
UD05 - CSS: Aspecte i Efectes

Índex

- CSS - Estètica i animació
 - Color, Contrast i espaiat
 - HSL vs RGB
 - Transparència
 - Fons
 - Sòlids, Gradients i Imatges
 - background-color
 - background-image
 - background-position
 - background-size
 - background-repeat
 - background-origin
 - background-clip
 - background-attachment
 - Combinació (Blending Modes)
 - Ombres
 - Tipografia (Web-Fonts)
 - Transicions
 - Transformacions
 - Animacions
 - Referències
- Javascript - APIs HTML5
 - Referències

CSS - Estètica i animació

Color, Contrast i espaiat

Com hem vist, tenim diverses propietats per ajustar els colors dels elements. Principalment:

- *color*: Per al color del text.
- *background-color*: Per al color del fons.
- *border-color*: Per al color del fons.

...I alguns shorthands com *background* i *border*.

Per especificar-los podem fer servir:

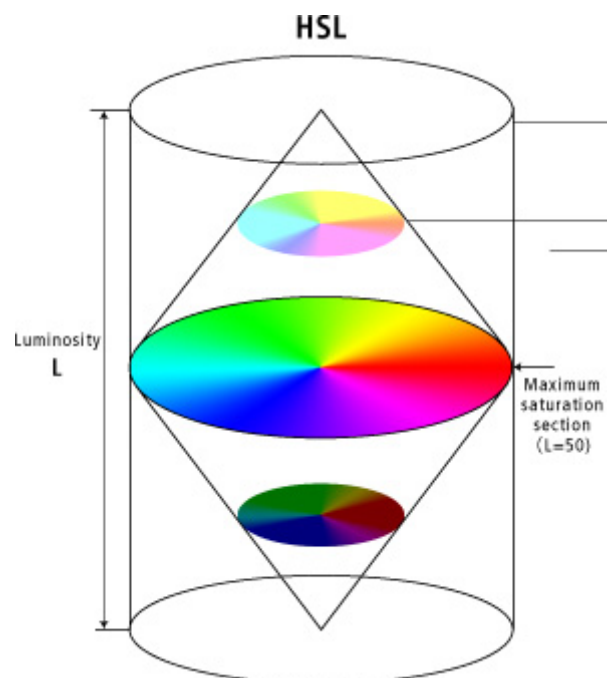
- Codi de color RGB hexadecimal: `#rrggbb`
- Color RGB amb la funció `rgb(r, g, b)`
 - on *r*, *g* i *b* son nombres entre 0 i 255 indicant la proporció de vermell (red), verd (green) i blau (blue), respectivament.
- Color RGBA (RGB + Alpha channel -transparència-) amb la funció `rgba(r, g, b, a)`.
- Color HSL (Hue Saturation Lightness) amb la funció `hsl(h, s, l)`.
- Color HSLA (HSL + Alpha channel) amb la funció `hsla(h, s, l, a)`.
- Noms de color predefinits (*red*, *green*, *blue*,...).
 - Podeu trobar la llista completa a les *referències*.

