    width: 1%;
}
.footer p {
    text-align: left;
    display: inline-block;
    vertical-align: middle;
}
#full_layout .left {
    text-align: left;
    float: left;
}
#full_layout .right {
    text-align: right;
    float: right;
}
#full_layout .center {
    text-align: center;
}
#full_layout .justify {
    text-align: justify;
}

```

```

}
.main .i {
  display: block;
  position: relative;
  top: 0;
  left: 0;
  float: left;
  margin: 0 1vw;
}
#full_layout .fullscreen #progress_bar {
  visibility: visible;
}
#full_layout .fullscreen #lateralmenu {
  visibility: visible;
}
#full_layout .fullscreen #lateralmenu.out {
  right: 0;
}
#full_layout .fullscreen #useful {
  visibility: hidden;
}
#full_layout .fullscreen #ch_style {
  visibility: hidden;
}
#full_layout .fullscreen #controls {
  visibility: visible;
  opacity: 0;
}
#full_layout .fullscreen #controls:hover, #full_layout .fullscreen #controls:focus,
#full_layout .fullscreen #controls:active {
  opacity: 1;
}
#full_layout .fullscreen #slideshow {
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
}
#full_layout .fullscreen #slideshow .comments {
  visibility: hidden;
}
#myCanvas {
  visibility: hidden;
  position: absolute;
  top: 0;
  left: 0;
  z-index: 10;
  margin: 0;
}
#full_layout .fullscreen #myCanvas {

```

```

    visibility: visible;
}
#full_layout #slideshow .slide.active {
    z-index: 1;
    visibility: visible;
}
#full_layout .display #larrow[aria-disabled="true"] {
    background: #8a8a8a url('photos/arrow_left.png') no-repeat center
center/50%;
    background: #8a8a8a url('photos/arrow_left.svg') no-repeat center
center/50%;
    background: url('photos/arrow_left.png') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
    background: url('photos/arrow_left.svg') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
}
#full_layout .display #larrow[aria-disabled="true"]:hover, #full_layout .display
#larrow[aria-disabled="true"]:focus {
    outline: none;
    background: #b9b9b9 url('photos/arrow_left.png') no-repeat center
center/50%;
    background: #b9b9b9 url('photos/arrow_left.svg') no-repeat center
center/50%;
    background: url('photos/arrow_left.png') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
    background: url('photos/arrow_left.svg') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
#full_layout .display #rarrow[aria-disabled="true"] {
    background: #8a8a8a url('photos/arrow_right.png') no-repeat center
center/50%;
    background: #8a8a8a url('photos/arrow_right.svg') no-repeat center
center/50%;
    background: url('photos/arrow_right.png') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
    background: url('photos/arrow_right.svg') no-repeat center center/50%,
        linear-gradient(230deg, #8a8a8a, #b6b6b6 20%, #8a8a8a);
}
#full_layout .display #rarrow[aria-disabled="true"]:hover, #full_layout .display
#rarrow[aria-disabled="true"]:focus {
    outline: none;
    background: #b9b9b9 url('photos/arrow_right.png') no-repeat center
center/50%;
    background: #b9b9b9 url('photos/arrow_right.svg') no-repeat center
center/50%;
    background: url('photos/arrow_right.png') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
    background: url('photos/arrow_right.svg') no-repeat center center/50%,
        linear-gradient(230deg, #aaaaaa, #e4e4e4 50%, #aaaaaa);
}
}

```

```

#full_layout .display #slideshow .slide .main .s8 {
  font-size: 1em;
}
#full_layout .display #slideshow .slide .main .s10 {
  font-size: 1.25em;
}
#full_layout .display #slideshow .slide .main .s12 {
  font-size: 1.5em;
}
#full_layout .display #slideshow .slide .main .s14 {
  font-size: 1.75em;
}
#full_layout .display #slideshow .slide .main .s16 {
  font-size: 2em;
}
#full_layout .display #slideshow .slide .main .s18 {
  font-size: 2.25em;
}
#full_layout .display #slideshow .slide .main .s20 {
  font-size: 2.5em;
}
#full_layout .display #slideshow .slide .main .s22 {
  font-size: 2.75em;
}
#full_layout .display #slideshow .slide .main .s24 {
  font-size: 3em;
}
#full_layout .display #slideshow .slide .main .s26 {
  font-size: 3.25em;
}
#full_layout .display #slideshow .slide .main .s28 {
  font-size: 3.5em;
}
#full_layout .display #slideshow .slide .main .s30 {
  font-size: 3.75em;
}
#full_layout.marker.fullscreen, #full_layout.pointer.fullscreen,
#full_layout.rubber.fullscreen,{
  cursor: crosshair;
}
</style>

</head>
<body>
  <div id="full_layout">
    <div class="display">
      <div id="progress_bar"></div>
      <div id="controls">
        <div id="rarrow" role="button" tabindex="0"></div>
        <div id="larrow" role="button" tabindex="0"></div>
      </div>
    </div>
  </div>

```

