

# 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

2.3	Bootstrap 'framework'	117
2.3.1	Com obtenir i instal·lar Bootstrap	118
2.3.2	El contingidor	119
2.3.3	Bootstrap 'grid'	119
2.3.4	Elements	125
2.4	Cas pràctic amb Bootstrap: Acadèmia esTUDia	129
<b>3</b>	<b>Generació de llocs web: prototips i plantilles</b>	<b>139</b>
3.1	Fases d'un projecte web	139
3.1.1	Planificació	139
3.1.2	Disseny	140
3.1.3	Desenvolupament	140
3.1.4	Publicació	141
3.1.5	Manteniment	141
3.2	Elaboració d'un prototipus web	141
3.2.1	Pas 1: obtenir la inspiració	142
3.2.2	Pas 2: creació del 'wireframe'	143
3.2.3	Pas 3: creació del 'mockup'	147
3.2.4	Pas 4: elaboració del prototip	150
3.3	Plantilles web	151
3.3.1	Criteris per triar una plantilla	151
3.3.2	Com utilitzar una plantilla	152
3.3.3	Cas pràctic: creació d'un prototip a partir d'una plantilla web	153
3.4	Sistema de gestió de continguts	158
3.4.1	WordPress	159
3.4.2	Joomla!	160
3.4.3	CMS per a comerç 'online'	160
3.5	Altres components d'un lloc web	161
3.5.1	Lloc de cerca	163
3.5.2	Espai de Xat	163
3.5.3	Formulari de contacte	164
3.5.4	Newsletter	165
3.5.5	Bases de dades en línia	165
3.5.6	Seccions protegides amb contrasenya	165
3.6	Posicionament web	166
3.6.1	Tècniques SEO	167
3.6.2	Optimitzar el lloc web	170

## 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.



## Resultats d'aprenentatge

En finalitzar aquesta unitat, l'alumne/a:

1. Planifica la creació d'una interfície web valorant i aplicant especificacions de disseny.

- Reconeix la importància de la comunicació visual i els seus principis bàsics.
- Analitza i selecciona els colors i les tipografies adequats per a la seva visualització en pantalla.
- Analitza alternatives per a la presentació de la informació en documents web.
- Valora la importància de definir i aplicar la guia d'estil en el desenvolupament d'una aplicació web.
- Utilitza i valora diferents aplicacions per al disseny de documents web.
- Utilitza marcs, taules i capes per presentar la informació de manera ordenada.
- Crea i utilitza plantilles de disseny.

2. Crea interfícies web homogènies definint i aplicant estils.

- Reconeix les possibilitats de modificar les etiquetes HTML.
- Defineix estils de manera directa.
- Defineix i associa estils globals en fulls externs.
- Defineix fulls d'estil alternatius.
- Redefineix estils.
- Identifica les diferents propietats de cada element.
- Crea classes d'estils.
- Utilitza eines de validació de fulls d'estils.
- Empra i actualitza la guia d'estil.



## 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.

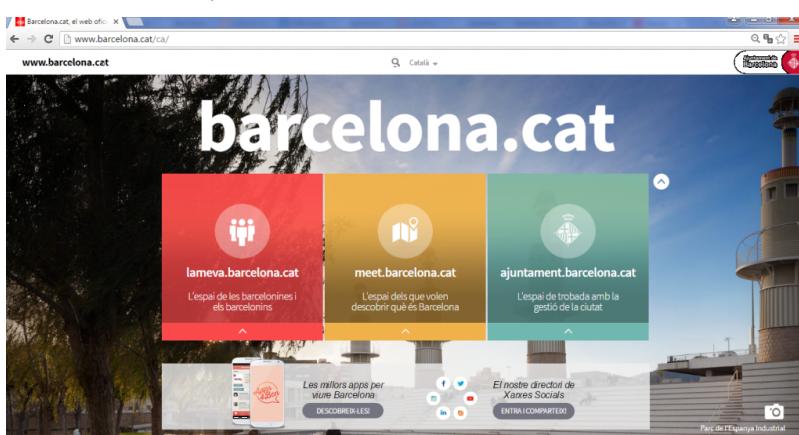
Així doncs, el disseny d'un lloc web no dependrà només de factors com el dispositiu des del qual s'hi accedeix, la usabilitat, l'accessibilitat, el disseny gràfic, la qualitat dels continguts o la utilitat, sinó que també fa referència a aquells aspectes relatius a les emocions, als sentiments i a la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

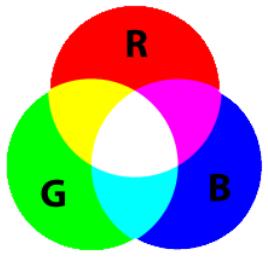
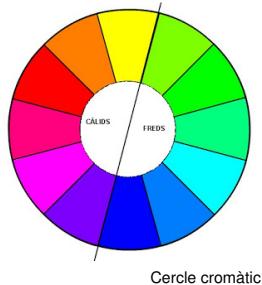


La utilització equivocada del color pot induir l'usuari a error.

## 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 luminància (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.