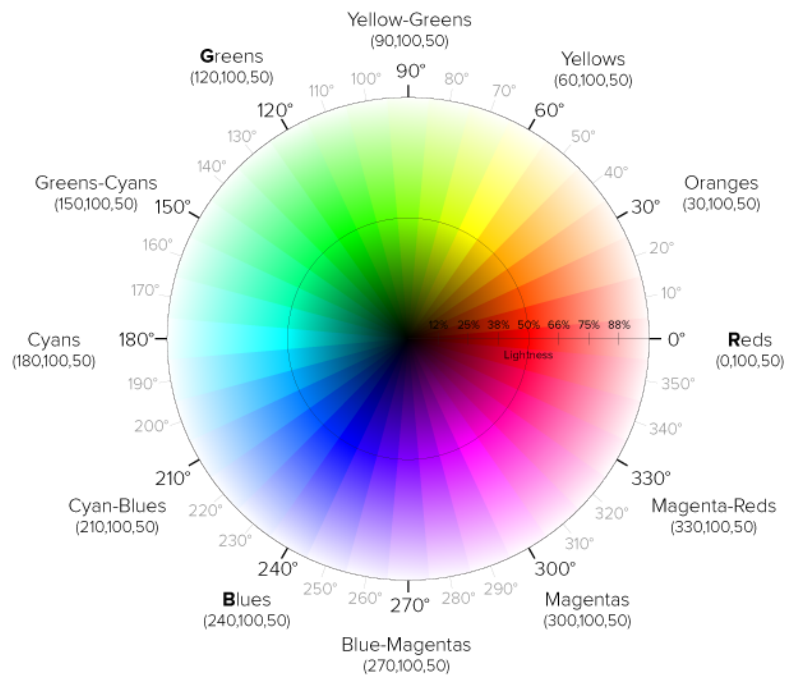
HSL vs RGB

La forma més "tradicional" de compondre els colors és mitjançant la sintaxi `#rrggbb` ajustant la proporció de cada component de color.

HSL en canvi ens permet assignar un nombre a cada tonalitat de color per a després ajustar independentment el contrast (saturació) i la lluminositat.

D'entrada aquest mètode ens pot semblar menys intuïtiu, però si fem servir una *roda de color* com la de la figura següent, resulta molt fàcil escollir la tonalitat desitjada:





L'avantatge però del sistema HSL és que, un cop hem triat un color, podem ajustar el contrast i la lluminositat del mateix **sense alterar el tò** (el color en si).

Això ens permet crear tantes variants com vulguem del mateix color (o si volem un de molt semblant alterant només lleugerament el to) només ajustant els valors.

D'aquesta manera podem crear paletes de colors més consistents i amb una aparença més professional.

Transparència

Un altre efecte que ens pot ajudar molt a millorar l'aspecte del nostre disseny (sempre i quan no n'abusem) són les transparències.

Per crear l'efecte de transparència podem fer servir dues tècniques:

- Fer servir colors amb *canal alpha*, mitjançant les funcions `rgba()` o `hsla()`.
- Ajustar la propietat *opacity* (entre 0 i 1) dels elements.

Fons

El fons d'un element comprèn, com a màxim (veure propietat *background-clip*) tot l'espai que ocupa aquest (inclosos els paddings i les voreres), **però no els marges**.

Sòlids, Gradients i Imatges

background-color

Especifica un color de fons sòlid que es mostrarà **sempre per davall** de qualsevol imatge de fons.

background-image

Especifica una imatge de fons a partir d'un fitxer o un gradient generat:

- Si és un fitxer, farem servir la funció *url()*.
 - Per exemple: `url("/images/photo.jpg")`.
- Per a generar un gradient, disposam de les següents funcions:
 - *linear-gradient()*
 - *radial-gradient()*
 - *repeating-linear-gradient()*
 - *repeating-radial-gradient()*

Totes les funcions per a generar gradients accepten una llista de colors separats per comes i altres paràmetres com la posició (en %, px, etc...) de cada color (excepte el primer que comença al principi), angle d'inici (en el cas dels radials), etc...

A les referències teniu l'enllaç a la documentació de la propietat *background-image* des de la que s'enllaça la de totes les funcions anteriors.



Un detall que no tot-hom coneix és que la propietat *background-image* accepta no una de sola, sino múltiples imatges i/o gradients separats per comes que s'aniran intercalant un darrera l'altre (el primer al front).

D'aquesta manera, podem combinar múltiples imatges amb transparència (o fent servir els *blending modes* que veurem seguidament) i/o gradients.

background-position

Estableix la posició inicial de la imatge (o gradient) de fons.

Accepta dos paràmetres: la posició horitzontal (x) i la vertical (y), sempre respecte del cantó superior esquerra, en qualsevol unitat de distància vàlida.

