

# Disseny de la interfície web. Estils

Montserrat Rovira Marco

Disseny d'interfícies web



# Índex

<b>Introducció</b>	<b>5</b>
<b>Resultats d'aprenentatge</b>	<b>7</b>
<b>1 Planificació i disseny d'interfícies web. HTML5</b>	<b>9</b>
1.1 Disseny web centrat en l'usuari . . . . .	9
1.2 Elements de disseny: percepció visual . . . . .	12
1.2.1 Factors humans . . . . .	12
1.2.2 Elements de percepció visual: colors . . . . .	13
1.2.3 Elements de percepció visual: tipografia . . . . .	14
1.2.4 Elements de percepció visual: icones . . . . .	18
1.2.5 Elements de percepció visual: composició del lloc web . . . . .	19
1.3 Llenguatges i aplicacions de disseny web . . . . .	23
1.3.1 Llenguatges de desenvolupament web . . . . .	24
1.3.2 Eines de desenvolupament. Editors HTML . . . . .	26
1.3.3 Navegadors web . . . . .	30
1.4 Disseny web amb HTML5 . . . . .	32
1.4.1 Format de l'HTML5 . . . . .	33
1.4.2 Estructura del codi HTML5 . . . . .	34
1.4.3 Elements d'HTML5 . . . . .	36
1.4.4 Formularis amb HTML5 . . . . .	49
1.4.5 Visualització d'elements amb HTML5 . . . . .	56
1.4.6 Estructura semàntica de la pàgina web . . . . .	57
1.4.7 Validació HTML5 . . . . .	61
1.4.8 Seguretat HTML5 . . . . .	61
<b>2 Creació i aplicació d'estils a la interfície web. CSS3</b>	<b>63</b>
2.1 Fulls d'estils: CSS3 . . . . .	63
2.1.1 Incloure estils CSS en un document HTML . . . . .	64
2.1.2 Característiques de CSS: cascada i herència . . . . .	66
2.1.3 Format d'una regla CSS . . . . .	69
2.1.4 Selectors . . . . .	70
2.1.5 Unitats de mesura CSS . . . . .	79
2.1.6 Propietats CSS . . . . .	81
2.1.7 El model caixa . . . . .	84
2.1.8 Regles arrova . . . . .	98
2.1.9 Test i validació CSS . . . . .	99
2.2 Disseny web adaptatiu . . . . .	102
2.2.1 Tècniques per crear un disseny adaptatiu . . . . .	103
2.2.2 'Media queries' . . . . .	104
2.2.3 Desenvolupar un 'grid' fluid . . . . .	108
2.2.4 Desenvolupar un 'grid' flexible . . . . .	113
2.2.5 Continguts i imatges flexibles . . . . .	114

## Introducció

L'èxit d'un lloc web no depèn solament dels continguts que hi formen part, sinó que és fonamental que estigui ben dissenyat, amb un aspecte atractiu i amb una informació fàcilment accessible. És per això que, a l'hora de dissenyar una web, cal tenir en compte no només la usabilitat i l'accessibilitat a tota la informació, sinó també la seva estètica, és a dir, una interfície que sigui suficientment amigable per resultar agradable accedir al lloc web.

En aquesta unitat es treballen els conceptes bàsics de disseny d'una interfície web, des del que són els elements bàsics que conformen la web, com llenguatges d'implementació, fins a les eines informàtiques i els editors que faciliten el seu desenvolupament.

Així, l'apartat “Planificació i disseny d'interfícies web. HTML 5” s'inicia amb els conceptes d'interfície web i experiència d'usuari com a concepte fonamental per centrar el disseny d'un lloc web en el seu objectiu. També s'expliquen les característiques dels elements de disseny bàsics, com són el color i la tipografia. A continuació es porta a terme el desenvolupament del llenguatge de programació HTML, en la seva versió HTML5, i la implementació d'aquest per mitjà de diferents editors.

L'apartat “Creació i aplicació d'estils a la interfície web. CSS3” se centra, bàsicament, en els fulls d'estil CSS que, entre d'altres, aporten una millora significativa a la presentació de documents web amb HTML. S'hi treballen tots els conceptes bàsics de CSS3, així com també la implementació per a disseny web adaptatiu, és a dir, la visualització del web des de qualsevol dispositiu. Finalment, s'expliquen les característiques bàsiques del *framework* Bootstrap que permeten el desenvolupament d'un web amb HTML5 i CSS3.

A l'apartat “Generació de llocs web. Prototips i plantilles” s'expliquen les fases de desenvolupament d'un projecte web incident en la part de disseny, corresponent a la realització de prototipus, i també en la fase de desenvolupament, en el que correspondria a la creació d'un web des de plantilles predissenyades, així com també les eines informàtiques que es fan servir en l'actualitat. Seguidament, s'explica què són els sistemes de gestió de continguts i algunes de les seves característiques per a la creació de llocs web. Finalment, s'aborda la importància del posicionament SEO, així com algunes tècniques i pautes a tenir en compte per millorar el posicionament del web.

En finalitzar aquesta unitat hauríeu de ser capaços de crear un lloc web amb HTML5 i CSS3 fent servir el *framework* Bootstrap i seguint els estàndards generals de disseny d'interfícies web.



## 1. Planificació i disseny d'interfícies web. HTML5

Avui dia és freqüent que qualsevol usuari faci ús de dispositius connectats a Internet per portar a terme qualsevol tipus de tasques quotidianes, des de fer la compra *online* d'un determinat producte per mitjà de la seva web fins a reservar les entrades d'un espectacle o senzillament accedir a les notícies d'un diari digital.

D'igual manera, les empreses són conscients de la importància de la seva presència a Internet i, així, tenir un lloc web ben dissenyat i accessible té una influència directa no tan sols en la imatge de l'empresa, sinó en el mateix negoci.

Però de què depèn que una web d'un diari digital, per exemple, tingui més visites que altre o que una reserva es faci des d'un portal web o un altre? Òbviament, una part fonamental serà el contingut o el producte final, però en moltes ocasions també dependrà de la facilitat i amigabilitat que l'usuari trobi en accedir a la informació de la web i de la simplicitat en fer les gestions corresponents.

Per aquest motiu és de vital importància no tan sols quin és el contingut i com està organitzada la web, sinó també quin aspecte presenta a l'usuari. Aquest aspecte és el que s'anomenarà *interfície web*.

És imprescindible dissenyar i desenvolupar la interfície d'un web de tal manera que l'usuari final pugui accedir fàcilment al servei que s'ofereix i, a més, que aquest accés li sigui agradable fins al punt de tornar-hi a accedir en un futur.

Així doncs, per tal de crear la interfície d'un lloc web serà necessari disposar d'uns paràmetres estàndard per dissenyar la informació a mostrar a l'usuari i utilitzar les tècniques i els llenguatges de programació web adients, com ara l'HTML5 i els fulls d'estil CSS3.

### 1.1 Disseny web centrat en l'usuari

Quan un usuari accedeix una web, la interfície és la part que visualitza i que el permet comunicar-se amb el sistema, és a dir, la que fa que l'usuari entri en contacte, física i cognitivament amb aquest.

Una **interfície web** és el conjunt de tots aquells elements gràfics que ofereixen a l'usuari la presentació i la navegació per un lloc web amb la finalitat de portar a terme unes accions determinades.

Atès que la interacció entre l'usuari i el sistema es produeix per mitjà de la interfície, aquesta cal que reflecteixi com es fa aquesta comunicació.

La **interacció persona-ordinador (IPO)** és una disciplina dedicada a estudiar la relació interactiva entre les persones i la tecnologia. L'objectiu principal és aconseguir productes interactius fàcils d'usar, satisfactoris i que resultin realment útils.

Així doncs, es poden identificar els tres elements clau de la IPO: tecnologia, persones i disseny:

- La IPO es dedica a estudiar tota la **tecnologia** que permeti la interacció i l'ús d'aquesta amb l'usuari. Aquesta tecnologia no es limita exclusivament a ordinadors, sinó a tots els dispositius que s'utilitzen quotidianament, com ara mòbils i tauletes.
- La IPO intenta millorar la relació interactiva entre les **persones** i la tecnologia. Aquest àmbit estudia el comportament de les persones, les seves capacitats i limitacions, com reaccionen, com resolen els problemes o com prenen decisions. Encara més, també té en compte dimensió social de les persones, és a dir, com influeix l'ús de la tecnologia en el seu context sociocultural i com afecta aquesta tecnologia l'entorn de l'usuari.
- L'últim dels aspectes clau de la IPO és el **disseny**, entès com el procés d'idear solucions a problemes d'interacció, recolzant-se per a això en els estudis elaborats sobre la tecnologia i les persones. Quan es dissenya un producte interactiu el que es fa és delimitar les possibles maneres d'usar-lo, i aquest disseny és el que determina si la interacció resulta satisfactoria o no.

Paral·lelament a la IPO apareix l'enginyeria de la usabilitat (EU), la qual intenta traslladar a l'entorn professional el coneixement teòric i metodològic de la IPO. Per a alguns autors, IPO i EU representen dues cares de la mateixa moneda.

A diferència de la IPO, l'EU s'ocupa menys de saber per què funciona un disseny i es focalitza en l'obtenció de resultats, és a dir, a demostrar si un disseny realment funciona.

El concepte d'**usabilitat** es refereix a la facilitat d'ús d'una aplicació interactiva o, com defineix l'estàndard ISO 9241: “Usabilitat és la mesura en què un producte pot ser utilitzat per determinats usuaris per aconseguir uns objectius específics amb efectivitat, eficiència i satisfacció en un context d'ús definit”.

---

Si un usuari no troba com comprar un producte en un temps raonable és molt probable que intenti comprar-lo en un altre lloc.

Un lloc usable intenta buscar que l'usuari no hagi de pensar com està organitzada la informació, de tal manera que fins i tot s'optimitzen el nombre de clics a fer per completar una tasca.

Un altre aspecte a tenir en compte a l'hora de fer el disseny d'una interfície web és l'accessibilitat que ofereix. L'objectiu és facilitar l'accés del lloc web a totes les

persones, independentment del tipus de dispositiu, idioma, cultura, localització geogràfica i capacitats físiques o cognitives dels usuaris.

**L'accessibilitat** tracta el disseny d'interfícies d'usuari per tal que siguin fàcils de percebre, operatives i comprensibles per a persones amb un ventall ampli d'habilitats o amb circumstàncies, entorns i condicions diferents. En definitiva, l'accessibilitat pot ser entesa com la usabilitat per a tothom.

Actualment, en el disseny d'interfície s'utilitzen metodologies que orienten el producte als seus usuaris finals, on cada decisió presa es basa en les necessitats, els objectius, les expectatives, les motivacions i les capacitats dels usuaris.

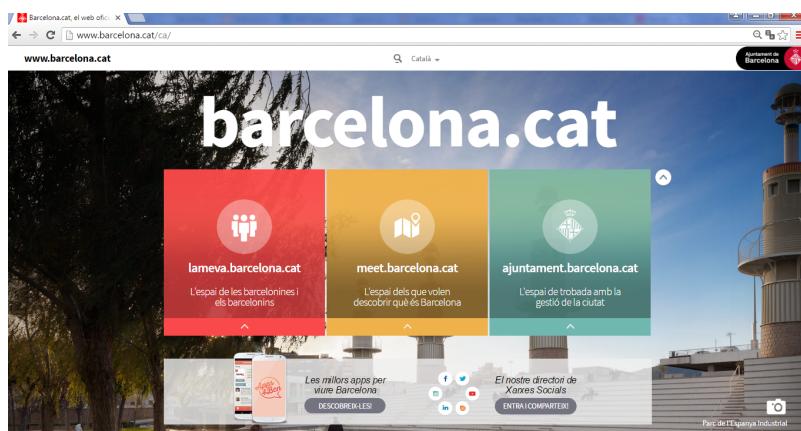
Així doncs, el disseny d'un lloc web no dependrà només de factors com el dispositiu des del qual s'hi accedeix, la usabilitat, l'accessibilitat, el disseny gràfic, la qualitat dels continguts o la utilitat, sinó que també fa referència a aquells aspectes relatius a les emocions, als sentiments i a la percepció de confiança en el producte o marca corporativa del web, entre altres aspectes.

**L'experiència d'usuari**, en anglès User Experience (UX), és una filosofia de disseny que té per objectiu la creació d'interfícies que resolen necessitats concretes dels seus usuaris finals, aconseguint així una major satisfacció d'aquests, així com també una millor experiència d'ús amb el mínim esforç.

D'aquesta manera, el disseny basat en l'experiència d'usuari utilitza metodologies de diferents camps per estudiar la interacció entre l'home i la màquina, com són la psicologia, la sociologia, l'etnologia, l'antropologia, la ciència cognitiva i l'estadística.

Vegeu en la figura 1.1 el disseny de la interfície d'un lloc web que té en compte aspectes sobre l'experiència d'usuari. S'hi pot apreciar la construcció d'interaccions digitals significatives gràcies a la utilització de colors adients, moviments que capten l'atenció però que no distreuen i que, en conseqüència, tenen com a resultat una interfície amigable. No obstant això, cal destacar que aquesta percepció d'amigabilitat té una part subjectiva.

**FIGURA 1.1.** Disseny web



Disseny de la interfície de la pàgina web de l'Ajuntament de Barcelona

### El concepte de disseny

Dissenyar és molt més que simplement acoblar o ordenar els elements; és agregar valor i significat. És a dir, il·luminar, simplificar, clarificar, persuadir i potser, fins i tot, entretenir.

En aquest sentit, el **dissenyador web** comprèn perfectament que un lloc web no hauria d'estar fet amb l'objectiu d'"estar a Internet", sinó amb l'ànim de posar a disposició una eina que es correspon amb les necessitats de l'usuari que hi accedeix.

Sovint, alguns dissenyadors web formaran part d'equips de projecte on diferents aspectes del procés de disseny seran responsabilitat de diverses persones, però en altres ocasions el dissenyador web les haurà de cobrir totes. Per exemple: dissenyadors gràfics (logotips, dissenys i botons), especialistes de màrqueting i posicionament a Internet, redactors d'Internet i dissenyadors UX. A més a més, el dissenyador web també ha d'estar informat de les últimes tendències en usabilitat i accessibilitat.

## 1.2 Elements de disseny: percepció visual

Els diferents continguts que compondran el lloc poden estar presentats i classificats de múltiples formes. Per tant, a l'hora d'escol·lir-ne una caldrà atendre una sèrie de consideracions referents a factors productius, tecnològics, perceptuals, culturals, artístics i d'innovació, entre d'altres.

En aquest sentit, el dissenyador ha d'avaluar una sèrie d'aspectes relacionats amb factors humans, així com també ha de tractar d'aconseguir una composició equilibrada, utilitzant tècniques bàsiques sobre la utilització del color, de la tipografia i de les icones i imatges a incloure a la web.

### 1.2.1 Factors humans

Per al disseny de la interfície web és important considerar aspectes culturals i de l'entorn de l'usuari final, perquè les interfícies web s'hi adeqüin.

#### Metàfora

La funció i la utilitat principal de les metàfores és afavorir la comprensió d'una situació desconeguda sobre similituds que es poden establir respecte a una realitat coneguda.

Així doncs, per una banda, els factors culturals poden influir en la percepció que un usuari pugui tenir d'un lloc web. I per una altra, cal que el dissenyador mostri els continguts utilitzant la metàfora computacional correctament i que no indueixi a errors. Així mateix, cal que tingui en compte el que els usuaris esperen trobar en un tipus de disseny per a cada servei.

Així, pot ocórrer que una mateixa web tingui un disseny o bé un altre, en funció dels usuaris que, presumiblement, l'han de visitar. Per exemple, la biblioteca de Nova York té un disseny diferent i especial per als nens que per als adults:

- [www.nypl.org](http://www.nypl.org)
- [kids.nypl.org](http://kids.nypl.org)

I per una altra banda, a l'hora de decidir quins elements formaran part del disseny web és important pensar des d'on s'hi accedirà. És a dir, si serà des d'un ambient

públic o privat, si és possible que hi hagi molta llum o bé que sigui probable que l'usuari es connecti des d'un ambient amb soroll.

Llavors resulta important escollir una bona gamma de colors perquè sigui visible en ambients amb molta o poca llum, o decidir si cal que hi hagi àudio en la web o altres tipus d'elements interactius que poden influenciar en l'atenció de l'usuari.

Un altre aspecte important seran les possibles limitacions físiques o cognitives que l'usuari pugui tenir, i que condicionaran l'accessibilitat a la web.

### 1.2.2 Elements de percepció visual: colors

El color és un dels recursos que tenen més impacte en l'estètica i l'atractiu d'una interfície. Així doncs, és una peça clau en la percepció final de l'usuari.