```

<div id="curr_slide">
  <span id="nslides"></span>
  <span id="sep">&nbsp;&nbsp;&nbsp;</span>
  <span id="tslides"></span>
</div>
<form class="no-widget" id="chSlide">
  <select id="choose_slide">
  </select>
  <div class="select" tabindex="0" role="listbox">
    <span class="value">Choose any slide</span>
    <ul class="optlist hidden" role="presentation">
    </ul>
  </div>
</form>
</div>
<canvas id="myCanvas">
</canvas>
<div id="useful">
  <a id="save" role="button" tabindex="0" href="#"
download="page.html">Save project</a>
  <div id="start" role="button" tabindex="0">Start the presentation</div>
  <div id="blank"></div>
  <form class="no-widget" id="font_size">
    <select id="font-size">
      <option value="s12">&nbsp;&nbsp;&nbsp;Choose the size of your
font&nbsp;&nbsp;&nbsp;</option>
      <option value="s8">&nbsp;&nbsp;&nbsp;8</option>
      <option value="s10">&nbsp;&nbsp;&nbsp;10</option>
      <option value="s12">&nbsp;&nbsp;&nbsp;12</option>
      <option value="s14">&nbsp;&nbsp;&nbsp;14</option>
      <option value="s16">&nbsp;&nbsp;&nbsp;16</option>
      <option value="s18">&nbsp;&nbsp;&nbsp;18</option>
      <option value="s20">&nbsp;&nbsp;&nbsp;20</option>
      <option value="s22">&nbsp;&nbsp;&nbsp;22</option>
      <option value="s24">&nbsp;&nbsp;&nbsp;24</option>
      <option value="s26">&nbsp;&nbsp;&nbsp;26</option>
      <option value="s28">&nbsp;&nbsp;&nbsp;28</option>
      <option value="s30">&nbsp;&nbsp;&nbsp;30</option>
    </select>
    <div class="select" tabindex="0" role="listbox">
      <span class="value">Choose the size of your font</span>
      <ul class="optlist hidden" role="presentation">
        <li role="option" class="option 12">Choose the size of your font</li>
        <li role="option" class="option 8">8</li>
        <li role="option" class="option 10">10</li>
        <li role="option" class="option 12">12</li>
        <li role="option" class="option 14">14</li>
        <li role="option" class="option 16">16</li>
        <li role="option" class="option 18">18</li>
        <li role="option" class="option 20">20</li>
      </ul>
    </div>
  </div>

```

```

        <li role="option" class="option 22">22</li>
        <li role="option" class="option 24">24</li>
        <li role="option" class="option">26</li>
        <li role="option" class="option 28">28</li>
        <li role="option" class="option">30</li>
    </ul>
</div>
</form>
<form class="no-widget" id="background_color">
    <select id="backg_color">
        <option value="black">&nbsp;Choose your background
color&nbsp;</option>
        <option value="black">&nbsp;Black</option>
        <option value="white">&nbsp;White</option>
        <option value="darkgrey">&nbsp;Dark grey</option>
        <option value="lightgrey">&nbsp;Light grey</option>
        <option value="yellow">&nbsp;Yellow</option>
        <option value="orange">&nbsp;Orange</option>
        <option value="red">&nbsp;Red</option>
        <option value="lightgreen">&nbsp;Light green</option>
        <option value="darkgreen">&nbsp;Dark green</option>
        <option value="lightblue">&nbsp;Light blue</option>
        <option value="darkblue">&nbsp;Dark blue</option>
        <option value="brown">&nbsp;Brown</option>
        <option value="pink">&nbsp;Pink</option>
        <option value="purple">&nbsp;Purple</option>
    </select>

    <div class="select" tabindex="0" role="listbox">
        <span class="value">Choose your background color</span>
        <ul class="optlist hidden" role="presentation">
            <li role="option" class="option black">Choose your background
color</li>
            <li role="option" class="option black">Black</li>
            <li role="option" class="option white">White</li>
            <li role="option" class="option darkgrey">Dark grey</li>
            <li role="option" class="option lightgrey">Light grey</li>
            <li role="option" class="option yellow">Yellow</li>
            <li role="option" class="option orange">Orange</li>
            <li role="option" class="option red">Red</li>
            <li role="option" class="option lightgreen">Light green</li>
            <li role="option" class="option darkgreen">Dark green</li>
            <li role="option" class="option lightblue">Light blue</li>
            <li role="option" class="option darkblue">Dark blue</li>
            <li role="option" class="option brown">Brown</li>
            <li role="option" class="option pink">Pink</li>
            <li role="option" class="option purple">Purple</li>
        </ul>
    </div>
</form>

```



```

<form class="no-widget" id="t_color">
  <select id="text_color">
    <option value="black">&nbsp;Choose your text color&nbsp;</option>
    <option value="black">&nbsp;Black</option>
    <option value="white">&nbsp;White</option>
    <option value="darkgrey">&nbsp;Dark grey</option>
    <option value="lightgrey">&nbsp;Light grey</option>
    <option value="yellow">&nbsp;Yellow</option>
    <option value="orange">&nbsp;Orange</option>
    <option value="red">&nbsp;Red</option>
    <option value="lightgreen">&nbsp;Light green</option>
    <option value="darkgreen">&nbsp;Dark green</option>
    <option value="lightblue">&nbsp;Light blue</option>
    <option value="darkblue">&nbsp;Dark blue</option>
    <option value="brown">&nbsp;Brown</option>
    <option value="pink">&nbsp;Pink</option>
    <option value="purple">&nbsp;Purple</option>
  </select>
  <div class="select" tabindex="0" role="listbox">
    <span class="value">Choose your text color</span>
    <ul class="optlist hidden" role="presentation">
      <li role="option" class="option black">Choose your text color</li>
      <li role="option" class="option black">Black</li>
      <li role="option" class="option white">White</li>
      <li role="option" class="option darkgrey">Dark grey</li>
      <li role="option" class="option lightgrey">Light grey</li>
      <li role="option" class="option yellow">Yellow</li>
      <li role="option" class="option orange">Orange</li>
      <li role="option" class="option red">Red</li>
      <li role="option" class="option lightgreen">Light green</li>
      <li role="option" class="option darkgreen">Dark green</li>
      <li role="option" class="option lightblue">Light blue</li>
      <li role="option" class="option darkblue">Dark blue</li>
      <li role="option" class="option brown">Brown</li>
      <li role="option" class="option pink">Pink</li>
      <li role="option" class="option purple">Purple</li>
    </ul>
  </div>
</form>
</div>
<aside id="lateralmenu">
  <div id="displayable" tabindex="0"></div>
  <div id="mouse" tabindex="-1"></div>
  <div id="marker" tabindex="-1"></div>
  <form class="no-widget" id="sel_color">
    <select id="selcolor">
      <option value="black">&nbsp;Choose any color&nbsp;</option>
      <option value="black">&nbsp;Black</option>
      <option value="white">&nbsp;White</option>
      <option value="darkgrey">&nbsp;Dark grey</option>

```