- El valor per defecte (*initial*) és 0% 0%.
- Si només s'especifica un valor, l'altre s'estableix automàticament a 50%.
- Podem fer servir tipus d'unitats distintes per a la coordenada horitzontal i la vertical.

- També accepta els següents keywords:
 - `left`, `center` i `right` per a la coordenada horitzontal.
 - `top`, `center` i `bottom` per a la vertical.



De la mateixa manera que amb *background-image* podem especificar múltiples imatges i/o gradients, tant amb *background-position* com amb la resta de propietats que veurem seguidament, podem especificar múltiples valors i aquests s'aplicaran a la imatge que els correspongui (seguint el mateix ordre).

background-size

Especifica el tamany de la imatge o gradient de fons.

Accepta dos valors:

- `width` (amplada)
- `height` (altura)

El valor per defecte de totes dues és `auto`, cas en el que es respectaran les dimensions originals de la imatge.

En el cas de que només una d'elles sigui `auto`, aquesta s'ajustarà mantenint les proporcions originals de la imatge.

Adicionalment, i a més de `initial` i `inherit` podem especificar els següents *keywords* especials:

- `cover`: La imatge s'escalarà, mantenint la proporció, al tamany mínim que permeti cobrir tota l'àrea del contenidor.
- `contain`: La imatge s'escalarà, mantenint la proporció, al tamany màxim possible de manera que càpiga dins el contenidor.

background-repeat

Determina si la imatge o gradient de fons s'ha de repetir o no.

Accepta els valors:

- `repeat`: Es repeteix indefinidament en les dues dimensions (és el valor per defecte).
- `repeat-x`: Es repeteix indefinidament en la dimensió horitzontal.
- `repeat-y`: Es repeteix indefinidament en la dimensió vertical.
- `no-repeat`: No es repeteix.
- `space`: Es repeteix tantes vegades com sigui possible sense tallar i l'espai sobrant es reparteix equitativament entre elles.
- `round`: Es repeteix tantes vegades sigui possible sense tallar i s'escalen segons calgui per ocupar tot l'espai.

background-origin

Determina a què es relatiu el posicionament del fons:

- `border-box`
- `padding-box` (valor inicial)

- content-box

background-clip

Especifica els límits de l'àrea de fons que s'emplena, puguent agafar els mateixos valors que *background-origin*:

- border-box
- padding-box (valor inicial)
- content-box

background-attachment

Especifica si el fons ha de romandre fix o s'ha de desplaçar junt amb el contingut.

Pot agafar, a més de *initial* i *inherit*, els següents valors:

- *scroll*: (Valor per defecte) El fons es desplaça en relació al desplaçament de la pàgina.
- *fixed*: El fons queda estàtic independentment del desplaçament de la pàgina.
- *local*: El fons es desplaça junt amb el contingut de l'element.

Combinació (Blending Modes)

Com hem vist anteriorment, podem tenir múltiples imatges (o gradients) de fons que es van col·locant un darrere de l'altre.

Per defecte, les que estan davall d'una altra, només es veuran a través de les que les cobreixen a les zones en que aquestes presentin algun grau de transparència (*canal alpha*).

A la resta de superfície queden ocultes per la imatge que es pinta al seu damunt.



Però això només és el comportament per defecte (*background-blending-mode: normal*).

La propietat *background-blending-mode* atorga, a aquelles imatges de fons per les que s'especifiqui un valor distint de *normal*, la capacitat de *mesclar-se* amb les capes inferiors.

És a dir: per a cada punt de la imatge, en comptes de veure el color de la imatge superior, veurem el color resultant de combinar el valor d'aquest amb el que té al davall aplicant una determinada fórmula matemàtica.

Aquesta fórmula matemàtica depèn del valor especificat a n'aquesta propietat, que pot ser qualsevol dels següents:

- normal
- multiply
- screen
- overlay
- darken
- lighten
- color-dodge
- saturation
- color

- luminosity

Explicar la lògica de les operacions matemàtiques darrera de cadascun d'aquests modes queda (i de molt llarg) més enllà de l'objectiu d'aquests apunts.