No obstant això, el color no té exclusivament una funció estètica o emocional, sinó que també té una funció comunicativa important. A més, quan és usat de manera incorrecta pot ocasionar problemes greus d'usabilitat en la interfície. L'experiència humana i la cultural ens diu que s'associen certs colors a emocions o sensacions molt variades. Per exemple, si es pensa en natura segurament el primer color que ve al cap és el verd, però de fet hi hauria milers de colors i tonalitats a escollir.

A més a més, no és recomanable utilitzar un nombre excessiu de colors diferents. És aconsellable, doncs, limitar-ne el nombre als que una persona pugui processar d'un sol cop d'ull; en funció de la complexitat del disseny, aquest nombre giraria a l'entorn de cinc colors.

Per tal d'escollir els colors que poden ser més adequats per al disseny d'un lloc web es poden utilitzar algunes tècniques, com el reconeixement de colors, la combinació de diferents colors o la utilització de colors complementaris.

#### Reconeixement de colors

El color s'utilitza com un recurs per reforçar el significat de certs elements de la interfície, sobretot tenint en compte el sentit que determinades cultures associen a determinats colors. Així doncs, la recomanació lògica és no incórrer en contradiccions que puguin interferir en la interpretació intuïtiva del missatge.

La figura 1.2 representa una mala utilització del color en el disseny de botons d'una interfície gràfica:

FIGURA 1.2. Utilització del color

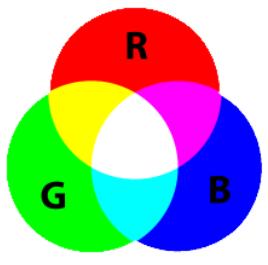
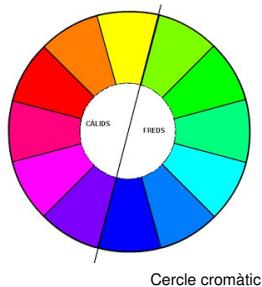


La utilització equivocada del color pot induir l'usuari a error.

## Combinacions de colors

Els diferents colors del cercle cromàtic es poden distingir entre:

- **Color primari:** són el cian, el magenta i el groc.
- **Color secundari:** al mig dels tres colors primaris tenim els tres colors secundaris, que són el vermell, el verd i el blau
- **Color terciari:** per completar els dotze colors de la roda cromàtica, els terciaris són els que es troben situats al mig d'un de primari, i els resultants de barrejar un color primari amb un de secundari adjacent.
- **Color neutre:** el blanc i el negre no se solen considerar colors i no apareixen al cercle cromàtic. El blanc és la presència de tots els colors i el negre l'absència total de color, i en combinar-se l'un i l'altre formen les diferents tonalitats de gris.



Combinació de colors primaris

La teoria sobre percepció del color i, en concret, la teoria sobre els processos oponents (Hurvich i Jamesonm 1957; Webster, 1996), diu que la visió canalitza el color en tres canals diferents: un codifica la lluminància (blanc-negre) i els altres dos, el color (vermell-verd i blau-groc). En conseqüència, trobem parells de colors que no es poden veure alhora, com el vermell amb el verd, el blau amb el groc i el negre amb el blanc (acromàtic).

Una aplicació d'aquesta teoria és la recomanació general d'evitar la combinació de colors incompatibles (vermell, verd, blau i groc), principalment dels purs o saturats. Per exemple, posar lletres en color blau sobre un fons groc en pot inhibir la visibilitat, o el text en vermell sobre un fons blau pot semblar que vibra (Cañas, Salmerón i Gámez, 2001).

Sovint ens pot interessar saber quin és un color determinat que es veu per pantalla. Tot i que existeix *software* específic per reconèixer colors, aquesta tasca també es pot dur a terme amb un editor d'imatges, com pot ser el GIMP o Adobe Photoshop, o bé també podem usar l'aplicació *online* Color Schemer: [www.colorschemer.com](http://www.colorschemer.com).

A més a més, per saber quina seria la combinació de colors que utilitza una determinada imatge es pot utilitzar l'aplicació *online* següent: [whatsitscolor.com](http://whatsitscolor.com).

Una altra eina online que permet crear una combinació de 5 colors a partir d'un de sol seleccionant diferents tipus de combinacions: anàlegs, triada, monocromàtica, etc, és Adobe Color CC: [color.adobe.com](http://color.adobe.com).

### 1.2.3 Elements de percepció visual: tipografia

#### Font tipogràfica

És un estil o una aparença d'un grup complet de caràcters, números i signes regits per unes característiques comuns.

La tipografia és l'art i la tècnica de crear i combinar tipus de fonts per comunicar un missatge. També s'ocupa de l'estudi i la classificació de diferents fonts tipogràfiques.

La tipografia té un paper crucial en el disseny, ja que la major part del contingut està expressat a través del text. Tot i això, cal tenir en compte que molts dels usuaris que accediran a la web faran un escombrat enllloc de llegir la totalitat del text.

En conseqüència, els dissenyadors han de tenir present quin tipus de tipografia és adequada per a la imatge i personalitat del lloc web, així com també per a la seva llegibilitat i intel·ligibilitat.

La **llegibilitat** es refereix a la facilitat amb què es pot llegir el text, factor que depèn de la presentació visual, és a dir, de la claredat de les lletres individuals. La **intel·ligibilitat**, en canvi, es refereix a la facilitat amb què es pot interpretar i comprendre el missatge, factor que té en compte la llargada de les paraules o de les frases.

Per escollir un tipus de tipografia o altra per al disseny del lloc web caldrà seguir unes determinades pautes estàndard.

## Família de la font

S'anomena *família de font* el conjunt de caràcters basat en una mateixa font amb algunes variacions, com poden ser el gruix o l'amplada, però sempre mantenint característiques comuns.

Actualment, els navegadors web poden utilitzar qualsevol font, tot i que molts dissenyadors continuen utilitzant les fonts més comunes. En concret, es defineix una llista de fonts que s'utilitzaran en l'ordre establert; és a dir, si la primera font no està disponible s'utilitzarà la segona, i així successivament.

### Els grups de fonts més usats són:

- Verdana, Geneva, sans serif
- Georgia, Times New Roman, Times, serif
- Courier New, Courier, monospace
- Arial, Helvetica, sans serif
- Tahoma, Geneva, sans serif
- Trebuchet MS, Arial, Helvetica, sans serif
- Arial Black, Gadget, sans serif
- Times New Roman, Times, serif
- Palatino Linotype, Book Antiqua, Palatino, serif
- Lucida Sans Unicode, Lucida Grande, sans serif
- MS Serif, New York, serif
- Comic Sans MS, cursive

### Font digital

És un conjunt de dibuixos vectorials que es poden escalar sense pèrdua de qualitat. S'emmagatzemen en arxius de tipus TrueType (TT) o Postscript TipusI (PSI).

# AaBbCc

Serif i sans serif. Els traços decoratius es veuen en vermell.

Podeu trobar catàlegs de fonts en les adreces següents:

- [fonts.google.com](https://fonts.google.com)
- [typedia.com](https://typedia.com)

Els tipus de font es poden dividir entre els de lletra *serif*, o de gràcia, i els de lletra *sans-serif*, o de pal sec. La diferència és que les fonts de gràcia presenten traços decoratius als extrems del traç de cada lletra i els de pal sec, no.

Hi ha autors que defensen que per millorar la llegibilitat és millor representar els textos digitals en fonts sense ornamentals, encara que actualment, per compensar un disseny web minimalist, hi ha una tendència creixent a utilitzar tipografies originals i cuidades com un element més i fonamental en el disseny de la pàgina web.

## Recomanacions per a la tipografia

El factor més important de la tipografia és el públic al qual va dirigit, com també ho és que aquest sigui capaç de llegir-lo perfectament sense cap inconvenient. En aquest sentit, el dissenyador ha de tenir en compte quin tipus d'usuari ho llegirà i des de quin dispositiu.

Els efectes web que es poden aplicar a les tipografies continuen sent bastant limitats. Per defecte, les característiques que es poden aplicar en el text d'una pàgina web per tal de definir l'aspecte que tindrà i millorar-ne la llegibilitat són:

- **Tipus** de font, és a dir, el seu aspecte.
- **Mida** de la font.
- **Estil**, que pot ser, normal, negreta, itàlic, subratllat...
- **Majúscula, minúscula o versaleta**.
- **Espaiat** entre lletres, paraules i paràgraf; es refereix a la distància que hi ha entre aquests.
- **Interlineat**: fa referència a la distància entre línies.
- **Alineació**: si és esquerra, centrada, dreta o justificada.
- **Contrast de color** entre el de la tipografia i el color de fons.
- **Ombrejats i gradients** de color.

A part d'escollar un tipus de font adient per al text del disseny, també es poden seguir les següents recomanacions, entre d'altres:

- El **tipus de font**: una font massa ampla o massa condensada perd llegibilitat. Es pot usar el mode condensat per a informacions secundàries que han d'ocupar poc espai.
- La **mida de la font**, que normalment s'expressa en píxels. A mode d'orientació, una mida d'11 o 12 píxels facilita la lectura de la majoria de tipografies, i gairebé mai es treballa amb quantitats inferiors a 8 o 9 píxels. Opcionalment, també es pot definir la mida com un percentatge de la tipografia mitjana.

- Amb referència a l'**estil** del text, es pot dir que l'itàlic s'utilitza per remarcar una paraula dins d'un bloc de text perquè destaci de la resta, alhora que un text escrit tot en estil itàlic resulta pesat de llegir. De la mateixa manera, un text en negreta augmenta el seu gruix i també es pot utilitzar per destacar. S'hauria d'évitar l'ús del subratllat, ja que en dificulta la lectura.
- Les **majúscules** tenen una alineació horitzontal homogènia que uniformitza les lletres. Les minúscules amb caràcters individualitzats faciliten la lectura.
- **Espaiat** entre lletres i paraules: un espaiat reduït pot percebre's com un munt de lletres amuntegades, mentre que si és massa ampli es pot percebre com un text fragmentat.
- Si l'**interlineat** és massa estret obtenim un text abarrocat amb difícil lectura. Si és massa ample s'interromp contínuament la lectura, atès que s'obliga a travessar espais en blanc contínuament. Es recomana incrementar l'interlineat un 150% més que en paper, especialment en tipografies sans-serif, que no posseeixen la guia horitzontal que formen les gràcies i que ajuden l'ull a seguir el flux del text.
- L'**alineació** a l'esquerra i la justificada són les més llegibles, encara que es preferible la primera, ja que la justificada pot provocar espais entre paules variables o massa grans. Per a texts curts es pot usar la centrada o a la dreta.
- **Contrast de color:** la utilització de colors semblants entre text i fons en dificultaran la lectura. S'ha de buscar un contrast fort, colors clars amb foscos o bé colors molt saturats amb poc saturats. Per exemple, els tipus negres sobre fons blanc aporten major llegibilitat; en canvi, el text blanc sobre fons negre ens fan perdre visibilitat.
- **Amplada de columna:** gràcies a la tendència del mercat de l'ús de pantalles panoràmiques, la majoria de webs tenen el doble d'amplada de l'abast adequat de l'ull, per la qual cosa exigeix un esforç extra per llegir el text. Una mida estàndard seria entre 12 i 15 paraules per columna. Una columna molt estreta fragmenta el text, i en una massa ampla el lector troba dificultat per trobar la línia següent.

## Altres aspectes tècnics

El text web conté moltes limitacions, com per exemple que la majoria de navegadors no poden representar text en 3D o amb un angle de rotació. Per altra banda, també presenta limitacions a l'hora d'adaptar-se automàticament al voltant de blocs rectangulars que estiguin alineats a la dreta o a l'esquerra.

En el disseny web, una solució seria representar el text com una imatge, cosa que ens permet crear qualsevol efecte que vulguem donar-li.

Tot i que, aquesta tècnica es podria utilitzar per a títols, menús o altres parts de la web, es desaconsella la seva utilització pels següents motius:



Exemples de títols fets amb imatges.  
Font: <http://goo.gl/jhSnWH/>

- Els buscadors no podrien indexar aquest text. Si el text fos curt, un petit remei seria posar un text alternatiu darrere de la imatge.
- La presentació d'aquest text esdevindria estàtica, no es podria adaptar automàticament a l'espai que hi hagi disponible.
- Si la pàgina agafa el text d'una base de dades seria molt complicat utilitzar imatges per representar aquest text.
- L'usuari no pot augmentar el text, i en alguns casos pot tenir problemes de llegibilitat.

Per tant, cal tenir en compte tots aquests aspectes a l'hora d'escollar un text amb una tipografia o una altra.

### **El color de la tipografia**

Conèixer i comprendre què transmeten els colors és important i bàsic. Així doncs, per aconseguir una bona llegibilitat haurem d'equilibrar acuradament les tres propietats del color: to, valor i intensitat, sense oblidar que el contrast entre les lletres i el seu fons haurà de ser adequat.

Les sensacions que determinats grups de colors ens poden transmetre són:

- Els colors càlids, que van del vermell al groc, són colors molt impactants, ja que destaquen molt sobre un fons. Per aquest motiu s'utilitzen per cridar l'atenció.
- Els colors freds són els verds i la gamma de blaus, i tenen la característica que són relaxants.
- El blanc, el gris i el negre no poden incloure's ni en el grup de càlids ni en el de freds.
- Els marrons, el crema i l'ocre representen enyorança i s'utilitzen per representar productes naturals i clàssics.
- Els colors primaris (magenta, cian i groc) s'utilitzen per a productes infantils, perquè aporten alegria i joventut.
- Les tipografies daurades o platejades sobre fons fosc denoten elegància i sofisticació.

És important destacar que, a primer cop d'ull, la tipografia i el color diuen més que el mateix text.

#### **1.2.4 Elements de percepció visual: icones**

Les icones són elements omnipresents en les interfícies gràfiques d'usuari i, en particular, en les interfícies web. A l'igual que el color, és un recurs que

millora l'estètica i l'atractiu visual de les interfícies. L'ús d'ícones es basa en representacions equivalents del món físic o convencions per explicar un concepte o una idea. A més, pot millorar la usabilitat del producte, però compte: quan estan mal dissenyades també poden provocar problemes greus en la interacció amb l'usuari.

En la figura 1.3 podeu veure diferents exemples d'ícones per a diferents accions quotidianes, com poden ser, imprimir, sortir, buscar, etc.

**FIGURA 1.3.** Exemples d'ícones quotidianes: imprimir, buscar, correu electrònic, sortir, informació, telèfon



Malgrat que es pensi que les ícones es reconeixen més ràpidament que el text, pot ser que no sigui així; fins i tot amb experiència d'ús, la velocitat de reconeixement d'ícones és la mateixa que la de textos.

**L'efectivitat** d'una icona es pot mesurar amb la facilitat amb què l'usuari en pot interpretar el significat.

Sovint, pot passar que usuaris no experts no utilitzin algunes de les opcions que presenta la interfície perquè no aconsegueixen interpretar correctament el significat de les ícones. Per això, en algunes ocasions és necessari etiquetar les ícones amb un text. Vegeu-ne un exemple en la figura 1.4.

**FIGURA 1.4.** Exemple d'ícones amb una etiqueta al costat



Font: vueling.com

## 1.2.5 Elements de percepció visual: composició del lloc web

A l'hora de començar una composició, el primer que s'ha de saber és l'espai de què es disposa i escollir quins elements gràfics i textuais s'usaran. És en aquest moment quan s'han de definir quines parts de la pantalla han d'estar ocupades i per quins elements; cal tenir en compte aspectes com els següents:

- Si es deixen molts espais buits la composició pot resultar descoordinada, ja que pot ser complicat establir relacions globals entre els elements o grups d'elements.

- Si el nombre d'elements és elevat podem incórrer en un disseny sobrecarregat i difícil d'entendre, en el qual no es distingeix amb claredat què és cada cosa i quin és el missatge que es vol transmetre.
- També cal tenir en compte l'agrupació lògica dels elements.