```

<option value="lightgrey">&nbsp;Light grey</option>
<option value="yellow">&nbsp;Yellow</option>
<option value="orange">&nbsp;Orange</option>
<option value="red">&nbsp;Red</option>
<option value="lightgreen">&nbsp;Light green</option>
<option value="darkgreen">&nbsp;Dark green</option>
<option value="lightblue">&nbsp;Light blue</option>
<option value="darkblue">&nbsp;Dark blue</option>
<option value="brown">&nbsp;Brown</option>
<option value="pink">&nbsp;Pink</option>
<option value="purple">&nbsp;Purple</option>
</select>
<div class="select" tabindex="-1" role="listbox">
  <span class="value">Choose any color</span>
  <ul class="optlist hidden" role="presentation">
    <li role="option" class="option black">Choose any color</li>
    <li role="option" class="option black">Black</li>
    <li role="option" class="option white">White</li>
    <li role="option" class="option darkgrey">Dark grey</li>
    <li role="option" class="option lightgrey">Light grey</li>
    <li role="option" class="option yellow">Yellow</li>
    <li role="option" class="option orange">Orange</li>
    <li role="option" class="option red">Red</li>
    <li role="option" class="option lightgreen">Light green</li>
    <li role="option" class="option darkgreen">Dark green</li>
    <li role="option" class="option lightblue">Light blue</li>
    <li role="option" class="option darkblue">Dark blue</li>
    <li role="option" class="option brown">Brown</li>
    <li role="option" class="option pink">Pink</li>
    <li role="option" class="option purple">Purple</li>
  </ul>
</div>
</form>
<form class="no-widget" id="sel_width">
  <select id="selwidth">
    <option value="vthin">&nbsp;Very thin</option>
    <option value="thin">&nbsp;Thin</option>
    <option value="medium">&nbsp;Medium</option>
    <option value="thick">&nbsp;Thick</option>
    <option value="vthick">&nbsp;Very thick</option>
  </select>
  <div class="select" tabindex="-1" role="listbox">
    <span class="value">Very thin</span>
    <ul class="optlist hidden" role="presentation">
      <li role="option" class="option vthin">Very thin</li>
      <li role="option" class="option thin">Thin</li>
      <li role="option" class="option medium">Medium</li>
      <li role="option" class="option thick">Thick</li>
      <li role="option" class="option vthick">Very thick</li>
    </ul>
  </div>

```

```

    </div>
</form>
<div id="rubber" tabindex="-1"></div>
<div id="pointer" tabindex="-1"></div>
<div id="exit" tabindex="-1"></div>
</aside>
<form id="ch_style" class="no-widget">
  <select id="choose_style">
    <option value="plain">Choose your favourite style</option>
    <option value="plain">Plain</option>
    <option value="classic">Classic</option>
    <option value="frame">Frame</option>
    <option value="wavy">Wavy</option>
    <option value="vabstraction">Vertical Abstraction</option>
    <option value="habstraction">Horizontal Abstraction</option>
  </select>
  <div class="select" tabindex="0" role="listbox">
    <span class="value">Choose your favourite style</span>
    <ul class="optlist hidden" role="presentation">
      <li role="option" class="option plain">Choose your favourite style</li>
      <li role="option" class="option plain">Plain</li>
      <li role="option" class="option classic">Classic</li>
      <li role="option" class="option frame">Frame</li>
      <li role="option" class="option wavy">Wavy</li>
      <li role="option" class="option vabstraction">Vertical Abstraction</li>
      <li role="option" class="option habstraction">Horizontal
Abstraction</li>
    </ul>
  </div>
</form>
<div id="slideshow">
  <div class="slide active"> <!-- This is the front slide -->
    <div class="header"><div>&nbsp;</div><div>&nbsp;</div></div>
    <div class="main title"><div>&nbsp;</div><h1 id="title">[Presentation's
title here]</h1><div>&nbsp;</div></div>
    <div class="footer"><div>&nbsp;</div><p>[Write the author name and
related here]</p><div>&nbsp;</div></div>
    <textarea class="comments" name="selfnotes" maxlength="300"
wrap="hard">Add any notes or comments.</textarea>
  </div>
  <div class="slide">
    <div class="header"><div>&nbsp;</div><h1>[Write a title for the
slide]</h1><div>&nbsp;</div></div>
    <div class="main"><img class="i" src=""><p class="content"> <!--Write
the URL of the image you want to insert (if you have one) inside src=""-->
[Write your main content here]
    </p></div>
    <div class="footer"><div>&nbsp;</div><p>[Write the footer of the
slide]</p><div>&nbsp;</div></div>

```