Però saber que existeixen i experimentar un poc amb ells, ens pot ajudar a aconseguir, amb molt poc esforç, efectes visuals molt vistosos que donaran un toc de distinció als nostres dissenys.

A les referències teniu l'enllaç a un interessant i molt ben il·lustrat article de la web *CSS-Tricks* on es mostren alguns d'ells.



Ombres

Per crear efectes d'ombra disposam de dues propietats:

- *box-shadow*: Que crea una ombra al voltant del contenidor (caixa) de l'element.
- *text-shadow*: Crea una ombra del text contingut per l'element (resseguint la forma d'aquest).

Ambdues propietats accepten (a soles) els keywords *none*, *initial* i *inherit* que ja coneixem.

Cas contrari hem de proveïr els següents valors:

- Offset horitzontal.
- Offset vertical.

...i, opcionalment també:

- "blur radius" o radi de borrositat.
- "spread radius" (només *box-shadow*) o radi de difusió.
- Color (segons l'especificació per defecte s'agafa el color del text, però alguns navegadors no mostren l'ombra si no s'especifica).

Tipografia (Web-Fonts)

Un altre aspecte que pot marcar la diferència en l'aspecte de la nostra web és la tipografia.

Com sabeu, la propietat *font* ens permet especificar el tipus de lletra d'un element (i els seus descendents, ja que és una propietat que s'hereta).

Aquesta propietat, a més, permet especificar varies fonts alternatives ja que depenem (depeníem) de que l'ordinador de l'usuari disposi d'elles instal·lades.

Així no és poc comú veure declaracions de l'estil:

```
font: Helvetica, Arial, sans-serif;
```

Però aquestes fonts, a més de presentar diferències entre elles que ens poden arribar a arruinar una maquetació, pel fet mateix de ser fonts molt comunes de les que la majoria d'usuaris disposarà, poden ser bastant *avorrides*.

Les *Web-Fonts* ens ofereixen la possibilitat de que el navegador es pugui descarregar les fonts que nosaltres escollim directament de la xarxa.

A les referències podeu trobar tres dels proveïdors de web-fonts més comuns.

Per fer-les servir només hem d'accedir als seus portals web, triar les que més ens agradin i seguir les instruccions que s'ens donen.

Finalment ens donaran un fragment de codi (podem triar si html (<link>) o CSS) que haurem d'incorporar al nostre projecte.

Transicions

Les transicions ens permeten fer que, cada cop que una propietat determinada canviï (per exemple perquè s'ha afegit o llevat una classe d'un element, el ratolí a entrat o sortit de damunt ell, etc...) el canvi es faci de forma progressiva.

No totes les propietats admeten transició. Per exemple, la propietat *display* pot valer `block` o `none`, entre d'altres, però no hi ha valors *intermigs* possibles.

En canvi, la propietat *opacity* sí que pot variar gradualment de 0 a 1, viceversa o entre qualsevol dels seus possibles valors.

Per controlar les transaccions tenim les següents propietats:

- *transition-duration*: Especifica el nombre de milisegons que tardarà la transacció en completar-se.
- *transition-property*: Especifica la propietat sobre la que volem aplicar la transacció.
- *transition-timing-function*: Especifica a funció que defineix els canvis de velocitat en el temps (`linear`, `ease-in`, `ease-out`, etc...).
- *transition-delay*: Especifica un temps d'espera abans d'iniciar la transacció (útil per a encadenament de múltiples transaccions).
- *transition*: Shorthand de les anteriors.



D'igual manera com passava amb les imatges de fons, les propietats per controlar les transaccions, permeten especificar múltiples propietats (amb la seva corresponent duració, funció, retard...) a transigir.

Només cal especificar-les una rera l'altra separades per comes igual com feiem amb les imatges de fons.