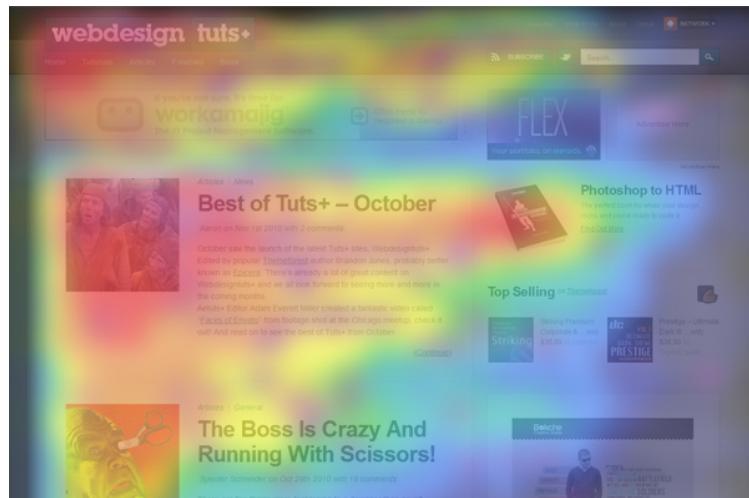
**La composició** es defineix com una distribució equilibrada de tots els elements que s'inclouen en un disseny. En aquest sentit, la composició del disseny d'un lloc web ha de ser estètica i funcional.

Un mapa de calor és una representació gràfica que dona informació sobre els focus d'atenció estimats dels usuaris.

Si seguim les directius que han analitzat la manera que té un usuari de recórrer una pàgina web, es dedueix que hi ha zones més destacades que d'altres. Per visualitzar aquesta informació podem recórrer al que s'anomena **mapa de calor**. També, la composició de disseny que un document web pot seguir el que s'anomena **patró basat en F**.

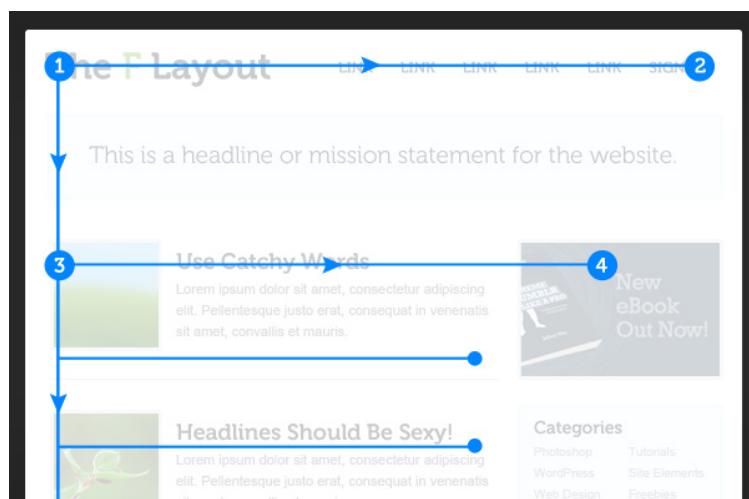
Vegeu un exemple a la figura 1.5 i la figura 1.6.

**FIGURA 1.5.** Mapa de calor



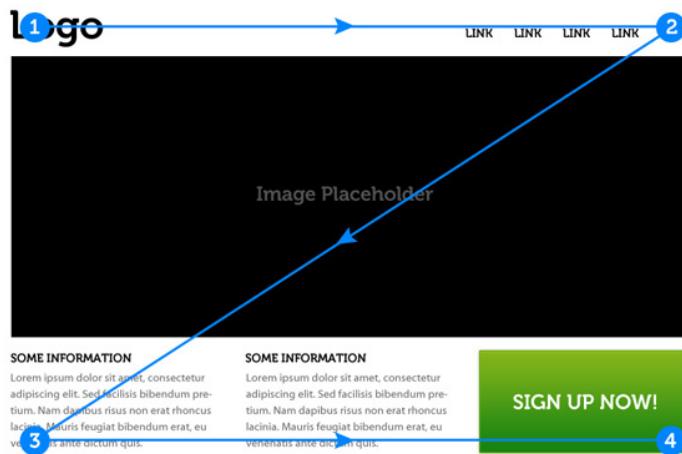
Font: <http://webdesign.tutsplus.com/es/articles/understanding-the-f-layout-in-web-design--webdesign-687>

**FIGURA 1.6.** Patró



També es pot optar per altres patrons de disseny, com el **patró basat en Z**, també utilitzat en molts llocs web. En teniu un exemple en la figura 1.7.

**FIGURA 1.7.** Patró de disseny Z



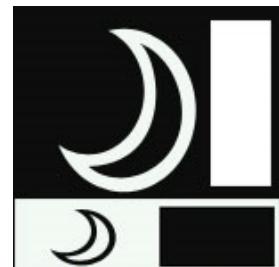
Font: <http://webdesign.tutsplus.com/articles/understanding-the-z-layout-in-web-design--webdesign-28>

Per distribuir l'espai de la nostra pàgina web de manera adequada i donar equilibri de proporcions als elements es poden utilitzar tècniques bàsiques d'equilibri visual, així com també es pot tenir en compte la **divina proporció** i la **regla dels terços**.

## Equilibri visual

Per tal d'aconseguir un disseny equilibrat es poden seguir unes normes bàsiques basades en:

- **La col·locació** dels diferents elements que conformaran el disseny. Així podem obtenir un disseny amb equilibri simètric, el qual ens transmet una sensació d'ordre, o bé un equilibri asimètric, que ens transmet agitació, tensió, dinamisme, alegria i vitalitat.
- **El contrast** de to o color dels elements del disseny. Els colors més foscos, purs i saturats tenen més pes visual que els colors secundaris o neutres, i aquests més gran que els terciaris o poc saturats. Així doncs, si es vol que un element perdi protagonisme cal disminuir la intensitat del seu to i també la seva mida, perquè així no es perdi equilibri en la composició.
- **La proporció** dels elements, de tal manera que els elements més grans, alts o amples tenen una càrrega visual superior a la dels més petits, baixos o fins, la qual cosa crea zones d'atracció més intenses.



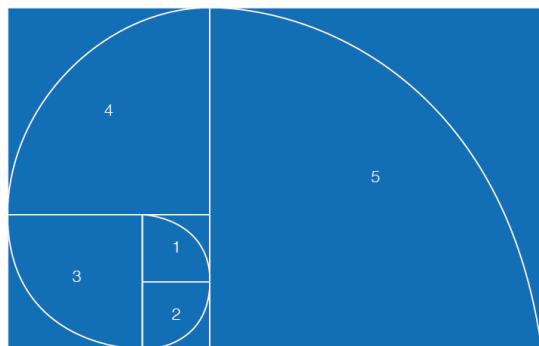
Exemple d'utilització del contrast de to

Per tant, cal definir una jerarquia visual clara, de tal manera que els elements o els continguts més rellevants han d'estar emfatitzats i han de destacar sobre la resta de la interfície. Amb això es reforça el missatge, ja que s'orienta i es facilita l'exploració visual de l'usuari.

## Divina proporció

La divina proporció està basada en la regla àuria (figura 1.8), la qual ens pot ajudar a decidir les mides dels espais que conformaran la nostra web per tal de proporcionar harmonia estructural.

**FIGURA 1.8.** Regla àuria



La **ràtio** de la divina proporció s'estableix en 1,618.

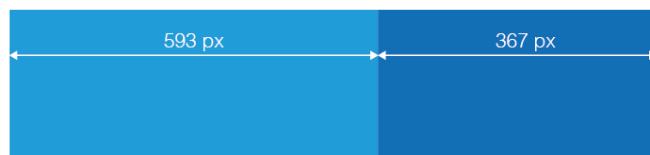
Així, per exemple, si es vol crear un disseny d'amplada fixa, 960 px, i es vol dividir l'espai en dues columnes, es deixarà una mida més gran per al contingut i una altra de més petita per a la barra lateral.

Si es vol assegurar que la ràtio entre les dues columnes és 1,618, es pot calcular l'amplada de cadascuna de la següent manera:

- Dividir la mida 960 px entre 1,618. El resultat és **593 px**, aprox.
- Restar 960 px i 593 px. El resultat és **367 px**, aprox.

Vegeu a la figura 1.9 el resultat de l'ús de la propietat de la divina proporció.

**FIGURA 1.9.** Aplicació de la divina proporció



Divisió de l'espai utilitzant la divina proporció

## Regla dels terços

Generalment, per col·locar correctament el missatge més important o la funcionalitat del lloc es pot utilitzar la **regla dels terços**.

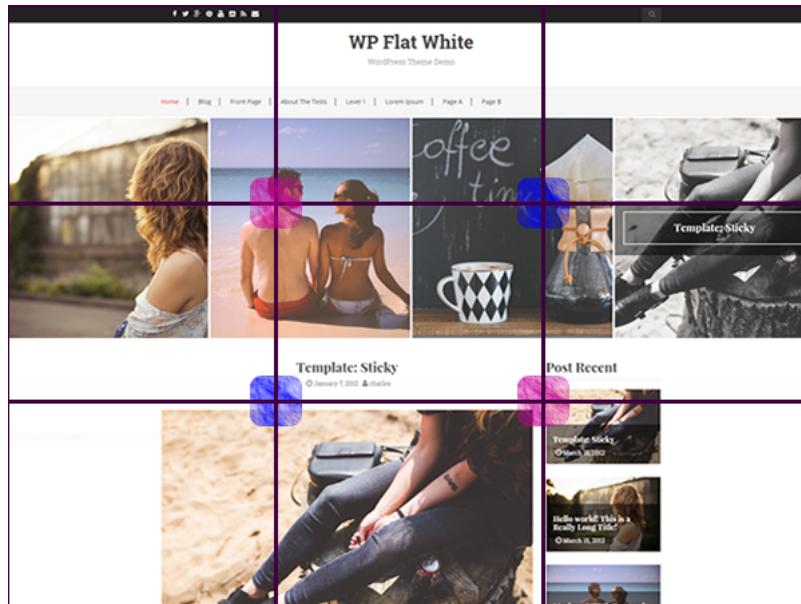
Aquesta regla s'aplica dividint la composició en 9 parts iguals, de tal manera que es divideix la composició mitjançant dues línies horitzontals i dues verticals equidistants. Els quatre punts que formen la intersecció d'aquestes línies es poden

usar per situar-hi els elements més importants. Per tant, alinear una composició d'acord amb la regla dels tres terços crea més tensió, energia i interès.

Sovint pot resultar impossible usar tots quatre punts per remarcar el contingut o les funcions més importants, però en podem emprar algun. De fet, el punt de dalt a l'esquerra és el més rellevant, ja que els usuaris acostumem a inspeccionar les pàgines web d'acord amb el patró basat en F.

Vegem en la figura 1.10 quins són els punts calents següent la regla dels terços.

**FIGURA 1.10.** Aplicació de la regla dels terços en una composició



Font: WordPress (<https://es.wordpress.org/themes/flat-white/>)

### 1.3 Llenguatges i aplicacions de disseny web

El desenvolupament d'un lloc web no és només la creació d'una pàgina web, sinó que en aquest procés hi intervenen diversos especialistes, com ara dissenyadors, programadors, comercials, *community managers*, redactors de continguts, especialistes SEO, gestors de servidors, etc.

Un **lloc web** (en anglès *Website*) és una col·lecció de pàgines web, imatges, vídeos i altres elements digitals servits per un únic domini o subdomini en el World Wide Web. Una **pàgina web** és un document HTML/XHTML que és accessible generalment mitjançant el protocol HTTP d'Internet.

La creació de pàgines web requereix conèixer diferents llenguatges de programació web com HTML5, CSS3 i Javascript, així com també coneixements d'algunes normes bàsiques d'usabilitat i accessibilitat.

#### Search Engine Optimization, SEO

Procés que té per objectiu augmentar la visibilitat d'un lloc web als motors de cerca, incrementant-ne la posició a la pàgina de resultats.

#### HyperText Transfer Protocol, HTTP

El protocol de transferència d'hipertext estableix com s'efectua l'intercanvi de documents d'hipertext i multimèdia al web. Disposa d'una variant xifrada mitjançant SSL anomenada HTTPS.

### 1.3.1 Llenguatges de desenvolupament web

Els llenguatges de desenvolupament web són:

- HTML
- CSS
- JavaScript

#### **HTML**

Des de l'aparició del World Wide Web, a principis dels anys noranta, l'HTML ha evolucionat fins a convertir-se en un llenguatge de marques relativament eficient, el qual, quan s'utilitza juntament amb elements JavaScript i CSS, pot utilitzar-se per crear llocs web i aplicacions interactives visualment sorprenents.

Per poder transferir un document entre equips, en aquest cas una pàgina web, cal codificar la informació per al seu enviament. Com que la majoria d'informació a enviar és text, s'utilitza la codificació ASCII, que relaciona cada caràcter amb una combinació numèrica en codi binari.

En un document HTML, s'envia junt amb la informació el format corresponent de la mateixa, com atributs de color, mida, alineació, etc. El llenguatge HTML proposa una sèrie de marques o etiquetes que inserides juntament amb la informació estableixen el seu format.



Logotip de W3 Consortium.

El llenguatge HTML és el resultat del treball de W3C (World Wide Web Consortium). Els seus treballs es publiquen en diferents etapes:

- Els **esborranyos**, Working Draft, es publiquen per compartir els avenços de les investigacions amb la resta.
- Les **especificacions** es publiquen quan el llenguatge està gairebé acabat i llest per ser usat.
- Les **recomanacions** es publiquen quan el llenguatge està oficialment aprovat i acabat.

#### **DHTML, Dynamic HTML**

Designa el conjunt de tècniques que permeten crear espais web interactius i animats utilitzant la combinació de llenguatge HTML estàtic, llenguatge interpretat per part del client (com el JavaScript), fulles d'estil en cascada (CSS) i el Document Object Model (DOM).

El Document Object Model (DOM) representa el document creat a partir de les etiquetes HTML i els estils CSS que el navegador mostrarà per la pantalla.

L'estàndard de W3C que actualment utilitzen els desenvolupadors per a la creació de pàgines web és el HTML5.

#### **L'evolució del llenguatge HTML**

L'origen d'HTML es remunta a finals de la dècada de 1980, quan el físic Tim Berners-Lee, investigador del CERN (Organització Europea per a la Investigació Nuclear) proposava un nou sistema "hipertext" per compartir documents.

A partir de 1996, els estàndards d'HTML els publica un altre organisme d'estandardització, el W3C (World Wide Web Consortium). La seva primera recomanació fou la versió HTML 3.2, que es va publicar el 14 de gener de 1997.

Des de la publicació d'HTML 4.01, l'activitat d'estandardització d'HTML es va parar i el W3C es va centrar en el desenvolupament de l'estàndard XHTML. Per aquest motiu, l'any 2004, les empreses Apple, Mozilla, Opera i més tard també Google van mostrar la seva preocupació per la falta d'interès del W3C en HTML i van decidir crear una nova associació anomenada **WHATWG** (Web Hypertext Association Technology Working Group).

XHTML 1.0 és una adaptació d'HTML 4.01 al llenguatge XML, així que manté quasi totes les seves etiquetes i característiques però afegeix algunes restriccions i elements propis d'XML. La versió XHTML 2.0 mai va arribar a ser una recomanació per W3C, i davant del seu fracàs, a finals del 2006, va prendre com a base les investigacions de **WHATWG** per seguir treballant amb HTML5. Així que HTML5 ja és un estàndard recomanat per W3C des del 6 d'agost de 2013. L'última recomanació es pot consultar a [www.w3.org/TR/html5](http://www.w3.org/TR/html5).

Al juliol de 2012, davant de les dificultats del treball en comú i la diferència d'objectius respecte de W3C, el WHATWG decideix **separar-se i seguir el seu propi camí de manera totalment independent**. La conseqüència és que, a partir de llavors, el WHATWG proposarà una versió "viva" de l'HTML i el W3C portarà a terme la normalització del llenguatge.

El WHATWG continua amb les seves investigacions independentment del W3C, i a l'HTML se l'anomena **HTML Living Standard** per no confondre'l amb l'**HTML5** de W3C.

Això complica la tasca de tots els desenvolupadors web, ja que han de conèixer els elements HTML5 proposats per WHATWG i comprovar si han estat validats per W3C i són reconeguts pels navegadors.

## CSS

Mentre que HTML ens permet definir l'estructura d'una pàgina web, els fulls d'estil en cascada (Cascading Style Sheets, CSS) són els que ofereixen la possibilitat de definir regles i estils que especificuen com mostrar els diferents elements HTML en diferents tipus de dispositius, com són pantalles d'ordinadors, mòbils, tauletes, etc.

CSS està dissenyat principalment per permetre la separació de contingut del document (escrit en HTML o en un llenguatge de marques similar) de la presentació del document, incloent-hi elements com la disposició, els colors i les fonts.

Aquesta separació pot millorar l'accessibilitat al contingut, proporcionar més flexibilitat i control en l'especificació de característiques de presentació i permetre que múltiples pàgines comparteixin un format comú.

Així doncs, els fulls d'estil permeten definir de manera eficient com s'ha de mostrar una pàgina web, i és un dels coneixements fonamentals que tot dissenyador web ha de conèixer per portar a terme la seva feina.