```

    <textarea class="comments" name="selfnotes" maxlength="300"
wrap="hard">Add any notes or comments.</textarea>
</div>
<div class="slide">
    <div class="header"><div>&nbsp;</div><h1>[Write a title for the
slide]</h1><div>&nbsp;</div></div>
    <div class="main"><img class="i" src=""><p class="content"> <!--Write
the URL of the image you want to insert (if you have one) inside src=""-->
[Write your main content here]
</p></div>
    <div class="footer"><div>&nbsp;</div><p>[Write the footer of the
slide]</p><div>&nbsp;</div></div>
    <textarea class="comments" name="selfnotes" maxlength="300"
wrap="hard">Add any notes or comments.</textarea>
</div>
<div class="slide">
    <div class="header"><div>&nbsp;</div><h1>[Write a title for the
slide]</h1><div>&nbsp;</div></div>
    <div class="main"><img class="i" src=""><p class="content"> <!--Write
the URL of the image you want to insert (if you have one) inside src=""-->
[Write your main content here]
</p></div>
    <div class="footer"><div>&nbsp;</div><p>[Write the footer of the
slide]</p><div>&nbsp;</div></div>
    <textarea class="comments" name="selfnotes" maxlength="300"
wrap="hard">Add any notes or comments.</textarea>
</div>
<div class="slide">
    <div class="header"><div>&nbsp;</div><h1>[Write a title for the
slide]</h1><div>&nbsp;</div></div>
    <div class="main"><img class="i" src=""><p class="content"> <!--Write
the URL of the image you want to insert (if you have one) inside src=""-->
[Write your main content here]
</p></div>
    <div class="footer"><div>&nbsp;</div><p>[Write the footer of the
slide]</p><div>&nbsp;</div></div>
    <textarea class="comments" name="selfnotes" maxlength="300"
wrap="hard">Add any notes or comments.</textarea>
</div>
<!-- If you want to have more slide, paste and copy the code below:
<div class="slide">
    <div class="header"><div>&nbsp;</div><h1>[Write a title for the
slide]</h1><div>&nbsp;</div></div>
    <div class="main"><img class="i" src=""><p class="content">
[Write your main content here]

</p></div>
    <div class="footer"><div>&nbsp;</div><p>[Write the footer of the
slide]</p><div>&nbsp;</div></div>

```

```

        <textarea class="comments" name="selfnotes" maxlength="300"
wrap="hard">Add any notes or comments.</textarea>
    </div>
    -->
</div>
</div>
</div>
<script>
/*global $, _ */
"use strict";
/*eslint-env browser*/
document.onkeydown = function(e) { // To activate fake buttons with
Enter/Return button after their are tabbed
    if(e.key === 'Enter') { // The Enter/Return key
        document.activeElement.click();
    }
};
//Functionality of the custom select widget
NodeList.prototype.forEach = function (callback) {
    Array.prototype.forEach.call(this, callback);
};
window.addEventListener("load", function () {
    var formList = document.querySelectorAll('form');
    formList.forEach(function (other){
        other.classList.remove("no-widget");
        other.classList.add("widget");
    });
});
function deactivateSelect(select) {
    if (!select.classList.contains('active')) return;
    var optlist = select.querySelector('.optlist');
    optlist.classList.add('hidden');
    select.classList.remove('active');
}
function activeSelect(select, selectList) {
    if (select.classList.contains('active')) return;
    selectList.forEach(deactivateSelect);
    select.classList.add('active');
};
function toggleOptList(select, show) {
    var optlist = select.querySelector('.optlist');
    optlist.classList.toggle('hidden');
}
function highlightOption(select, option) {
    var optionList = select.querySelectorAll('.option');
    optionList.forEach(function (other) {
        other.classList.remove('highlight');
    });
    option.classList.add('highlight');
};
};

```

```

function updateValue(select, index) {
var nativeWidget = select.previousElementSibling;
var value = select.querySelector('.value');
var optionList = select.querySelectorAll('.option');
optionList.forEach(function (other) {
other.setAttribute('aria-selected', 'false');
});
optionList[index].setAttribute('aria-selected', 'true');
nativeWidget.selectedIndex = index;
value.innerHTML = optionList[index].innerHTML;
highlightOption(select, optionList[index]);
};
function getIndex(select) {
var nativeWidget = select.previousElementSibling;
return nativeWidget.selectedIndex;
};
window.addEventListener('load', function () {
var selectList = document.querySelectorAll('.select');
selectList.forEach(function (select) {
var optionList = select.querySelectorAll('.option'),
selectedIndex = getIndex(select);
select.tabIndex = 0;
select.previousElementSibling.tabIndex = -1;
updateValue(select, selectedIndex);

optionList.forEach(function (option, index) {
option.addEventListener('mouseover', function () {
highlightOption(select, option);
});
option.addEventListener('click', function (event) {
updateValue(select, index);
listName(select, index);
});
});
select.addEventListener('click', function (event) {
toggleOptList(select);
});
select.addEventListener('focus', function (event) {
activeSelect(select, selectList);
});
select.addEventListener('blur', function (event) {
deactivateSelect(select);
});
select.addEventListener('keyup', function (event) {
var length = optionList.length,
index = getIndex(select);
if (event.keyCode === 40 && index < length - 1) { index++; }
if (event.keyCode === 38 && index > 0) { index--; }
updateValue(select, index);
});
});

```