Transformacions

La propietat `transform`, com el seu nom indica, ens permet definir *transformacions* sobre l'aspecte de l'element en que s'apliquen.

Per fer-ho disposam d'una llarga llista de funcions que, de nou separades per comes, podem especificar a la propietat *transform*.

Aquestes funcions ens permeten des de moure l'element de la seva ubicació original fins a complexes operacions en 3D com modificar la perspectiva, passant per rotar-lo (en 2 i 3 dimensions), escalar-lo, etc...

A les referències teniu l'enllaç a la documentació amb la llista completa de les funcions disponibles.



La propietat *transform* accepta *transicions*. De manera que, combinant ambdues funcionalitats, podem obtenir vistoses transicions com les, de fet molt senzilles d'implementar, que heu estat veient a les diapositives d'aquest curs.

D'igual manera a quan movem (*top*, *right*, *bottom* i/o *left*) un element amb `position: relative`, quan aplicam qualsevol transformació sobre l'element, si aquest tenia reservat un espai al flux del document, (*position static* o *relative*), el conserva de manera que **la transformació no té efecte sobre els altres elements**.

Cab ressaltar també, que tots els motors de CSS fan servir el hardware d'acceleració gràfica per a implementar les transformacions i transicions. De manera que aquestes no només aporten un elevat dinamisme als nostres dissenys sinó que, a més, ho fan a un cost gairebé imperceptible.

Animacions

Finalment, les animacions, de manera semblant a les transicions, ens permeten fer que algunes propietats (transformacions incloses) variïn de forma progressiva.

Simplement les *animacions* ens permeten especificar processos més llargs, amb múltiples passos intermitjos i, opcionalment, fer que es repeteixin en el temps si ho desitjam.

Per a més informació, a les referències teniu l'enllaç a la documentació del *w3Schools*.

Referències

- Noms de Colors: https://www.w3schools.com/colors/colors_names.asp
- Propietat *background-image*: https://www.w3schools.com/cssref/pr_background-image.asp
- Blending Modes: <https://css-tricks.com/basics-css-blend-modes/>
- Web Fonts:
 - Google Fonts <https://fonts.google.com/>
 - Typekit <https://www.typekit.com/>
 - Webtype <https://www.webtype.com>
- Transicions: https://www.w3schools.com/css/css3_transitions.asp
- Transformacions: https://www.w3schools.com/cssref/css3_pr_transform.asp
- Animacions: https://www.w3schools.com/css/css3_animations.asp

Javascript - APIs HTML5



API: Una interfície de programació d'aplicacions (en anglès Application Programming Interface, API) és una interfície que especifica com diferents components de programes informàtics haurien d'interaccionar.

Dit d'una altra manera, és un conjunt d'indicacions, quant a funcions i procediments, ofert per una biblioteca informàtica o programoteca per ser utilitzat per un altre programa per interaccionar amb el programa en qüestió. O, dit encara d'una altra manera, és un conjunt de declaracions que defineix el contracte d'un component informàtic amb qui farà ús dels seus serveis.

(Font: Viquipèdia)

En HTML5 disposam de múltiples APIs a les que podem accedir, via javascript, per accedir a un ampli ventall de funcionalitats.

Algunes d'elles son:

- **Geolocation:** Ens permet obtenir la ubicació de l'usuari a través del GPS si en té i/o altres tècniques basades en Bases de Dades de xarxes WiFi i adreces IP.
- **WebSocket:** Permet la comunicació bidireccional entre el navegador i el servidor Web mitjançant una connexió TCP permanentment oberta (així és com he estat controlant les diapositives des del mòbil).
- **Local Storage:** Permet desar dades al client que podrem recuperar en futures sessions.
- **Web Workers.**
- **Drag and Drop.**
- **Application Cache.**
- **SSE:**
- ...

A les referències teniu l'enllaç a la llista completa de les APIs HTML5 existents.

Referències

- HTML5 APIs Index: <http://html5index.org/>