La primera versió de CSS va ser publicada per W3C a finals del 1996, i va anar guanyant popularitat fins a arribar a l'estàndard actual CSS3.

## Javascript

Javascript és un llenguatge de programació interpretat i orientat a objectes. S'utilitza al costat del client, és a dir, que s'executa en el navegador web, la qual cosa permet millors en la interfície d'usuari. El seu ús més comú és escriure funcions que s'inclouen en les pàgines web i que interactuem amb el DOM de la pàgina.

AJAX, Asynchronous JavaScript And XML, és una tècnica de desenvolupament web per crear aplicacions interactives.

Tradicionalment s'utilitzava en pàgines web HTML per fer operacions i únicament en el marc de l'aplicació client, sense accés a funcions del servidor. Avui dia és àmpliament utilitzat per enviar i rebre informació del servidor amb l'ajuda d'altres tecnologies com AJAX.

JavaScript va ser desenvolupat originàriament per Brendan Eich de Netscape el desembre de 1995, i el 1997 va ser adoptat com un estàndard ISO.

### 1.3.2 Eines de desenvolupament. Editors HTML

Per tal de desenvolupar un lloc web és útil fer servir un **editor o gestor d'HTML**. Aquest tipus de programari ofereix facilitats per a l'edició de documents HTML, i a més a més també pot proporcionar funcionalitats avançades per gestionar el codi i administrar el lloc web.

Existeixen diverses opcions per a l'edició del codi d'un lloc web: editors de text, editors en línia o bé IDE. No obstant això, les funcionalitats més importants que aquests inclouen, i que per això són adequats per al desenvolupament web, són:

- Creació de documents nous d'una manera ràpida i còmoda.
- Gestió de plantilles de diversos documents.
- Sintaxi acolorida per a una millor llegibilitat dels documents.
- Validació del codi.
- Accés ràpid al codi amb l'ajuda de marcadors.
- Configuració del sagnat i de l'estil del codi.
- Autocompleció del codi amb tancament automàtic d'etiquetes.
- Inserció d'etiquetes i atributs mitjançant auxiliars.
- Gestió de fragments de codi reusable, en anglès *snippets*.
- Gestió de projectes.
- Connectors addicionals que donen més funcionalitats al gestor.

Posteriorment, per a la visualització de la pàgina web desenvolupada s'utilitza el **navegador web**. Alhora, aquest incorpora una sèrie d'eines per als desenvolupadors que ajudaran en la depuració del codi de la pàgina.

#### Editors HTML

Avui dia hi ha una gran quantitat d'editors d'HTML que permeten el desenvolupament d'un lloc web, tots els quals tenen característiques comunes. Alguns dels

#### Editors WYSIWYG (What you see is what you get)

Permeten generar el codi dels documents HTML a partir d'una interfeïcі gràfica amb la qual es dibuixa la pàgina resultant. Són molt senzills d'usar, però poden generar molt codi no desitjat (etiquetes buides, salts de línia, etc.).

Els *snippets* són parts de codi font reusables. Així es minimitza l'ús de codi repetit que és comú en diverses funcions.

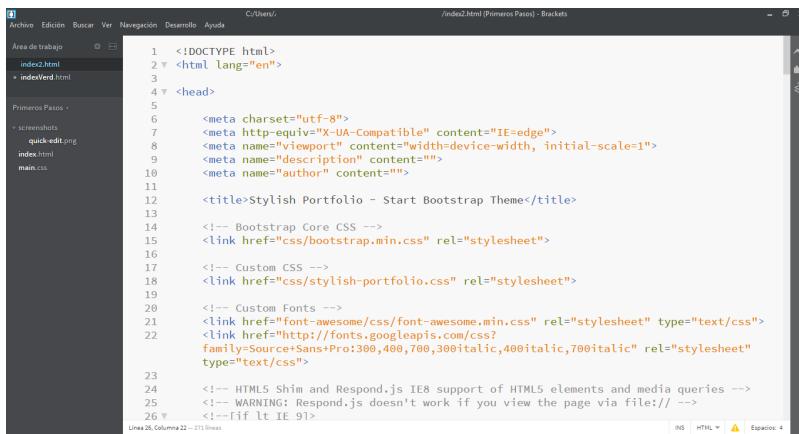
editors d'HTML amb els quals podem treballar actualment són Brackets, Sublime Text, Atom, etc.

A continuació veurem algunes de les característiques més rellevant d'aquests:

**1. Brackets:** permet veure dos o més arxius a la vegada mitjançant la funció *Inline Editors*. També incorpora la funcionalitat *Live Preview*, que permet visualitzar l'aspecte de la pàgina web a mesura que es va desenvolupant. A més a més, facilita la manera de trobar, afegir o modificar les correlacions entre documents HTML i CSS gràcies a la funcionalitat *Preprocessor Support*. És de codi obert, compatible amb GitHub, i es troba disponible per a Windows, Linux i Mac OS. El podem descarregar a: [brackets.io](http://brackets.io).

En la figura 1.11 veiem una captura de pantalla de la seva interfície.

**FIGURA 1.11.** Interfície de treball de l'editor Brackets



Logotip de l'editor Brackets

## Git

Git és un programari de sistema de control de versions pensat en l'eficiència i confiabilitat de manteniment de versions d'aplicacions amb una enorme quantitat de fitxers de codi font.

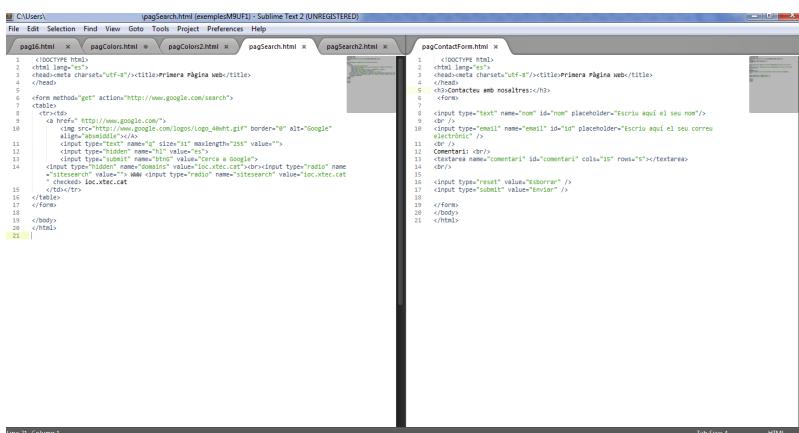
## Github

Plataforma de desenvolupament col·laboratiu per allotjar projectes utilitzant el sistema de control de versions Plugit.

**2. Sublime Text:** disposa de moltes comandes de teclat que agilitzen l'accés a les funcions i les característiques que ofereix. Així mateix, també compta amb *plugins* i complements que estan disponibles per a llenguatges de programació específics. La seva interfície és en pestanyes i mostra una vista en miniatura per visualitzar en quina part del codi s'està treballant. Es distribueix com un *software* d'avaluació, es troba disponible per a Windows, Linux i Mac OS X i és compatible amb GitHub. Es pot descarregar a: [www.sublimetext.com](http://www.sublimetext.com).

En la figura 1.12 podem veure una captura de pantalla de la seva interfície.

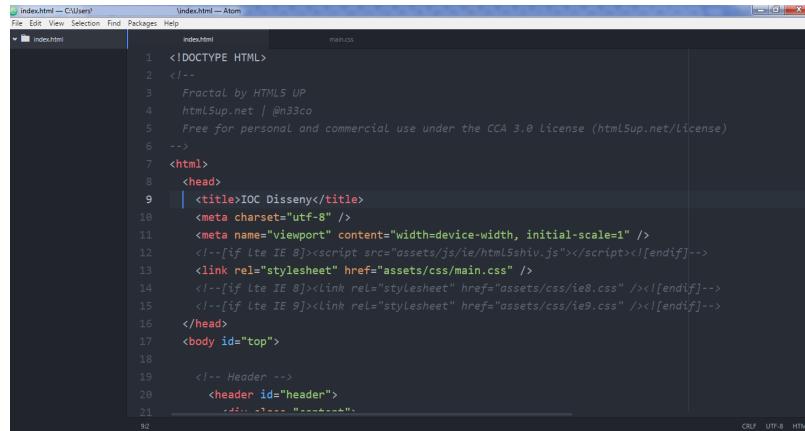
**FIGURA 1.12.** Interfície de treball de l'editor Sublime Text



**3. Atom:** proporciona les funcions més importants per al desenvolupament web, està integrat amb eines dels navegador i també alplugitHub. És de codi obert i està disponible per a Windows, Linux i Mac OS X. El podem descarregar a: [atom.io](http://atom.io).

En la figura 1.13 es pot veure una captura de pantalla de la seva interfície.

**FIGURA 1.13.** Interfície de treball de l'editor Atom



```

<!DOCTYPE HTML>
<!--
  Fractal by HTML5 UP
  html5up.net | @n33co
  Free for personal and commercial use under the CCA 3.0 License (html5up.net/license)
-->
<html>
  <head>
    <title>IOC Disseny</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <!--[if lt IE 8]><script src="assets/js/ie/html5shiv.js"></script><![endif]-->
    <link rel="stylesheet" href="assets/css/main.css" />
    <!--[if lt IE 8]><link rel="stylesheet" href="assets/css/ie8.css" /><![endif]-->
    <!--[if lt IE 9]><link rel="stylesheet" href="assets/css/ie9.css" /><![endif]-->
  </head>
  <body id="top">
    <!-- Header -->
    <header id="header">
      <!-- Header Content -->
    </header>
    <!-- Main Content -->
    <div id="main">
      <!-- Main Content -->
    </div>
    <!-- Footer -->
    <div id="footer">
      <!-- Footer Content -->
    </div>
  </body>
</html>

```

#### IDE, Integrated Development Environment

Aplicació informàtica que proporciona serveis integrals per facilitar el desenvolupament de software. Normalment incorpora un editor de codi font, eines de construcció automàtiques i un depurador.

També es pot optar per desenvolupar el lloc web en un IDE, el qual normalment ofereix un entorn de desenvolupament que agrupa diferents funcions: editor de codi, compilador, depurador, etc. S'allunya del concepte d'editor oferint una solució completa especialitzada en certs llenguatges de programació, entre els quals els llenguatges de programació web. Alguns exemples d'IDE que permeten l'elaboració de pàgines web són Netbeans, disponible a [netbeans.org](http://netbeans.org); Eclipse, disponible a [eclipse.org/downloads](http://eclipse.org/downloads), o CodeLite, disponible a [codelite.org](http://codelite.org).

En aquesta unitat es pot utilitzar qualsevol de les eines esmentades, ja que les seves característiques fan que la tasca de disseny web sigui molt més agradable i productiva.

## Editors en línia

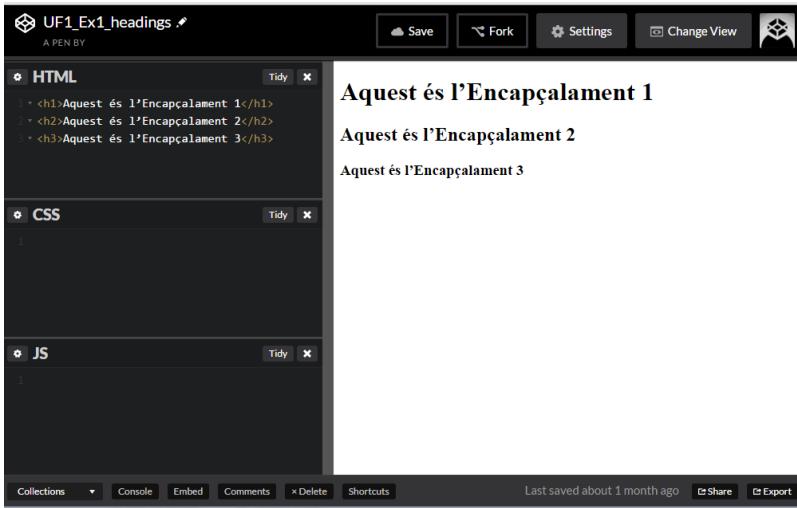
A l'hora de realitzar el disseny d' una pàgina web es pot optar per l'ús d'eines en línia que permeten escriure el codi directament des del navegador, com poden ser Codepen, Online HTML Editor o Real-Time HTML Editor, entre molts altres. Les seves característiques principals són les següents:

**1. Codepen:** aquesta eina està enfocada a crear un prototip ràpid i provar de fragments de codi. Així com, també permet que varis usuaris col·laborin en un mateix projecte i que comparteixin fragments de codi. La trobem disponible a: [codepen.io/pen](http://codepen.io/pen).

Codepen ofereix una interfície d'usuari i funcionalitats bastant intuïties i fàcils d'usar. Permet actualitzar ràpidament el codi HTML, CSS o JavaScript i obtenir resultats immediats, així com també guardar i compartir fragments de codi a través d'enllaços personalitzats, i incentiva la col·laboració oferint l'oportunitat de contribuir amb altres desenvolupadors.

En la figura 1.14 es pot apreciar un exemple d'utilització de Codepen.

**FIGURA 1.14.** Utilització de Codepen

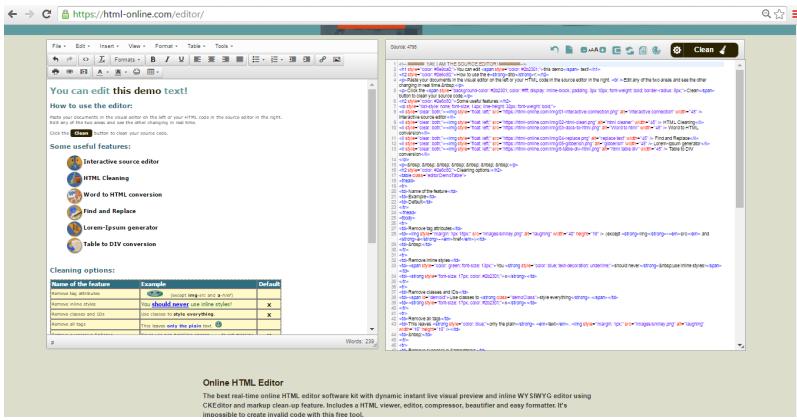


Interfície gràfica de Codepen

**2. Online HTML Editor:** aquest editor permet generar el codi dels documents HTML a partir d'una interfície gràfica. Consta de dues parts, una per crear la pàgina web mitjançant un interfície gràfica (a l'esquerra), la qual permet donar format al text, afegir imatges, etc., i una altra per escriure directament el codi HTML (a la dreta). La trobem disponible a: [html-online.com/editor](http://html-online.com/editor).

En la figura 1.14 es pot apreciar un exemple d'utilització d'aquest editor.

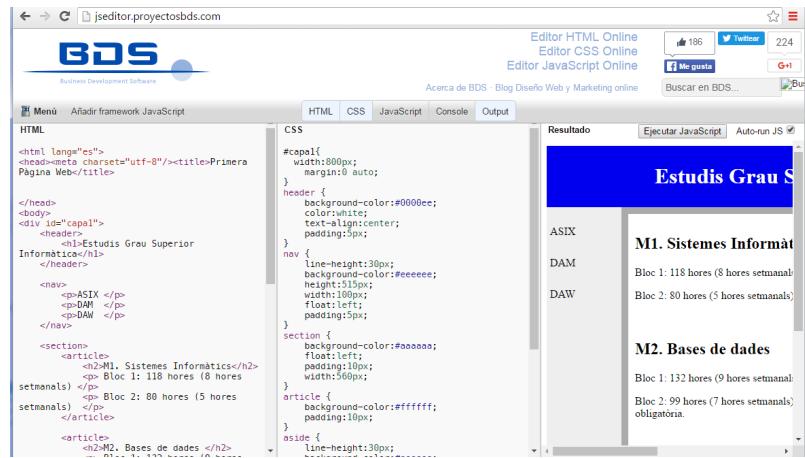
**FIGURA 1.15.** Utilització d'Online HTML Editor



Interfície gràfica de l'editor Online HTML Editor

**3. BDS Editor Online:** permet visualitzar diferents panells on es pot incloure codi HTML, Javascript o CSS. També permet una previsualització de la pàgina web. Està disponible a: [jseditor.proyectosbds.com](http://jseditor.proyectosbds.com).

En la figura 1.16 es pot apreciar un exemple d'utilització d'aquest editor.

**FIGURA 1.16.** Utilització de BDS Editor

Interfície gràfica de l'editor BDS Editor

### 1.3.3 Navegadors web



Logotips dels navegadors més importants

Per accedir a un determinat lloc web necessitem un determinat programari que ens permeti l'accés i que s'anomena navegador web.

Un **navegador web**, en anglès *browser*, és un programari informàtic que proporciona a l'usuari la visualització de documents d'hipertext juntament amb altres tipus d'arxius multimèdia, com són imatges, vídeos, sons i animacions.