```

});
});
// Functionality of the current slide and related
var slideShow = document.querySelector('#slideshow');
var startBtn = document.querySelector('#start');
var leftArrow = document.querySelector('#larrow');
var rightArrow = document.querySelector('#rarrow');
var numText = document.querySelector('#nslides');
var activeSlide = document.querySelector('.active');
var progressBar = document.querySelector('#progress_bar');
var myScreen = document.querySelector('.display');
var fullLayout = document.querySelector('#full_layout');
var numSlideTotal = [];
var selectedOption = [];
var slides = document.querySelectorAll('.slide');
var displayable = document.querySelector('#displayable');
var latMenu = document.querySelector('aside');
var exit = document.querySelector('#exit');
var pointer = document.querySelector('#pointer');
var rubber = document.querySelector('#rubber');
var marker = document.querySelector('#marker');
var mouse = document.querySelector('#mouse');
var controls = document.querySelector('#controls');
var chSlide = document.querySelector('#choose_slide');
var chSlide2 = document.querySelector('#chSlide ul');
var html = document.querySelector('html');
var title = document.querySelectorAll('h1');
var saveBtn = document.querySelector('#save');
var mainTitle = document.querySelector('title');
var rawContent = [];
var sel_color;
var sel_width;
var timeout;
var lastTap = 0;
var paintingIsRunning = false;
function calcProgress(slideNum, slideTotal) {
    var operation = (slideNum/slideTotal)*100;
    return operation;
}
function goLeft() {
    if(!(numSlide === 1)) {
        clearCanvas();
        var prevSlide = activeSlide.previousElementSibling;
        activeSlide.setAttribute('class', 'slide');
        prevSlide.setAttribute('class', 'slide active');
        activeSlide = document.querySelector('.active');
        leftArrow.removeAttribute('aria-disabled');
        rightArrow.removeAttribute('aria-disabled');
        numSlide -= 1;
        numText.textContent = numSlide;
    }
}

```

```

    } else if(numSlide = 1) {
        leftArrow.setAttribute('aria-disabled','true');
        rightArrow.removeAttribute('aria-disabled');
        numSlide = 1;
        numText.textContent = numSlide;
    }
}
function goRight() {
    if(!(numSlide === numSlideTotal[numSlideTotal.length-1])) {
        clearCanvas();
        var nextSlide = activeSlide.nextElementSibling;
        activeSlide.setAttribute('class','slide');
        nextSlide.setAttribute('class','slide active');
        activeSlide = document.querySelector('.active');
        leftArrow.removeAttribute('aria-disabled');
        rightArrow.removeAttribute('aria-disabled');
        numSlide += 1;
        numText.textContent = numSlide;
    } else if(numSlide = numSlideTotal[numSlideTotal.length-1]) {
        rightArrow.setAttribute('aria-disabled','true');
        leftArrow.removeAttribute('aria-disabled');
        numSlide = numSlideTotal[numSlideTotal.length-1];
        numText.textContent = numSlide;
    }
}
}

function getOut() {
    paintingIsRunning = false;
    clearCanvas();
    if(myScreen = document.querySelector('.fullscreen')){
        myScreen.setAttribute('class','display');
    } else {
        return;
    }
}
}

if (chSlide.childElementCount >= slideShow.childElementCount) {
    var i = chSlide.childElementCount;
    while (i > slideShow.childElementCount) {
        currSlide = i;
        var totalText = document.querySelector('#tslides');
        var lastChild = chSlide.lastElementChild;
        var lastChild2 = chSlide2.lastElementChild;
        numSlideTotal.pop(currSlide);
        totalText.textContent = slideShow.childElementCount;
        chSlide.removeChild(lastChild);
        chSlide2.removeChild(lastChild2);
        i -= 1;
    }
    for(i = 0; i < slideShow.childElementCount; i += 1) {
        var currSlide = i + 1;

```



```

    var chOptions = chSlide.querySelector('option');
    var chLis = chSlide2.querySelectorAll('li');
    chOptions.textContent = currSlide + ': ' + title[i].textContent;
    chLis[i].textContent = currSlide + ': ' + title[i].textContent;
    numSlideTotal.push(currSlide);
  }
} else if (chSlide.childElementCount < slideShow.childElementCount) {
  i = chSlide.childElementCount;
  while (i < slideShow.childElementCount) {
    currSlide = i + 1;
    chOptions = document.createElement('option');
    var chLi = document.createElement('li');
    totalText = document.querySelector('#tslides');
    chOptions.textContent = currSlide + ': ' + title[i].textContent;
    chLi.textContent = currSlide + ': ' + title[i].textContent;
    chLi.setAttribute('role', 'option');
    chLi.setAttribute('class', 'option');
    slides[i].setAttribute('id', currSlide);
    totalText.textContent = slideShow.childElementCount;
    chSlide.appendChild(chOptions);
    chSlide2.appendChild(chLi);
    i += 1;
  }
}
for(i = 0; i < slideShow.childElementCount; i += 1) {
  currSlide = i + 1;
  chOptions = chSlide.querySelector('option');
  chLis = chSlide2.querySelectorAll('li');
  chOptions.textContent = currSlide + ': ' + title[i].textContent; //something
  chLis.textContent = currSlide + ': ' + title[i].textContent;
  numSlideTotal.push(currSlide);
}
}
function goFullscreen() {
  if(myScreen = document.querySelector('.display')) {
    myScreen.setAttribute('class', 'fullscreen');
    clearCanvas();
    if(numSlide <= 2) {
      progressBar.hidden = false;
    } else if (numSlide === 1) {
      progressBar.hidden = true;
    }
  }
  progressBar.style.width = calcProgress(numSlide-
1,numSlideTotal[numSlideTotal.length-2]) + '%';
  document.onkeydown = function(e) {
    var fullscreen = fullLayout.firstChild.getAttribute('class');
    if (fullscreen === 'fullscreen') {
      if(e.key === 'Left' || e.key === 'ArrowLeft' || e.key === 'Down' || e.key ===
'ArrowDown' || e.key === 'a' || e.key === 's' || e.key === '-' || e.key === '4' ||
e.key === '2') {
        goLeft();

```

```

    } else if(e.key === 'Right' || e.key === 'ArrowRight' || e.key === 'Up' ||
e.key === 'ArrowUp' || e.key === 'w' || e.key === 'd' || e.key === '+' || e.key ===
'8' || e.key === '6' || e.key === ' ') {
    goRight();
    } else if(e.key === 'Escape') {
    getOut();
    } else if(e.key === 'Shift') {
    clearCanvas();
    } else if(e.key === 'm') {
    displayMenu();
    } else if(e.key === 'Enter') {
    document.activeElement.click();
    }
    } else {
    return;
    }
    progressBar.style.width = calcProgress(numSlide-
1,slideShow.childElementCount-1) + '%';
    };
    latMenu.setAttribute('class','');
    exit.addEventListener('touchend', function (event) {
    event.preventDefault();
    getOut();
    });
    exit.addEventListener('click', getOut);

window.addEventListener("resize", resize);
document.addEventListener("mousemove", draw);
document.addEventListener("mousedown", setPosition);
document.addEventListener("touchstart", setPosition);
document.addEventListener("mouseenter", setPosition);
marker.addEventListener('touchend', function (e) {
    e.preventDefault();
    fullLayout.setAttribute('class','marker');
    draw(e);
});
marker.addEventListener('click', function (e) {
    fullLayout.setAttribute('class','marker');
    draw(e);
});
rubber.addEventListener('touchend', function (e){
    e.preventDefault();
    fullLayout.setAttribute('class','rubber');
    draw(e);
});
rubber.addEventListener('click', function (e){
    fullLayout.setAttribute('class','rubber');
    draw(e);
});
pointer.addEventListener('touchend', function (e){

```

```

    e.preventDefault();
    fullLayout.setAttribute('class','pointer');
    draw(e);
});
pointer.addEventListener('click', function (e){
    fullLayout.setAttribute('class','pointer');
    draw(e);
});
mouse.addEventListener('touchend', function (e){
    e.preventDefault();
    fullLayout.setAttribute('class','mouse');
    draw(e);
});
mouse.addEventListener('click', function (e){
    fullLayout.setAttribute('class','mouse');
    draw(e);
});
canvas.addEventListener('touchend', function(event) {
    event.preventDefault();
    var currentTime = new Date().getTime();
    var tapLength = currentTime - lastTap;
    clearTimeout(timeout);
    if (tapLength < 300 && tapLength > 0) {
        goLeft();
        progressBar.style.width = calcProgress(numSlide-
1,slideShow.childElementCount-1) + '%';
        event.preventDefault();
    } else {
        if(paintingIsRunning) {
            return;
        } else {
            timeout = setTimeout(function() {
                clearTimeout(timeout);
                goRight();
                progressBar.style.width = calcProgress(numSlide-
1,slideShow.childElementCount-1) + '%';
            }, 300);
        }
    }
    lastTap = currentTime;
});
canvas.addEventListener('mouseup', function(event) {
    var currentTime = new Date().getTime();
    var tapLength = currentTime - lastTap;
    clearTimeout(timeout);
    if (tapLength < 300 && tapLength > 0) {
        goLeft();
        progressBar.style.width = calcProgress(numSlide-
1,slideShow.childElementCount-1) + '%';
        event.preventDefault();
    }
});

```

```

    } else {
      if(paintingIsRunning) {
        return;
      } else {
        timeout = setTimeout(function() {
          clearTimeout(timeout);
          goRight();
          progressBar.style.width = calcProgress(numSlide-
1,slideShow.childElementCount-1) + '%';
        }, 300);
      }
    }
    lastTap = currentTime;
  });
  controls.addEventListener('touchmove', function(){
    controls.setAttribute('style', 'opacity: 1;');
  }, { passive : true });
} else {
  return;
}
}
var chSlideOptions = document.querySelectorAll('#choose_slide option');
window.addEventListener('load', function () {
  saveBtn.setAttribute('href', '#');
  marginAdjust();
});
saveBtn.addEventListener('touchend', function () {
  for(var i = 0; i < slideShow.childElementCount - 1; i += 1) {
    contents = document.querySelectorAll('.main .content');
    contents[i].textContent = rawContent[i];
  };
  event.preventDefault();
  saveData();
});
saveBtn.addEventListener('click', function () {
  for(var i = 0; i < slideShow.childElementCount - 1; i += 1) {
    contents = document.querySelectorAll('.main .content');
    contents[i].textContent = rawContent[i];
  };
  saveData();
});
rightArrow.addEventListener('touchend', function (event) {
  event.preventDefault();
  goRight();
});
rightArrow.addEventListener('click', goRight);
leftArrow.addEventListener('touchend', function (event) {
  event.preventDefault();
  goLeft();
});

```