Actualment, els navegadors més utilitzats són Internet Explorer, Mozilla Firefox, Google Chrome, Safari i Opera.

Cal tenir en compte que no tots els navegadors interpreten el codi HTML i CSS de la mateixa manera, i que entre ells existeixen variacions en la visualització de la pàgina web que fan que el resultat no sigui el mateix amb un que amb un altre. A més a més, algunes d'aquestes diferències són tan importants que poden fer que algunes parts de la pàgina no es vegin o no funcionin.

Algunes consideracions a tenir en compte per millorar la compatibilitat d'un lloc web amb els diferents navegadors són:

Elements com els títols o els paràgrafs poden tenir una mida, un interlineat o un marge diferent en un navegador i en un altre.

- **Validar el codi del lloc web:** és molt recomanable validar el codi del conjunt de les pàgines que conformen un lloc web, ja que, si no està ben escrit, un navegador pot interpretar el codi de manera diferent i en conseqüència el resultat pot no ser l'esperat.
- **Inicialitzar els estils CSS:** per molt que no utilitzem estils, cada element HTML té uns estils CSS per defecte, i aquests no són exactament iguals en tots els navegadors. Així, si s'inicialitzen aquests valors per defecte, es controla l'aspecte de cada element de manera més efectiva. Es poden obtenir alguns exemples de fulls d'estils per reinicialitzar els estils del navegador a: [cssreset.com](http://cssreset.com).

- **Usar tècniques suportades:** els llenguatges HTML5 i CSS3 ofereixen moltes tècniques per millorar l'aspecte de les pàgines web, però no totes estan suportades per tots els navegadors.

Les característiques dels navegadors que es tenen en compte a l'hora de comparar-los entre si són:

- ús dels recursos, memòria RAM i CPU
- temps d'arrencada, tant en fred com en calent
- temps de càrrega
- rendiment JavaScript
- rendiment gràfic
- compliment d'estàndards

Actualment, hi ha eines com Acid Test, que posen a prova els navegadors per saber si compleixen els estàndards web. Podeu trobar més informació a: [acid3.acidtests.org](http://acid3.acidtests.org).

### Eines per al desenvolupador web

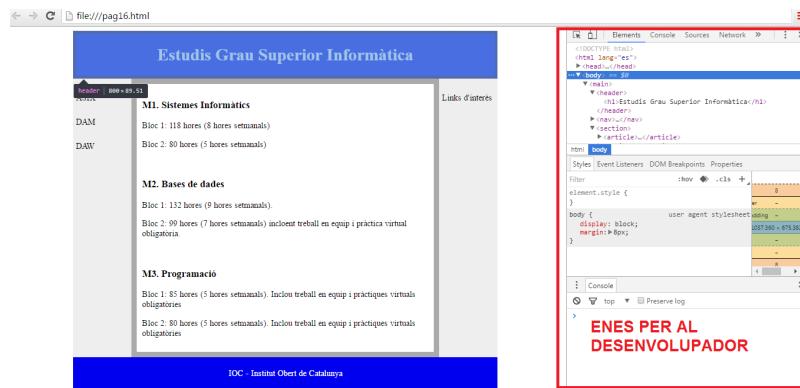
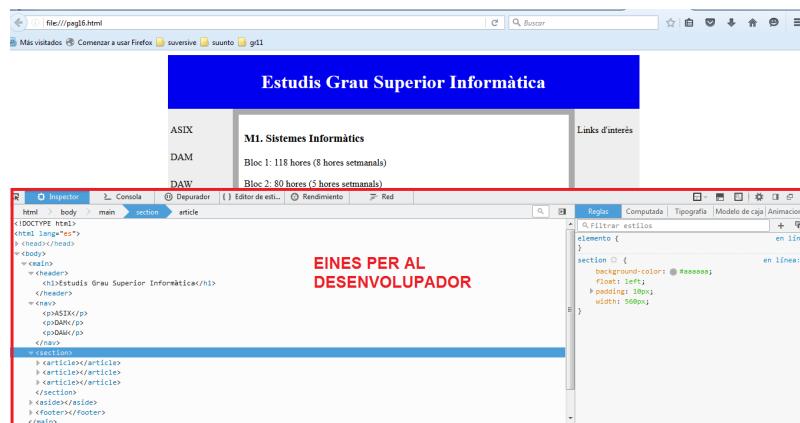
Els navegadors més importants ofereixen una sèrie d'eines que faciliten el desenvolupament d'un lloc web. D'aquesta manera, Chrome ofereix les **Google Chrome Dev Tools**, i Firefox permet incloure l'extensió **Firebug**.

Les funcions més importants que ofereixen són:

- Depurar el codi Javascript.
- Inspeccionar el DOM.
- Analitzar el temps d'execució de cada funció per optimitzar el rendiment.

Per accedir-hi se selecciona la icona situada al costat superior dret del navegador i després es fa clic a les eines per als desenvolupadors. Llavors apareix l'aplicació per inspeccionar el contingut de la pàgina web en una finestra lateral o a sota.

Vegeu en la figura 1.17 i la figura 1.18 la visualització d'aquestes eines en els navegadors Chrome i Firefox, respectivament.

**FIGURA 1.17.** Visualització de l'inspector al navegador Chrome**FIGURA 1.18.** Visualització de l'inspector al navegador Firefox

Des d'aquestes eines es pot seleccionar qualsevol element HTML o estil CSS i modificar-lo per veure com queda. Cal tenir en compte que els canvis que es facin no es guarden al document HTML.

## 1.4 Disseny web amb HTML5

El llenguatge de marques HTML5 és la versió actual del llenguatge HTML. És el llenguatge que s'utilitza per al disseny de pàgines web i fa referència a un estàndard desenvolupat per W3C.



Logotip d'HTML5

L'estàndard HTML5 conté un conjunt ampli de tecnologies que permet crear llocs web, així com aplicacions web d'envergadura. Algunes característiques d'HTML5 agrupades per la seva funció són:

- **Semàntica:** permet descriure amb major precisió quin és el seu contingut.
- **Rendiment i integració:** Proporciona una major optimització de la velocitat i un major ús del maquinari a través de la introducció d'un disseny adaptatiu.

- **CSS3:** llenguatge de fulls d'estils que ens ofereix una gran varietat d'opcions per fer dissenys més sofisticats
- **Multimèdia:** ens proporciona un excel·lent suport per utilitzar contingut multimèdia, com són àudio i vídeo nadiuament.
- **Gràfics i efectes 2D/3D:** proporciona una àmplia gamma de noves característiques que s'ocupen dels gràfics a la web, com són canvas 2D, WebGL, SVG, etc.

El que s'anomena *web semàntica* és un concepte que proposa la inclusió de metadades semàntiques a la web, perquè agents informàtics, com poden ser els navegadors o robots dels motors de cerca, puguin processar el seu contingut, raonar amb aquest, combinar-lo i fer deduccions lògiques per resoldre i donar una resposta més encertada. Així doncs, els elements que l'especificació HTML5 aporta per poder estructurar un document web amb semàntica estandarditzada són: <header>, <footer>, <section>, <aside>, <nav>, <article>, <hgroup>, <time>, etc.

#### 'Web crawler' o robot web

Programari que inspecciona pàgines web de manera metòdica i automatitzada. S'utilitza per poder referenciar les pàgines visitades a través d'un motor de cerca, com pot ser Google, Yahoo, Bing, etc.

El llenguatge HTML5 també permet realitzar dissenys web adaptatius, en anglès *Responsive Web Design*. Així doncs, disposa de tècniques perquè una web sigui visible de manera òptima en tot tipus de dispositius capaços de visualitzar pàgines web, és a dir, des de mòbils fins a ordinadors o tauletes. Per això s'utilitzen estructures amb imatges fluides i fulls d'estils CSS.

### 1.4.1 Format de l'HTML5

Els objectius de l' HTML5 són la simplicitat i la permissivitat. Així doncs, HTML5 reprèn la sintaxi permissiva d'HTML4 i fa desaparèixer les regles estrictes de sintaxi d'XHTML. No obstant això, és recomanable seguir algunes de les normes de sintaxis d'XHTML:

- Usar només minúscules.
- Usar les cometes dobles per indicar el valor dels atributs.
- Tancar totes les etiquetes.
- Sagnar el codi per augmentar la llegibilitat.

El **format** d'una etiqueta HTML és:

```

1  <NomEtiqueta atribut1="valor atribut1" ... atributN="valor
   atributN">
2   Elements afectats per l'etiqueta
3  </NomEtiqueta>
```

Cada etiqueta pot tenir una sèrie d'atributs que modifiquen el comportament de la mateixa, per exemple:

### Exemple de format d'una etiqueta

```

1  <!-- En la següent línia definim un paràgraf -->
2  <p>Aquesta és la primera pàgina web</p>
3
4  <!-- Fixem-nos en el tancament de la següent etiqueta,
   directament amb la /
5  -->
6  <br />
7
8  <!-- En la següent línia estem definint un link -->
9  <a href="https://ioc.xtec.cat/educacio/" target="_blank">Link IOC
   </a>
```

**On** a és el nom de l'etiqueta.

**On** href és l'atribut a través del qual definim on ha de portar l'enllaç.

**On** target permet especificar on s'ha d'obrir l'enllaç; \_blank significa que s'obrirà en una finestra nova.

**On** Link IOC és el text que fa de link i que es veu afectat per l'etiqueta <a>

**On** </a> serveix per tancar l'etiqueta i marcar on deixa de tenir efecte la mateixa.

Com es pot apreciar en l'exemple anterior, els comentaris d'HTML5 es posen entre els caràcters <! - i ->.

### 1.4.2 Estructura del codi HTML5

Generalment, l'estructura d'una pàgina web en HTML5 acostuma a tenir un patró semblant.

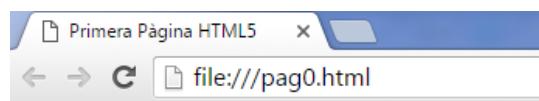
#### Estructura d'una pàgina web en HTML5

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8" />
5      <title>Primera Pàgina HTML5</title>
6  </head>
7  <body>
8      <h1> Això és HTML5 </h1>
9      <p>Aquí hi veiem el contingut de la pàgina web. </p>
10 </body>
11 </html>
```

En la figura 1.19 es pot apreciar com es visualitza una pàgina web en el navegador.

**FIGURA 1.19.** Visualització d'una pàgina web en un navegador



## Això és HTML5

Aquí hi veiem el contingut de la pàgina web.

A l'exemple anterior es pot apreciar que el document html es diferencia en dues parts: *head* i *body*.

Al **head** hi ha la informació addicional de la pàgina web, com pot ser el títol que es visualitza a la pestanya del navegador a través de l'element <title>, així com també elements <meta>, <script> i <link>, entre d'altres.

Al **body** hi ha el contingut que es visualitza de la pàgina web, el qual pot estar format per multitud d'elements html que donen format al document.

Encara que en l'especificació HTML5 s'ha fet un gran esforç per relaxar la sintaxi i que les etiquetes <html>, <head> i <body> no serien obligatòries, és del tot recomanable que hi siguin.

Vegem alguns dels apartats que es poden definir en una pàgina web:

**1. El tipus de document:** la primera línia d'un document HTML correspon a la declaració del tipus de document, el **doctype**, i serveix per indicar quina versió d'HTML s'ha utilitzat en el document.

---

DTD, Document Type Declaration, és una descripció d'estructura i sintaxi d'un document, en aquest cas HTML. S'utilitza per validar el document.

```

1  <!--Declaració del doctype en HTML5-->
2  <!DOCTYPE html>
3
4  <!-- Declaració del doctype en HTML4.01 transicional -->
5  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.
   org/TR/html4/loose.dtd">
6
7  <!-- Declaració del doctype en XHTML 1.0 estricto-->
8  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
   xhtml1/DTD/xhtml1-strict.dtd">
```

Així doncs, en l'exemple anterior es pot apreciar que un dels objectius d'HTML5 és la simplicitat i sobretot la compatibilitat en versions anteriors, com també versions futures; per tant, incloure el nombre de versió no tindria sentit.

**2. El document HTML:** l'element html indica el principi del contingut de la pàgina HTML.

```

1  <!-- Sintaxi en HTML5. -->
2  <html lang="es">
3
4  <!-- Sintaxi en XHTML 1.0 -->
5  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es" >
```

**3. La codificació de caràcters:** se sol indicar quina codificació de caràcters s'ha utilitzat en una pàgina web.

```

1  <!-- Sintaxi en HTML5 -->
2  <meta charset="UTF-8" />
3
4  <!-- Sintaxi en HTML 4.01 transicional -->
5  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >
6
7  <!-- Sintaxi en XHTML 1.0 estricta -->
8  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

**4. Estils CSS:** per indicar els estils que s'utilitzaran en la pàgina html.

```

1  <!-- Sintaxi en HTML5 -->
2  <link href="elsMeusEstils.css" rel="stylesheet" />
3
4  <!-- Sintaxi en XHTML 1.0 estricta -->
5  <link href="elsMeusEstils.css" rel="stylesheet" type="text/css" />
```

**5. Els scripts:** en una pàgina web s'hi poden incloure *scripts*.

```

1  <!-- Sintaxi en HTML5 -->
2  <script src="elsMeusScripts.js"> </script>
3
4  <!-- Sintaxi en XHTML 1.0 estricta -->
5  <script src="elsMeusScripts.js" type="text/javascript"> </script>
```

### 1.4.3 Elements d'HTML5

Podeu consultar els elements obsolets en l'adreça següent:  
[www.w3.org/TR/html5/obsolete.html](http://www.w3.org/TR/html5/obsolete.html).

Tots els elements de l'estàndard HTML5 i els seus atributs, tant els obligatoris o requerits com els que són opcionals, es poden consultar, per exemple, a la web [developer.mozilla.org/es/docs/HTML/HTML5/HTML5\\_lista\\_elementos](https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos).

A l'especificació HTML5 també es determina quins elements s'han declarat obsolets. HTML5 ha de ser compatible amb les versions anteriors; així doncs, els elements obsolets seguiran essent interpretats correctament pels navegadors.

Si es vol consultar tots elements HTML estàndards, els no estàndards i els que estan obsolets i els que no, es pot accedir, per exemple, a [developer.mozilla.org/en-US/docs/Web/HTML/Element](https://developer.mozilla.org/en-US/docs/Web/HTML/Element).

### Encapçalaments

Els encapçalaments, *headings*, són molt importants en HTML5. Es defineixen amb les etiquetes `<h1>` fins a `<h6>`, essent `<h1>` la més important, i `<h6>` definiria l'encapçalament menys important.

Automàticament, els navegadors afegirien un espai (marge) abans i després de cada encapçalament.

A l'apartat HTML de Codepen es recomana només posar el contingut que aniria a l'element `body` i, per tant, no és necessari escriure tota l'estructura de la pàgina web.

És millor utilitzar els encapçalaments per escriure títols enlloc d'utilitzar, per exemple, les etiquetes `<bold>` o `<big>`, ja que els motors de cerca usen els encapçalaments per indexar l'estruatura i el contingut del lloc web.

Vegeu el resultat de l'exemple en la figura 1.20.

#### Exemple de visualització de diferents encapçalaments

```
1 <!doctype html>
2
3 <html lang="es">
4   <head>
5     <meta charset="UTF-8" />
6     <title>Primera Pàgina Web</title>
7   </head>
8   <body>
9     <h1>Aquest és l'Encapçalament 1</h1>
10    <h2>Aquest és l'Encapçalament 2</h2>
11    <h3>Aquest és l'Encapçalament 3</h3>
12  </body>
13 </html>
```

FIGURA 1.20. Visualització de diferents encapçalaments en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/oc-daw-m09/pen/xZJXVJ?editors=1000](https://codepen.io/oc-daw-m09/pen/xZJXVJ?editors=1000).

## Paràgrafs

Els paràgrafs es defineixen amb l'element `<p>`. Realment no podem estar segurs de com es visualitzarà un paràgraf, ja que hi ha diferents dimensions de pantalla i el paràgrafs es veuran utilitzant més o menys línies.

Amb HTML no es pot canviar la sortida afegint espais en blanc o salts de línia, ja que el navegador abolirà aquests caràcters extres per mostrar la pàgina. D'aquesta manera, qualsevol nombre d'espais en blanc o de salts de línia comptarà com un sol espai en blanc.

Si es vol que es tinguin en compte els espais en blanc i/o salts de línia, podem utilitzar l'element `<pre>`.