```

leftArrow.addEventListener('click', goLeft);
startBtn.addEventListener('touchend', function (event) {
    event.preventDefault();
    goFullscreen();
});
startBtn.addEventListener('click',goFullscreen);
displayable.addEventListener('touchend', function (event) {
    event.preventDefault();
    displayMenu();
});
displayable.addEventListener('click', displayMenu);
displayable.addEventListener('touchend', function (event) {
    event.preventDefault();
    focusable();
});
displayable.addEventListener('click', focusable);
// Hotkeys or useful keys
document.addEventListener('keydown', function(e) {
    if(e.key === 'F4') {
        goFullscreen();
    }
});
function saveData () {
    saveBtn.setAttribute('href', '#');
    saveBtn.href='data:text/html;charset=UTF-
8,'+encodeURIComponent(document.documentElement.outerHTML);
    saveBtn.setAttribute('download', mainTitle.textContent + '.html');
}
function displayMenu() {
    var position = latMenu.getAttribute('class');
    if(position === "" || position === undefined || position === NaN || position ===
null) {
        latMenu.setAttribute('class','out');
    } else {
        latMenu.removeAttribute('class');
    }
}
function focusable() {
    var position = latMenu.getAttribute('class');
    var menuSelect = latMenu.querySelectorAll('.select');
    if(position === "" || position === undefined || position === NaN || position ===
null) {
        mouse.setAttribute('tabindex', '-1');
        marker.setAttribute('tabindex', '-1');
        menuSelect.forEach(function(all) {
            all.setAttribute('tabindex', '-1');
        });
        rubber.setAttribute('tabindex', '-1');
        pointer.setAttribute('tabindex', '-1');
        exit.setAttribute('tabindex', '-1');
    }
}

```

```

} else {
  mouse.setAttribute('tabindex', '0');
  marker.setAttribute('tabindex', '0');
  menuSelect.forEach(function(all) {
    all.setAttribute('tabindex', '0');
  });
  rubber.setAttribute('tabindex', '0');
  pointer.setAttribute('tabindex', '0');
  exit.setAttribute('tabindex', '0');
}
}
function listName (select, index) {
  var slide = document.querySelector('.slide.active');
  var header = document.querySelector('.active .header');
  var footer = document.querySelector('.active .footer');
  var content = document.querySelector('.active .content');
  var currTitle = document.querySelector('.active h1');
  var currFoot = document.querySelector('.active .footer p');
  var nativeWidget = select.previousElementSibling;
  var optionList = nativeWidget.querySelectorAll('option');
  var parent = select.parentElement;
  var idParent = parent.getAttribute('id');
  nativeWidget.selectedIndex = index;
  var valueAttr = optionList[index].getAttribute('value');

  if(idParent === 'ch_style') {
    slide.setAttribute('name', valueAttr);
    if(!(valueAttr === 'classic')) {
      header.setAttribute('style', 'background-color: none;');
      footer.setAttribute('style', 'background-color: none;');
    }
  } else if (idParent === 'background_color') {
    var style = slide.getAttribute('name');
    if(style === 'plain' || style === '' || style === null) {
      slide.setAttribute('style', 'background-color: ' + colorPicker(valueAttr) +
'0.85;'); //colorPicker(valueAttr) + 'da';
    } else if(style === 'classic') {
      header.setAttribute('style', 'background-color: ' + colorPicker(valueAttr) +
'0.8;');
      footer.setAttribute('style', 'background-color: ' + colorPicker(valueAttr) +
'1;'); //colorPicker(valueAttr) + 'cd'; //colorPicker(valueAttr);
    }

    } else if(style === 'vabstraction' || style === 'habstraction' || style === 'wavy' ||
style === 'frame') {
      slide.setAttribute('style', 'background-color: ' + colorPicker(valueAttr) +
'0.9;'); //colorPicker(valueAttr) + 'e6';
    }
  } else if(idParent === 't_color') {
    currTitle.setAttribute('style', 'color: ' + colorPicker(valueAttr) + '1;');
    currFoot.setAttribute('style', 'color: ' + colorPicker(valueAttr) + '1;');
  }
}

```

```

    if(content) {
        content.setAttribute('style', 'color: ' + colorPicker(valueAttr) + '1);');
    //colorPicker(valueAttr);
    } else {
        return;
    }
} else if(idParent === 'font_size') {
    content.setAttribute('class', 'content ' + valueAttr);
} else if(idParent === 'sel_color') {
    sel_color = colorPicker(valueAttr);
} else if(idParent === 'sel_width') {
    sel_width = widthPicker(valueAttr);
} else if(idParent === 'chSlide') {
    changeSlide (index);
}
}
}
var numSlide = parseInt(activeSlide.getAttribute('id'), 10);
numText.textContent = numSlide;
function changeSlide (index) {
    var newSlideOpt = numSlideTotal.find(function (element) {
        return element === index + 1;
    });
    var slctdOptionStr = newSlideOpt.toString();
    var newSlide = document.getElementById(slctdOptionStr);
    activeSlide.setAttribute('class','slide');
    newSlide.setAttribute('class','slide active');
    activeSlide = document.querySelector('#slideshow .active');
    numSlide = parseInt(activeSlide.getAttribute('id'), 10);
    numText.textContent = numSlide;
    progressBar.style.width = calcProgress(numSlide-
1,slideShow.childElementCount-1) + '%';
}
function colorPicker (color) {
    var rgbColor;
    if (color === 'white') {
        rgbColor = 'rgba(255,255,255,1);'
    } else if (color === 'darkgrey') {
        rgbColor = 'rgba(76,76,76,1);'
    } else if (color === 'lightgrey') {
        rgbColor = 'rgba(153,153,153,1);'
    } else if (color === 'yellow') {
        rgbColor = 'rgba(255,255,0,1);'
    } else if (color === 'orange') {
        rgbColor = 'rgba(255,127,0,1);'
    } else if (color === 'red') {
        rgbColor = 'rgba(255,0,0,1);'
    } else if (color === 'lightgreen') {
        rgbColor = 'rgba(0,255,0,1);'
    } else if (color === 'darkgreen') {
        rgbColor = 'rgba(0,76,0,1);'
    }
}

```