### Exemple de visualització de paràgrafs

```

1  <!doctype html>
2
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8" />
6  <title>Primera Pàgina Web</title>
7  </head>
8  <body>
9      <h1>Aquest és l'Encapçalament 1</h1>
10     <p>
11         Aquest paràgraf,
12         conté diverses línies
13         en el codi font,
14         però el navegador
15         no les mostra.
16     </p>
17
18     <pre>
19         Aquest paràgraf,
20         conté diverses línies
21         en el codi font,
22         però el navegador
23         no les mostra.
24     </pre>
25  </body>
26  </html>

```

El resultat de l'exemple es pot visualitzar en la figura 1.21.

**FIGURA 1.21.** Visualització de paràgrafs en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/pgZWMK?editors=1000](http://codepen.io/ioc-daw-m09/pen/pgZWMK?editors=1000).

### Colors en HTML5

Hi ha diverses maneres d'especificar els diferents colors que s'utilitzin per mostrar el contingut del lloc web:

- amb el nom del color en anglès
- amb el seu codi RGB
- amb el seu codi hexadecimal

En HTML5, els colors se solen representar a través d'estils en CSS; així, una manera d'aplicar un color a un element és a través del seu atribut `style`.

Vegem amb una mica més de detall cadascuna de les opcions anteriors:

Podeu trobar una llista de colors en l'adreça següent:  
[es.wikipedia.org/wiki/Colores\\_web](https://es.wikipedia.org/wiki/Colores_web).

**1. Amb el nom del color amb anglès:** és una manera senzilla d'especificar un color, encara que aquest mètode limita la quantitat de colors representables.

El resultat del següent exemple el podeu visualitzar en la figura 1.22.

#### Exemple de visualització de colors

```

1 <p>Colors pel seu nom:</p>
2 <p style="color:red">Text de color vermell.</p>
3 <p style="color:orange">Text de color taronja.</p>
4 <p style="color:yellow">Text de color groc.</p>
5 <p style="color:cyan">Text de color cian.</p>
6 <p style="color:blue">Text de color blau.</p>
```

**FIGURA 1.22.** Visualització de colors referenciat per el seu nom en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/vLzpPz?editors=1000](https://codepen.io/loc-daw-m09/pen/vLzpPz?editors=1000).

Les sigles RGB signifiquen Red, Green, Blue.

**2. Amb el seu codi RGB:** la paleta de colors RGB prové de la combinació de tres colors primaris: vermell, verd i blau. Per tant, per especificar el codi d'un color en RGB ho farem a través de tres números referents a la quantitat de vermell, verd i blau, respectivament. D'aquesta manera podem especificar qualsevol combinació de colors.

#### Exemple de colors definits amb el seu codi RGB

```

1 <p>Colors amb el seu codi RGB:</p>
2 <p style="color:rgb(255,0,0)">Text de color vermell.</p>
3 <p style="color:rgb(255,165,0)">Text de color taronja.</p>
4 <p style="color:rgb(255,255,0)">Text de color groc.</p>
5 <p style="color:rgb(0,255,255)">Text de color cian.</p>
6 <p style="color:rgb(0,0,255)">Text de color blau.</p>
```

Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/VeGQPq](https://codepen.io/loc-daw-m09/pen/VeGQPq).

**3. Amb seu codi RGBA:** és una extensió dels valors de color RGB afegint un valor *alpha* que especifica l'opacitat de l'objecte.

Un valor de color RGBA s'especifica de la següent manera:

```
rgba(vermell, verd, blau, alfa)
```

On el paràmetre alfa és un número entre 0.0 (completament transparent) i 1.0 (totalment opac).

#### Exemple de colors RGBA

```
1 <p>Colors RGB amb opacitat:</p>
2 <p style="background-color: rgba(255, 0, 0, 0.3);">Vermell</p>
3 <p style="background-color: rgba(0, 255, 0, 0.3);">Verd</p>
4 <p style="background-color: rgba(0, 0, 255, 0.3);>Blau</p>
```

Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/RaVYya](https://codepen.io/loc-daw-m09/pen/RaVYya).

**4. Amb el seu codi hexadecimal:** el codi RGB també es pot especificar en hexadecimal, enlloc d'especificar tres números decimals. Aquest codi s'expressa amb una tira de caràcters que comença amb el caràcter # seguit de sis caràcters hexadecimals, dos per a cada color primari.

#### Exemple de colors definits en hexadecimal

```
1 <p> Colors amb el seu codi HEX:</p>
2 <p style="color:#FF0000">Text de color vermell.</p>
3 <p style="color:#FFA500">Text de color taronja.</p>
4 <p style="color:#FFFF00">Text de color groc.</p>
5 <p style="color:#00FFFF">Text de color cian.</p>
6 <p style="color:#0000FF">Text de color blau.</p>
```

Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/qbMxXM](https://codepen.io/loc-daw-m09/pen/qbMxXM).

## Estils

Per donar un format d'estil a un element HTML s'usarà l'atribut `style`. Per aplicar estils a través d'aquest atribut es farà de la següent manera:

```
style="proprietat1:valor1; ... proprietatN:valorN;"
```

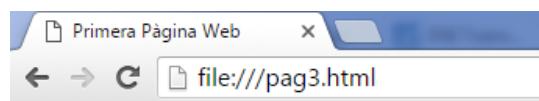
On *proprietat* fa referència a una propietat CSS i *valor* fa referència a un valor CSS.

El resultat del següent exemple el podeu veure en la figura 1.23.

#### Exemple d'atribut "style"

```
1 <h1 style="font-family:Verdana; color:blue;">Titoli</h1>
2
3 <p style="font-family:arial; color:red; font-size:200%;">
4 Paràgraf amb format
5 </p>
```

**FIGURA 1.23.** Visualització del codi anterior en un navegador



## Títol1

### Paràgraf amb format

Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/XXBzWG?editors=1000](https://codepen.io/loc-daw-m09/pen/XXBzWG?editors=1000).

#### Format del text

A banda de donar format al text a través de l'atribut `style`, el llenguatge HTML també proporciona altres elements per fer-ho, com els que es poden veure en la taula 1.1.

**TAULA 1.1.** Elements per donar format al text

Element	Descripció
<b>	Defineix text en negreta
<em>	Defineix text emfatitzat
<i>	Defineix text en itàlic, cursiva
<small>	Defineix text més petit
<strong>	Defineix text important
<sub>	Defineix text que ha d'aparèixer al subíndex
<sup>	Defineix text que ha d'aparèixer al superíndex
<ins>	Defineix text inserit, subratllat
<del>	Defineix text ratllat
<mark>	Defineix text ressaltat

#### Format negreta i itàlica

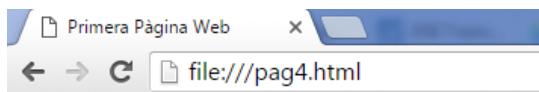
```

1  <p>Aquest text és normal.</p>
2  <p><b>Aquest text està en negreta</b>.</p>
3  <p><strong>Aquest text és important</strong>.</p>
4  <p><i>Aquest text està en itàlica</i>.</p>
5  <p><em>Aquest text està emfatitzat</em>.</p>
6

```

El resultat de l'exemple es pot visualitzar en la figura 1.24.

**FIGURA 1.24.** Visualització de formats de text en negreta i itàlica en un navegador



Aquest text és normal.

**Aquest text està en negreta.**

**Aquest text és important.**

*Aquest text està en itàlica.*

***Aquest text està emfatitzat.***

Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/LGBOJb?editors=1000](http://codepen.io/ioc-daw-m09/pen/LGBOJb?editors=1000).

Fixeu-vos en l'exemple anterior que la visualització del text a través dels elements <b> i <strong> és pràcticament igual, però si s'utilitza l'element <strong> s'hi afegeix significat semàntic al text i s'està indicant que aquest és important.

Podeu apreciar el mateix entre els elements <i> i <em>. D'aquesta manera, si s'utilitza l'element <em> també s'està afegint significat semàntic al text.

#### Altres formats

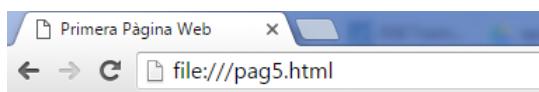
```

1  <h2>HTML format del text <small>petit</small></h2>
2
3  <h2>HTML format del text <mark>remarcat</mark></h2>
4
5  <p>Aquest text és <del>obsolet</del>.</p>
6  <p>Aquest text és <ins>obligatori</ins>.</p>
7
8  <p> Aquí posem un superíndex: E = m·c<sup>2</sup></p>
9  <p> Aquí posem un subíndex: H<sub>2</sub>O</p>
10

```

El resultat de l'exemple es pot visualitzar en la figura 1.25.

**FIGURA 1.25.** Visualització de diferents formats de text en un navegador



**HTML format del text petit**

**HTML format del text remarcat**

Aquest text és obsolet.

Aquest text és obligatori.

Aquí posem un superíndex: E = m·c<sup>2</sup>

Aquí posem un subíndex: H<sub>2</sub>O

Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/VeByKQ?editors=1000](http://codepen.io/ioc-daw-m09/pen/VeByKQ?editors=1000).

## Imatges

En HTML, les imatges es defineixen a través de l'element `<img>`.

### Exemple d'imatges

```
1 
```

On a l'atribut `src` hi especificuem l'URL o la ruta de la imatge.

On a l'atribut `alt` hi especificuem un text alternatiu per si la imatge no pot ser mostrada.

On a l'atribut `style` hi especificuem l'estil de la imatge; per exemple, les dimensions de la imatge a mostrar.

La mida de la imatge també es pot especificar a través dels atributs `width` i `height`. És recomanable, però, fer-ho a través de l'atribut `style`, per prevenir possibles fulls d'estils que puguin canviar la mida de la imatge.

### Visualització d'una imatge

```
1 <head>
2   <meta charset="utf-8"/>
3   <title>Primera Pàgina Web</title>
4   <style>
5     img {width:100%;}
6   </style>
7 </head>
8 <body>
9
10  
11  
12
13 </body>
```

El resultat de l'exemple es pot visualitzar en la figura 1.26.

**FIGURA 1.26.** Visualització d'una imatge en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/EPporb?editors=1100](http://codepen.io/loc-daw-m09/pen/EPporb?editors=1100).

Veiem, doncs, que un estil en CSS podria modificar la visualització de la imatge.

---

El navegador Internet Explorer no suporta els elements `<picture>` ni els atributs `srcset` ni `sizes`.

Sovint, perquè el disseny del lloc web sigui adaptatiu, cal mostrar una imatge o una altra més petita en funció de la mida de la pantalla o de la resolució del dispositiu, així com en funció de la mida del navegador des del qual accedim al lloc web.

En la unitat "Elements multimèdia al web: creació i integració" treballareu com incloure imatges en dissenys adaptatius.

Per aquest motiu, HTML5 inclou el nou element <picture> i nous atributs, **srcset** i **sizes**, per a l'element <img>.

No obstant això, a l'hora de visualitzar una imatge cal tenir clars els conceptes de resolució de pantalla i de la imatge:

**La resolució de pantalla** s'expressa en píxels per polzada (PPP) i relaciona el nombre total de píxels d'un monitor amb la seva mida en polzades (*inches*), en la direcció horitzontal i la vertical. En anglès, Pixels Per Inch, PPI.

**La resolució d'una imatge** indica la quantitat de detall que pot observar-se en aquesta. S'expressa amb dos números enters, referents a la quantitat de columnes de píxels (nombre de píxels d'ample) i a la quantitat de files de píxels (nombre de píxels d'alt).

Evitarem mostrar una **imatge gran**, és a dir, amb molta resolució, en un espai petit de la pantalla.

Per a una correcta visualització i per reduir la quantitat d'informació descarregada és important intentar adaptar al màxim els píxels en què es visualitzarà la imatge amb els píxels que realment té la imatge.

## Enllaços

Els enllaços són els elements que permeten a l'usuari navegar d'una pàgina web a una altra.

L'element que permet definir un enllaç en HTML és <a>.

Vegeu els atributs que es poden especificar per a aquest element en la taula taula 1.2.

**TAULA 1.2.** Atributs dels enllaços

Atribut	Descripció
<code>href</code>	Permet definir el destí de l'enllaç.
<code>target</code>	Permet definir on obrir el document enllaçat.
<code>id</code>	Permet definir una marca dins de la pàgina.
<code>style</code>	Permet definir l'estil del <i>link</i> .

### Exemple d'enllaços

```

1   <a href="http://www.google.es" target="_blank">Enllaç a Google<
2       /a>
3   <br />
4   <a href="pagina.html" style="color: darkblue; text-decoration:
5       none"> Enllaç a una pàgina del nostre lloc web </a>
6   <br />
```

```
6 <a href="../imatgeGran.jpg"></a>
```

El resultat de l'exemple es pot visualitzar en la figura 1.27.

**FIGURA 1.27.** Visualització d'enllaços en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/PZdBgO](https://codepen.io/ioc-daw-m09/pen/PZdBgO).

Vegeu algunes consideracions sobre els exemples anteriors:

- L'atribut `href` és obligatori.
- Si el destí de l'enllaç és una pàgina externa al vostre lloc web, cal introduir l'URL sencer.
- Si el destí és una pàgina del nostre lloc web només cal que poseu la ruta d'aquesta pàgina, i en podeu obviar el domini i el protocol.
- Si voleu que l'enllaç permeti la descàrrega d'un fitxer o recurs, en el destí de l'enllaç hi haurà la ruta d'aquest fitxer descarregable. Fixeu-vos també que una imatge pot ser un enllaç.
- L'atribut `target` pot prendre per valor el nom de l'element on voleu que s'obri l'enllaç. El valor predefinit `_blank` significa que s'obre en una nova pestanya.
- Els enllaços, per defecte, són de color blau i apareixen subratllats, però podeu modificar el seu estil mitjançant l'atribut `style`.

## Taules

Si es vol crear una taula en HTML5 s'utilitzaran els elements de la taula taula 1.3.

**TAULA 1.3.** Elements per a les taules

Element	Descripció
<code>&lt;table&gt;</code>	Permet definir una taula.
<code>&lt;tr&gt;</code>	Permet definir una fila de la taula.
<code>&lt;td&gt;</code>	Permet definir un cel de la taula.
<code>&lt;th&gt;</code>	Permet definir un cel de capçalera d'una taula.

**TAULA 1.3** (continuació)

Element	Descripció
<td>	Permet definir una columna d'una fila.
<th>	Permet definir les capçaleres de la taula.
<caption>	Permet definir el títol d'una taula.

No sempre és necessari utilitzar tots els elements per definir una taula, però veureu com els elements <table>, <tr> i <td> són indispensables per visualitzar-la correctament.

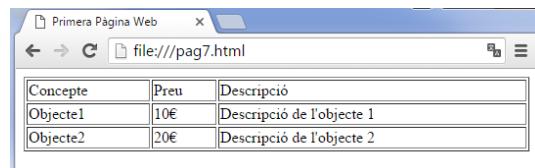
#### Exemple de taula

```

1  <table id="taula1" border="1" style="width:100%;">
2    <tr>
3      <td>Concepte</td>
4      <td>Preu</td>
5      <td>Descripció</td>
6    </tr>
7    <tr>
8      <td>Objecte1</td>
9      <td>€10</td>
10     <td>Descripció de l'objecte 1</td>
11   </tr>
12   <tr>
13     <td>Objecte2</td>
14     <td>€20</td>
15     <td>Descripció de l'objecte 2</td>
16   </tr>
17 </table>

```

El resultat de l'exemple es pot visualitzar en la figura 1.28.

**FIGURA 1.28.** Visualització d'una taula en un navegador

Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/XXBZZB?editors=1000](http://codepen.io/ioc-daw-m09/pen/XXBZZB?editors=1000).

Tot i que en l'exemple anterior hem utilitzat l'atribut `border` per especificar que es vegi la vora de la taula, tots els aspectes del format de visualització d'una taula se solen especificar en llenguatge CSS a través, per exemple, del seu atribut `style`.

Per combinar cel·les d'una taula s'utilitzen els atributs `colspan` i `rowspan` dels elements <td> o <th>.

#### Exemple de taula amb files combinades

```

1  <table id="taula2" style="border:1px solid black">
2    <caption> Notes Finals</caption>
3    <tr>
4      <th>Alumne</th>
5      <th>Mòdul</th>
6      <th>Nota</th>
7    </tr>

```

```

8   <tr>
9     <td rowspan="3" style="vertical-align:text-top;">Nom Alumne1<
10    /td>
11    <td>M1 Sistemes Operatius</td>
12    <td>9</td>
13  </tr>
14  <tr>
15    <td>M2 Bases de Dades</td>
16    <td>8</td>
17  </tr>
18  <tr>
19    <td>M3 Programació</td>
20    <td>7</td>
21  </tr>
22
23  <tr>
24    <td rowspan="3" style="vertical-align:text-top;">Nom Alumne2<
25    /td>
26    <td>M1 Sistemes Operatius</td>
27    <td>7</td>
28  </tr>
29  <tr>
30    <td>M2 Bases de Dades</td>
31    <td>6</td>
32  </tr>
33  <tr>
34    <td>M3 Programació</td>
35    <td>5</td>
36  </tr>
37 </table>

```

El resultat de l'exemple es pot visualitzar en la figura 1.29.

**FIGURA 1.29.** Visualització d'una taula amb files combinades en un navegador

Notes Finals		
Alumne	Mòdul	Nota
Nom Alumne1	M1 Sistemes Operatius	9
	M2 Bases de Dades	8
	M3 Programació	7
Nom Alumne2	M1 Sistemes Operatius	7
	M2 Bases de Dades	6
	M3 Programació	5

Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/eJrpp?editors=1000](http://codepen.io/loc-daw-m09/pen/eJrpp?editors=1000).

Com veieu a l'exemple anterior, a les capçaleres, a la fila 1 i a la fila 5 es defineixen tres columnes. En canvi, a les files 2, 3, 6 i 7 només es defineixen dues columnes, ja que la primera columna està ja ocupada.

L'atribut colspan funciona de manera similar, i el resultat de l'exemple en el navegador es pot visualitzar en la figura 1.30.

#### Exemple de taula amb cel·les combinades

```
1 <table id="taula3" border="1">
```

```

2   <caption> Notes Finals</caption>
3   <tr>
4     <td rowspan="2" style="vertical-align:bottom;">Alumne</td>
5     <td colspan="3">Mòdul 1 Sistemes</td>
6   </tr>
7   <tr>
8     <td>EAC</td>
9     <td>PAF</td>
10    <td>Final</td>
11  </tr>
12 <tr>
13  <td>Nom Alumne1</td>
14  <td>8</td>
15  <td>8</td>
16  <td>8</td>
17 </tr>
18 <tr>
19  <td>Nom Alumne2</td>
20  <td>7</td>
21  <td>7</td>
22  <td>7</td>
23 </tr>
24 <tr>
25  <td>Nom Alumne3</td>
26  <td>6</td>
27  <td>6</td>
28  <td>6</td>
29 </tr>
30 <tr>
31  <td>Nom Alumne4</td>
32  <td>7</td>
33 </tr>
34 </table>

```

**FIGURA 1.30.** Visualització d'una taula amb cel·les combinades en un navegador

Alumne	Mòdul 1 Sistemes		
	EAC	PAF	Final
Nom Alumne1	8	8	8
Nom Alumne2	7	7	7
Nom Alumne3	6	6	6

Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/MKBGOy?editors=1000](http://codepen.io/loc-daw-m09/pen/MKBGOy?editors=1000).

Com veieu a l'exemple anterior, la cel·la "alumne" ocupa dues files i la cel·la "Mòdul1 Sistemes" ocupa tres columnes.

## Logotip o icona

Si voleu veure el logotip del lloc web o la icona a la barra d'adreses o a la pestanya del navegador haureu de crear la icona *.ico*, reanomenar-la amb el nom favicon.ico i posar-la a l'arrel del nostre lloc web.

El motiu que el fitxer es digui favicon.ico i es posi a l'arrel del lloc és perquè, per defecte, els navegadors com Firefox, Safari o Opera busquen en aquesta ubicació.

Per assegurar-nos que el navegador troba el nostre *favicon* és recomanable inserir la següent línia de codi a la capçalera del nostre document HTML (entre les etiquetes *<head>* i *</head>*), indicant la ruta del *favicon*, que en aquest cas és l'arrel del lloc:

```
1 <link rel="Shortcut Icon" href="favicon.ico" type="image/x-icon" />
```

Per crear la nostra icona, primer cal fer una imatge quadrada, per exemple 32x32 píxels, en format JPG, GIF o PNG.

Si es vol que la icona tingui un fons transparent, llavors aquesta imatge inicial ha de ser en format PNG i amb el fons transparent.

Per convertir aquesta imatge a .ico es pot fer a través de la web [www.favicon.cc](http://www.favicon.cc).

### 1.4.4 Formularis amb HTML5

Els formularis s'utilitzen perquè l'usuari pugui introduir dades per després ser enviades i processades pel servidor.

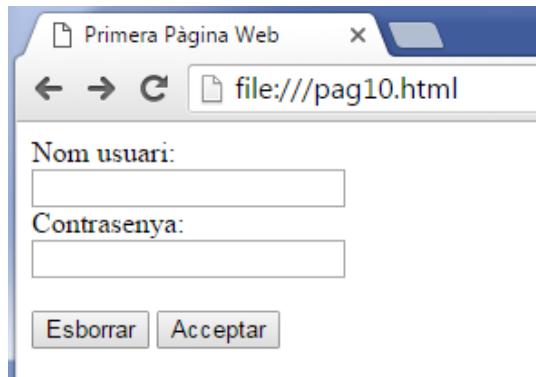
L'element HTML que permet crear formularis és *<form>*, i en el seu contingut s'hi defineixen els diferents camps que en formen part.

#### Exemple de formulari

```
1 <form id="ValidaUsuari" action="mailto:nom@correu.com" method="post" autocomplete="on">
2
3   Nom usuari: <br />
4   <input type="text" name="nomUsuari" /><br />
5   Contrasenya: <br />
6   <input type="password" name="contrasenya" />
7   <br />
8   <br />
9   <input type="reset" value="Esborrar"/>
10  <input type="submit" value="Acceptar"/>
11
12 </form>
```

El resultat de l'exemple es pot visualitzar en la figura 1.31.

**FIGURA 1.31.** Visualització d'un formulari en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/NxBMQG?editors=1000](https://codepen.io/ioc-daw-m09/pen/NxBMQG?editors=1000).

Vegem, en l'exemple, els atributs més importants que té l'element <form>:

- **id:** indica l'identificador del formulari.
- **action:** indica l'URL o el nom de la pàgina que s'encarrega de processar les dades del formulari.
- **method:** indica el mètode d'enviament del formulari, que pot ser `get` o `post`. Si és `get`, la informació del formulari s'enviarà a través de l'URL al qual s'envia el formulari. Si és `post`, s'envia encapsulada en el paquet HTTP.
- **autocomplete:** indica si el navegador completarà automàticament els valors dels camps del formulari basant-se en valors que l'usuari hagi entrat amb anterioritat. El valor d'aquest atribut pot ser `on` o `off`.
- **novalidate:** si aquest atribut és present es desactiva la validació nadia que HTML5 ofereix en els formularis.
- **name:** permet definir un nom al formulari.
- **enctype:** permet definir el tipus de dades que s'enviaran.

Els camps d'un formulari es poden definir a través de diferents elements HTML5:

- l'element <input>
- l'element <textarea>
- l'element <select>
- l'element <button>
- l'element <datalist>

## L'element <input>

Mitjançant l'element <input>, i a través del seu atribut `type`, podem inserir diferents tipus de camp.

Vegem els diferents tipus en la taula taula 1.4.

**TAULA 1.4.** Valors de l'atribut "type" de l'element "input"

Valor	Descripció
<code>text</code>	Permet inserir un text.
<code>password</code>	Permet inserir una contrasenya, amagant els caràcters.
<code>hidden</code>	Permet inserir un camp ocult per a l'usuari.
<code>submit</code>	Permet inserir un botó per enviar les dades al servidor.
<code>image</code>	Permet inserir una imatge com a botó per enviar les dades.
<code>button</code>	Permet inserir un botó.
<code>radio</code>	Permet inserir un botó <i>radio</i> .
<code>checkbox</code>	Permet inserir un <i>checkbox</i> .
<code>color</code>	Permet seleccionar un color en el sistema de selecció de colors del sistema operatiu.
<code>date</code>	Permet inserir una data. Els navegadors proposaran una interfície de selecció de la data.
<code>datetime</code>	Permet inserir una data i l'hora amb la diferència horària.
<code>datetime-local</code>	Permet inserir una data i l'hora sense la diferència horària.
<code>time</code>	Permet inserir solament l'hora.
<code>month</code>	Permet inserir el mes d'una data.
<code>week</code>	Permet inserir la setmana.
<code>email</code>	Permet inserir un correu electrònic. Els dispositius mòbils adaptaran el teclat.
<code>number</code>	Permet inserir un número. Els dispositius mòbils adaptaran el teclat.
<code>range</code>	Mostrarà una barra de selecció amb un cursor amb el qual podrem escollir un valor. No suportat en IE.
<code>search</code>	Permet inserir un camp d'introducció de dades específic per a les cerques.
<code>tel</code>	Permet inserir un número de telèfon. Els dispositius mòbils adaptaran el teclat.
<code>url</code>	Permet inserir un URL. Els dispositius mòbils adaptaran el teclat.

Vegem que a través de l'element `input` es poden crear diferents tipus de camp. I, de fet, hi ha diversos camps, com per exemple `email`, `number`, `url`, etc., que aparentment són iguals, però la seva utilització ajudarà a la validació correcta de les dades introduïdes en el formulari i que en dispositius mòbils el seu teclat s'adapti per a una millor utilització.

A part de l'atribut `type`, també podem definir altres atributs per a l'element `<input>`. Vegem-ne un recull en la taula taula 1.5.

**TAULA 1.5.** Atributs de l'element "input"

Atribut	Descripció
<code>id</code>	Permet definir un identificador.
<code>name</code>	Permet definir un nom per identificar el camp.
<code>value</code>	Permet definir quin valor ha de prendre el camp del formulari.
<code>readonly</code>	Permet definir que aquest camp només és de lectura.
<code>disabled</code>	Permet desactivar el camp.
<code>selected</code>	Permet definir que aquest camp estigui seleccionat.
<code>checked</code>	Permet definir que un botó de <i>radio</i> o <i>checkbox</i> estigui seleccionat.
<code>size</code>	Permet definir la mida del camp.
<code>maxlength</code>	Permet definir la mida màxima del valor del camp.
<code>autocomplete</code>	Permet activar la funció <i>autocomplete</i> .
<code>autofocus</code>	Permet definir que aquest camps és el que té el focus per defecte.
<code>min i max</code>	Permet definir un valor mínim i màxim.
<code>placeholder</code>	Permet definir una descripció o exemple per al valor del camp.
<code>required</code>	Permet definir que aquest camp és obligatori i s'utilitza per a la validació nadiua d'HTML5 del formulari.

No tots els atributs són aplicables a tots el tipus de camp; així per exemple, no té sentit aplicar l'atribut `placeholder` per a un camp de data o aplicar els atributs `min i max` a un camp de contrasenya.

Vegem un exemple per comprovar la utilització d'alguns tipus camp amb els seus atributs, el resultat del qual el podem veure en la figura 1.32.

#### Exemple d'un formulari amb diferents camps de tipus <input>

```

1  <h3>Dades de l'usuari</h3>
2  <hr />
3  <form method="post" action="mailto:usuari@correu.com"
4      autocomplete>
5      <input type="text" name="nom" size="50" maxlength="10"
6          placeholder="Escriu el nom" required autofocus/><br />
7      <input type="text" name="cognom" size="50" maxlength="20"
8          placeholder="Escriu el cognom" required /><br />
9      <input type="email" name="email" size="50" placeholder="Escriu
10         el correu electrònic" /><br />
11     <input type="password" name="contrasenya" size="50" placeholder
12         ="Escriu la contrasenya" disabled/><br />
13     Data de naixement: <br />
14     <input type="date" name="dataNaix"/>
15     <hr />
16     Gènere: <br />
17     <input type="radio" name="gender" value="masculi" checked />
18         Home <br />
19     <input type="radio" name="gender" value="femeni" />Dona <br />
20     <hr />

```

```

15 Aficions: <br />
16 <input type="checkbox" name="aficio1" value="esport" />Esport <
17     br />
18 <input type="checkbox" name="aficio2" value="cinema" />Cinema <
19     br />
20 <input type="checkbox" name="aficio3" value="llegir" />Lectura
21     <br />
22 <input type="checkbox" name="aficio4" value="altres" />Altres <
23     br />
24 Selecciona el teu color preferit:
25 <input type="color" name="colorPreferit" /> <br />
26 <hr />
27 Foto:
28 <input type="file" name="foto" />
29 <hr />
30 <input type="button" value="Fes clic!" onclick="alert('HOLA
31     !!!');" />
32 <input type="reset" value="Esborrar" />
33 <input type="submit" value="Enviar" />
34 <hr />
35
36 </form>

```

**FIGURA 1.32.** Visualització d'un formulari amb diferents tipus de camps en un navegador

The screenshot shows a web browser window with the title "Primera Pàgina Web". The address bar displays "file:///pag11.html". The page content is a form titled "Dades de l'usuari". The form fields include:

- Text input fields for "Escriu el nom", "Escriu el cognom", "Escriu el correu electrònic", and "Escriu la contrasenya".
- A text input field for "Data de naixement" with the placeholder "dd / mm / aaaa".
- Gender selection: Radio buttons for "Home" (selected) and "Dona".
- Hobbies selection: Checkboxes for "Esport", "Cinema", "Lectura", and "Altres".
- A color selection input field labeled "Selecciona el teu color preferit" with a small color swatch.
- A file input field for "Foto" with the placeholder "Tria un fitxer" and the message "No s'ha triat cap fitxer".
- Action buttons at the bottom: "Fes clic!", "Esborrar", and "Enviar".

Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/YwjRzy?editors=1000](https://codepen.io/ioc-daw-m09/pen/YwjRzy?editors=1000).

Algunes consideracions sobre l'exemple serien:

- Perquè els botons de *radio* es discriminin entre si han de tenir el mateix nom, al contrari que els botons *checkbox*, que cadascun ha de tenir un nom diferent.
- Cada camp té un nom, el qual s'utilitzarà perquè el servidor que rep les dades pugui accedir a cadascuna de les dades rebudes. També es pot especificar un identificador a través de l'atribut *id*, i el servidor també hi podria accedir a través d'aquest. Però fixem-nos que els botons de *radio* necessiten especificar l'atribut *name* per al seu funcionament.
- Fixem-nos també en la diferència dels atributs *maxlength* i *size*: el primer fa referència a la mida del valor que se li donarà al camp i el segon, a la mida en què es visualitza el camp de text.
- També en alguns camps s'han utilitzat els atributs *disabled*, *placeholder* i *required*.

## L'element <textarea>

Aquest element permet definir un camp de text de diverses línies.

### Exemple de camp de text de diverses línies

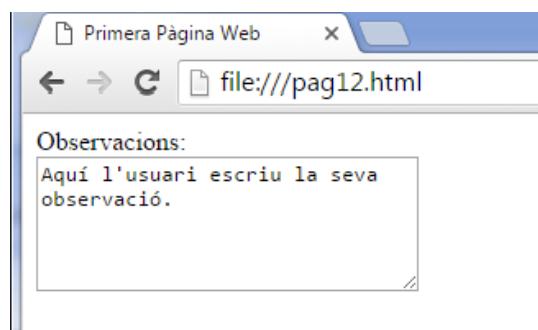
```

1 Observacions: <br />
2 <textarea id="observacions" rows="10" cols="30">
3 Aquí l'usuari escriu la seva observació.
4 </textarea>

```

El resultat de l'exemple es pot visualitzar en la figura 1.33.

**FIGURA 1.33.** Visualització de l'element <textarea> en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/LGJYPw?editors=1000](https://codepen.io/ioc-daw-m09/pen/LGJYPw?editors=1000).

## L'element <select>

Permet definir una llista d'opcions seleccionables. Cada opció s'especifica amb l'element <option>.

### Exemple d'utilització de l'element <select>

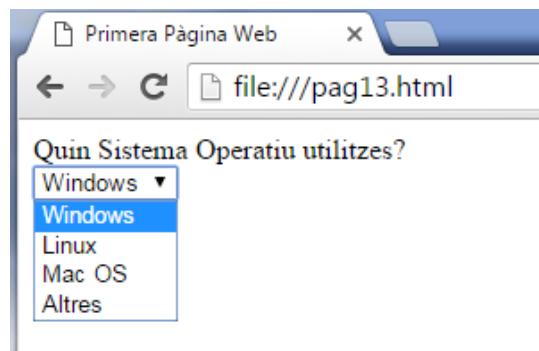
```

1 Quin Sistema Operatiu utilitze? <br />
2 <select id="SO" name="SO">
3   <option value="windows">Windows</option>
4   <option value="linux">Linux</option>
5   <option value="mac">Mac OS</option>
6   <option value="altres">Altres</option>
7 </select>

```

El resultat de l'exemple es pot visualitzar en la figura 1.34.