```

    } else if (color === 'lightblue') {
      rgbColor = 'rgba(0,255,255,1)';
    } else if (color === 'darkblue') {
      rgbColor = 'rgba(0,0,255,1)';
    } else if (color === 'brown') {
      rgbColor = 'rgba(94,36,0,1)';
    } else if (color === 'pink') {
      rgbColor = 'rgba(255,0,255,1)';
    } else if (color === 'purple') {
      rgbColor = 'rgba(127,0,255,1)';
    } else {
      rgbColor = 'rgba(0,0,0,1)';
    }
  }
  return rgbColor;
}
function widthPicker (width) {
  var strokeWidth;
  if (width === 'vthin') {
    strokeWidth = 1;
  } else if (width === 'medium') {
    strokeWidth = 13;
  } else if (width === 'thick') {
    strokeWidth = 19;
  } else if (width === 'vthick') {
    strokeWidth = 25;
  } else {
    strokeWidth = 7;
  }
  return strokeWidth;
}
function marginAdjust () {
  var img = document.querySelectorAll('.i');
  var main = document.querySelector('.main');
  var mainHeight = main.offsetHeight;
  var imgHeight;
  img.forEach(function (other) {
    imgHeight = other.offsetHeight;
  });
  var operation = (mainHeight - imgHeight)/2;
  img.forEach(function (other){
    var src = other.getAttribute('src');
    if(src === "" || src === undefined || src === NaN) {
      other.setAttribute('style', 'margin: 0;');
    } else {
      other.setAttribute('style','margin-top: ' + operation + 'px; margin-bottom: ' +
operation + 'px');
    }
  });
}
// The drawing app

```



```

var canvas = document.querySelector('#myCanvas');
var cWidth = canvas.width = window.innerWidth;
var cHeight = canvas.height = window.innerHeight;
var ctx = canvas.getContext('2d');
resize();
var pos = { x: 0, y: 0};
var pointLifetime = 200;
var points = [];
function resize() {
    ctx.canvas.width = window.innerWidth;
    ctx.canvas.height = window.innerHeight;
}
function setPosition(e) {
    pos.x = e.clientX;
    pos.y = e.clientY;
}
function fMarker(e){
    if (e.buttons !== 1) {
        return;
    } else {
        paintingIsRunning = true;
        var clr = sel_color; // colorPicker
        var width = sel_width;
        ctx.globalCompositeOperation = 'source-over';
        ctx.lineWidth = width; // width picker
        ctx.lineCap = "round";
        ctx.strokeStyle = clr + '1';
        ctx.beginPath();
        ctx.moveTo(pos.x, pos.y);
        setPosition(e);
        ctx.lineTo(pos.x, pos.y);
        ctx.stroke();
    }
}
function fMouse() {
    ctx.lineWidth = 0;
    ctx.lineCap = "round";
    ctx.strokeStyle = 'none';
    ctx.beginPath();
    ctx.stroke();
}
function clearCanvas() {
    ctx.clearRect(0, 0, cWidth, cHeight);
}
function fRubber(e) {
    if (e.buttons !== 1) {
        return;
    } else {
        paintingIsRunning = true;
        var width = sel_width;

```

```

    ctx.lineWidth = width; // width picker
    ctx.lineCap = "round";
    ctx.globalCompositeOperation = 'destination-out';
    ctx.beginPath();
    ctx.moveTo(pos.x, pos.y);
    setPosition(e);
    ctx.lineTo(pos.x, pos.y);
    ctx.stroke();
  }
}
function fPointer() {
  document.addEventListener('mousemove', onPointerMove);
  pointsLife();
}
function onPointerMove(e) {
  points.push({
    time: Date.now(),
    x: e.clientX,
    y: e.clientY
  });
}
function pointsLife() {
  // Remove old points
  points = points.filter(function(point) {
    var age = Date.now() - point.time;
    return age < pointLifetime;
  });
  drawPointer();
}
function drawPointer() {
  var clr = sel_color;
  var width = sel_width;
  ctx.lineWidth = width; // width picker
  ctx.lineCap = "round";
  ctx.lineJoin = "round";
  ctx.globalCompositeOperation = 'source-over';
  ctx.clearRect(0, 0, cWidth, cHeight);
  for (var i = 1; i < points.length; i++) {
    var point = points[i];
    var previousPoint = points[i - 1];
    var age = Date.now() - point.time;
    var opacity = (pointLifetime - age) / pointLifetime;
    ctx.strokeStyle = clr + opacity + ')'; //something missing
    ctx.beginPath();
    ctx.moveTo(previousPoint.x, previousPoint.y);
    ctx.lineTo(point.x, point.y);
    ctx.stroke();
  }
}
function draw(e) {

```

```

var paintId = fullLayout.getAttribute('class');
if(paintId === '' || paintId === undefined || paintId === null || paintId === NaN ||
paintId === 'mouse') {
  cancelAnimationFrame(drawPointer);
  fMouse();
  paintingIsRunning = false;
} else if(paintId === 'marker') {
  cancelAnimationFrame(drawPointer);
  fMarker(e);
} else if(paintId === 'rubber') {
  cancelAnimationFrame(drawPointer);
  fRubber(e);
} else if(paintId === 'pointer') {
  fPointer();
  requestAnimationFrame(pointsLife);
  paintingIsRunning = false;
}
}
//CommonMark
var contents = document.querySelectorAll('.main .content');
var commonmark = window.commonmark;
var writer = new commonmark.HtmlRenderer();
var reader = new commonmark.Parser();
function parseAndRender () {
  contents.forEach(function (event) {
    rawContent.push(event.textContent);
    var parsed = reader.parse(event.textContent);
    var result = writer.render(parsed);
    event.innerHTML = result;
  });
}
window.addEventListener('load', parseAndRender);
</script>
</body>
</html>

```