**FIGURA 1.34.** Visualització d'una llista desplegable <select> en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/gPdOrg?editors=1000](https://codepen.io/loc-daw-m09/pen/gPdOrg?editors=1000).

## L'element <button>

Amb aquest element es pot inserir un botó al formulari, i és molt similar a <input type="button" >. La diferència és que a través d'aquest es pot modificar no tan sols el text del botó, sinó que per exemple s'hi podria especificar una imatge.

### Exemple d'utilització de l'element <button>

```

1 <button type="button" onclick="alert('Hola a tots!')"><img src=
  fletxa.png" /></button>

```

### JavaScript

És un llenguatge *script* que s'executa a la part del client, és a dir, en el navegador. Permet millores a la interfície d'usuari d'aplicacions i pàgines web dinàmiques.

## L'element <datalist>

Amb aquest element podem especificar una llista d'opcions per suggerir a l'usuari algunes opcions predefinides per a un element <input>. Els usuaris veuran una llista desplegable amb les opcions predefinides a mesura que vagin introduint el text, però poden acabar escrivint un altre text.

### Exemple d'utilització de l'element <datalist>

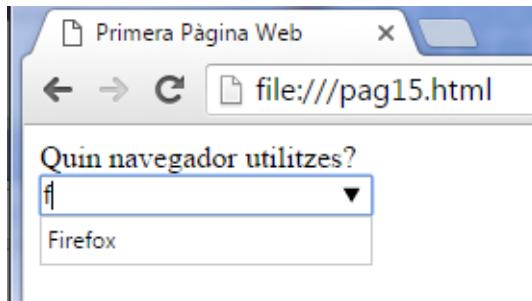
```

1  <input list="navegadors">
2  <datalist id="navegadors">
3      <option value="Internet Explorer">
4      <option value="Firefox">
5      <option value="Chrome">
6      <option value="Opera">
7  </datalist>

```

El resultat de l'exemple es pot visualitzar en la figura 1.35.

**FIGURA 1.35.** Visualització d'un element <datalist> en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/pgOoRZ?editors=1000](http://codepen.io/ioc-daw-m09/pen/pgOoRZ?editors=1000).

## 1.4.5 Visualització d'elements amb HTML5

Cada element HTML té un valor de visualització per defecte. En la majoria d'elements, aquest valor és *block* o *inline*.

La visualització per defecte dels elements HTML tant de tipus *block* com de tipus *inline* es pot modificar mitjançant propietats CSS per tal d'aconseguir un disseny adaptatiu o, en anglès, *Responsive Web Design*.

Un element amb visualització de ***block*** sempre comença a visualitzar-se en una nova línia, i ocupa tota l'amplada disponible. Un element amb visualització ***inline*** no comença en una nova línia i només agafa l'amplada necessària.

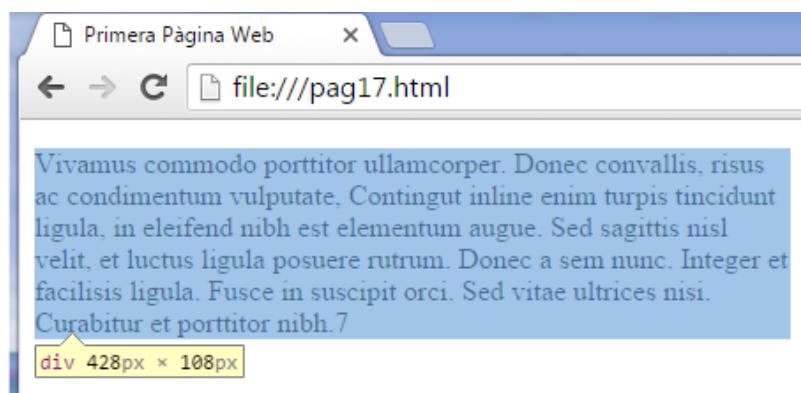
### Exemple d'element "block" i "inline"

```

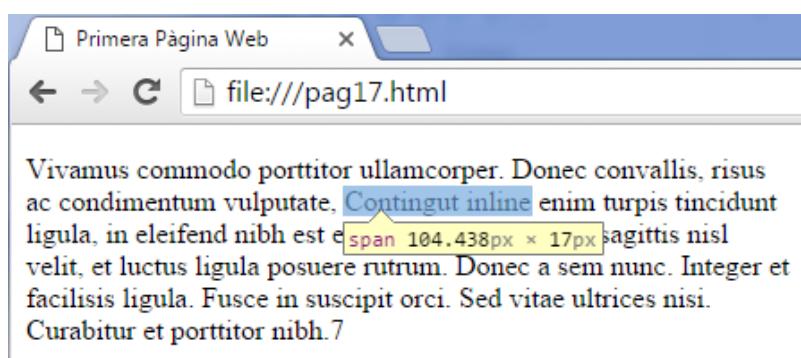
1  <div>
2  <p>
3  Vivamus commodo porttitor ullamcorper. Donec convallis, risus ac
4      condimentum vulputate,
5  <span>Contingut inline</span>
6  enim turpis tincidunt ligula, in eleifend nibh est elementum
7      augue. Sed sagittis nisl velit,
8  et luctus ligula posuere rutrum. Donec a sem nunc. Integer et
9      facilisis ligula. Fusce in suscipit orci.
10 Sed vitae ultrices nisi. Curabitur et porttitor nibh.7
11 </p>
12 </div>

```

El resultat el podeu veure en les figura 1.36 i figura 1.37.

**FIGURA 1.36.** Visualització d'una caixa de tipus "block"

Podem apreciar el contingut de tipus 'block' remarcat.

**FIGURA 1.37.** Visualització d'una caixa de tipus "inline"

Podem apreciar el contingut de tipus "inline" remarcat.

#### 1.4.6 Estructura semàntica de la pàgina web

En contraposició a la manera d'estructurar el contingut de les pàgines web que utilitzava HTML4 a través de gairebé únicament element <div>, amb el nou estàndard HTML5 arriben nous elements que aporten estructura **semàntica**.

Destaquem els següents elements que aporten semàntica a l'estructura de la pàgina (*page layout*):

---

Es pot estructurar una pàgina web amb taules o marcs, però actualment aquestes tècniques es consideren obsoletes i han caigut en desús.

---

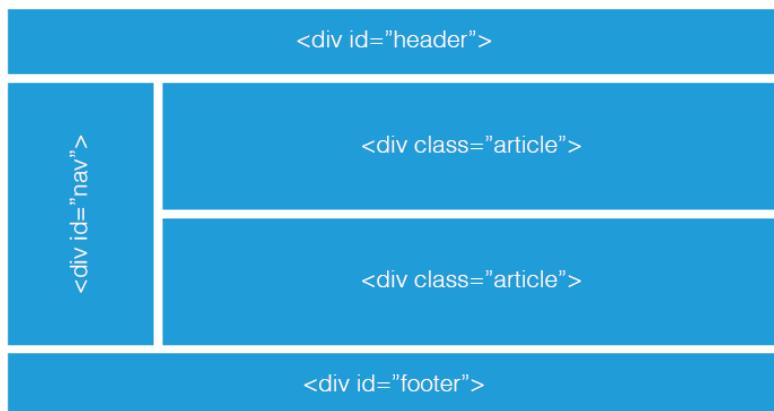
- Element <header>: permet inserir una zona de visualització per a les capçaleres.
- Element <footer>: permet inserir una zona de visualització per als peus de pàgina.
- Element <nav>: permet inserir una zona de visualització per a la navegació amb vincles hipertext.
- Element <section>: permet agrupar elements que tinguin la mateixa temàtica.
- Element <article>: permet inserir un contingut autònom.

- Element `<aside>`: permet mostrar un contingut relacionat amb el contingut al qual estigui vinculat.
- Element `<main>`: permet representar el contingut principal de la pàgina.

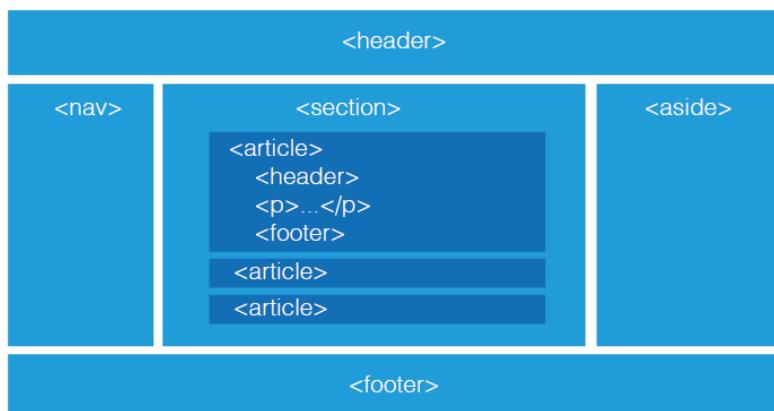
Cal mencionar que en HTML5 continua sent de molta utilitat l'ús de les caixes `<div>`.

Podeu apreciar en les figura 1.38 i figura 1.39 l'estrucció del contingut d'una pàgina web en HTML4 i HTML5, respectivament.

**FIGURA 1.38.** Estructura dels continguts en HTML4



**FIGURA 1.39.** Estructura dels continguts en HTML5



Vegem un exemple de pàgina web utilitzant aquestes etiquetes amb sentit semàntic.

#### Estructura del contingut d'una pàgina web en HTML5

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head><meta charset="utf-8"/><title>Primera Pàgina Web</title>
4  <style>
5  #capa1{
6    width:800px;
7    margin:0 auto;
8  }
9  header {
10   background-color:#0000ee;
11   color:white;

```

```
12     text-align:center;
13     padding:5px;
14 }
15 nav {
16     line-height:30px;
17     background-color:#eeeeee;
18     height:515px;
19     width:100px;
20     float:left;
21     padding:5px;
22 }
23 section {
24     background-color:#aaaaaa;
25     float:left;
26     padding:10px;
27     width:560px;
28 }
29 article {
30     background-color:#ffffff;
31     padding:10px;
32 }
33 aside {
34     line-height:30px;
35     background-color:#eeeeee;
36     height:515px;
37     width:100px;
38     float:right;
39     padding:5px;
40 }
41 footer {
42     background-color:#0000ee;
43     color:white;
44     clear:both;
45     text-align:center;
46     padding:5px;
47 }
48 </style>
49
50 </head>
51 <body>
52 <div id="capa1">
53     <header>
54         <h1>Estudis Grau Superior Informàtica</h1>
55     </header>
56
57     <nav>
58         <p>ASIX </p>
59         <p>DAM </p>
60         <p>DAW </p>
61     </nav>
62
63     <section>
64         <article>
65             <h2>M1. Sistemes Informàtics</h2>
66             <p> Bloc 1: 118 hores (8 hores setmanals) </p>
67             <p> Bloc 2: 80 hores (5 hores setmanals) </p>
68         </article>
69
70         <article>
71             <h2>M2. Bases de dades </h2>
72             <p> Bloc 1: 132 hores (9 hores setmanals). </p>
73             <p> Bloc 2: 99 hores (7 hores setmanals) incloent-hi
74                 treball en equip i pràctica virtual obligatòria.
75             </p>
76         </article>
77
78         <article>
```

```

77      <h2>M3. Programació </h2>
78      <p> Bloc 1: 85 hores (5 hores setmanals). Inclou
79          treball en equip i pràctiques virtuales obligatò
80          ries </p>
81      </article>
82
83  </section>
84
85  <aside>
86      <p>Links d'interès</p>
87  </aside>
88
89  <footer>
90      <p>IOC – Institut Obert de Catalunya</p>
91  </footer>
92
93 </div>
94 </body>
95 </html>

```

El resultat del codi es pot veure en la figura 1.40.

**FIGURA 1.40.** Visualització d'una pàgina web amb estructura semàntica en un navegador



Podeu veure aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/pgOoQG?editors=1100](http://codepen.io/ioc-daw-m09/pen/pgOoQG?editors=1100).

Per a la visualització de cada element s'utilitza un estil en llenguatge CSS. Així veiem que la mida, la posició i el color de cadascun d'aquests elements està definit a través d'un estil estil CSS al *head* d'aquesta pàgina, i per tant, en el *body* només hi trobem el contingut.

## 1.4.7 Validació HTML5

Resulta de gran utilitat poder verificar els errors que hi pot haver a les pàgines.

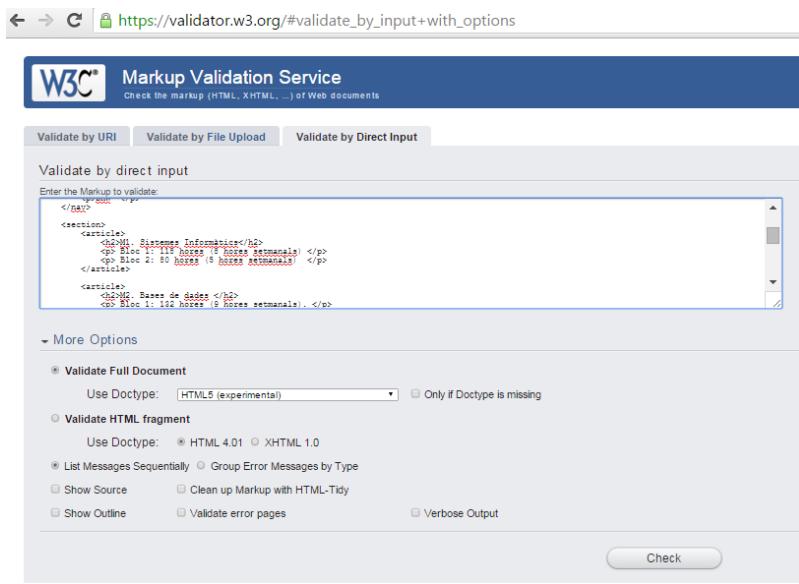
L'eina en línia accessible des de [validator.w3.org](https://validator.w3.org) permet validar el codi HTML de diferents maneres:

- Indicant la URL completa de la pàgina que desitgem validar.
- Pujant el document HTML.
- Escrivint el codi HTML directament.

Una vegada indicat el codi HTML a validar, polseu l'opció *More Options*, i allí indiqueu el *Document Type*, en el vostre cas HTML5.

Finalment, feu clic al botó *Check* perquè s'executi la validació, tal com es mostra en la figura 1.41.

**FIGURA 1.41.** Validació d'HTML



Interfície gràfica de l'aplicació per validar HTML

En uns instants tindreu el llistat i els detalls d'errors i alertes, si és que el validador en detecta.

## 1.4.8 Seguretat HTML5

Actualment existeixen moltes aplicacions web disponibles en HTML5, però aquest llenguatge té potencials vulnerabilitats de seguretat. Algunes d'elles provenen dels elements de l'estàndard en si, i d'altres provenen de les implementacions de cada navegador.

Algunes de les vulnerabilitats més importants d'HTML5 són:

- Accés a les dades d'altres dominis: HTML5 integra una API anomenada postMessage que permet a un *script* d'un domini passar dades a un *script* que estigui en un altre domini. Per assegurar que aquestes peticions no són malicioses, postMessage inclou propietats perquè el programador pugui verificar l'origen de la petició. Però aquestes comprovacions no són obligatòries, i per tant depèn del programador. Podeu consultar més informació sobre aquesta vulnerabilitat a: [code.google.com/archive/p/html5security/wikis/CrossOriginRequestSecurity.wiki](http://code.google.com/archive/p/html5security/wikis/CrossOriginRequestSecurity.wiki).
- Alguns elements HTML5 permeten l'execució de codi JavaScript, i això ho podria aprofitar un atacant per injectar codi JavaScript i així accedir al control de la navegabilitat de la web. Podeu consultar més informació sobre aquesta vulnerabilitat a: [code.google.com/archive/p/html5security/wikis/RedirectionMethods.wiki](http://code.google.com/archive/p/html5security/wikis/RedirectionMethods.wiki).
- Emmagatzemament local: HTML5 permet l'emmagatzemament *offline*, i fins i tot es pot accedir a una base de dades local amb JavaScript. D'aquesta manera, les aplicacions web poden funcionar més ràpidament, però aquest sistema també presenta vulnerabilitats, així que només s'hauria d'utilitzar l'emmagatzemament local en dades que no siguin sensibles i que siguin accessibles per la totalitat dels usuaris. Podeu consultar més informació sobre aquesta vulnerabilitat a: [code.google.com/archive/p/html5security/wikis/WebSQLDatabaseSecurity.wiki](http://code.google.com/archive/p/html5security/wikis/WebSQLDatabaseSecurity.wiki) o bé, a [developer.mozilla.org/es/docs/Web/HTML/Recursos\\_offline\\_en\\_firefox](http://developer.mozilla.org/es/docs/Web/HTML/Recursos_offline_en_firefox).