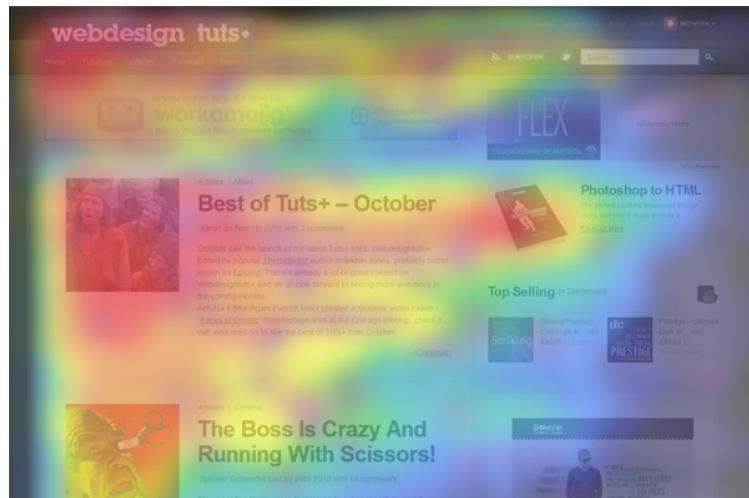
**La composició** es defineix com una distribució equilibrada de tots els elements que s'inclouen en un disseny. En aquest sentit, la composició del disseny d'un lloc web ha de ser estètica i funcional.

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**.

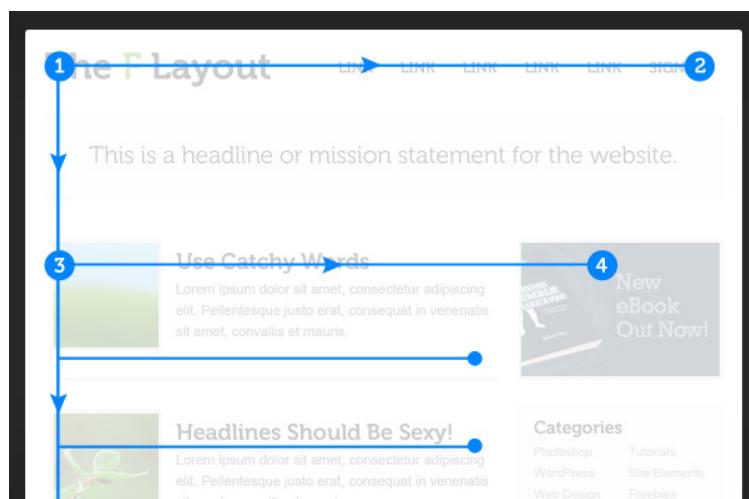
Vegeu un exemple a la figura 1.5 i la figura 1.6.

**FIGURA 1.5.** Mapa de calor



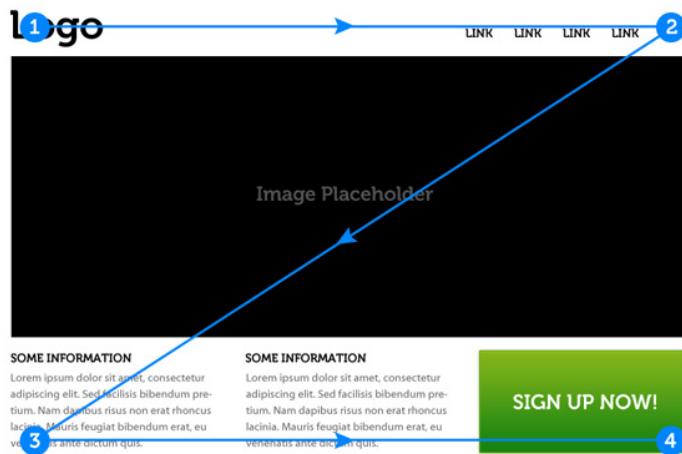
Font: <http://webdesign.tutsplus.com/es/articles/understanding-the-f-layout-in-web-design--webdesign-687>

**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



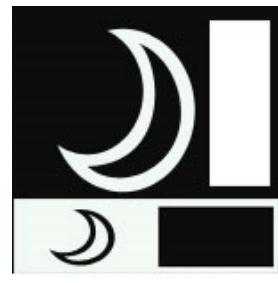
Font: <http://webdesign.tutsplus.com/articles/understanding-the-z-layout-in-web-design--webdesign-28>

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.



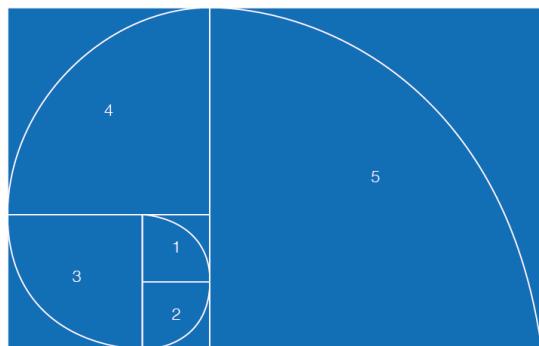
Exemple d'utilització del contrast de to

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.

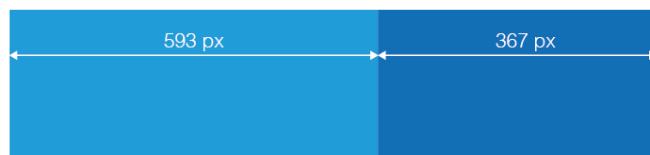
Així, per exemple, si es vol crear un disseny d'amplada fixa, 960 px, i es vol dividir l'espai en dues columnes, es deixarà una mida més gran per al contingut i una altra de més petita per a la barra lateral.

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**.

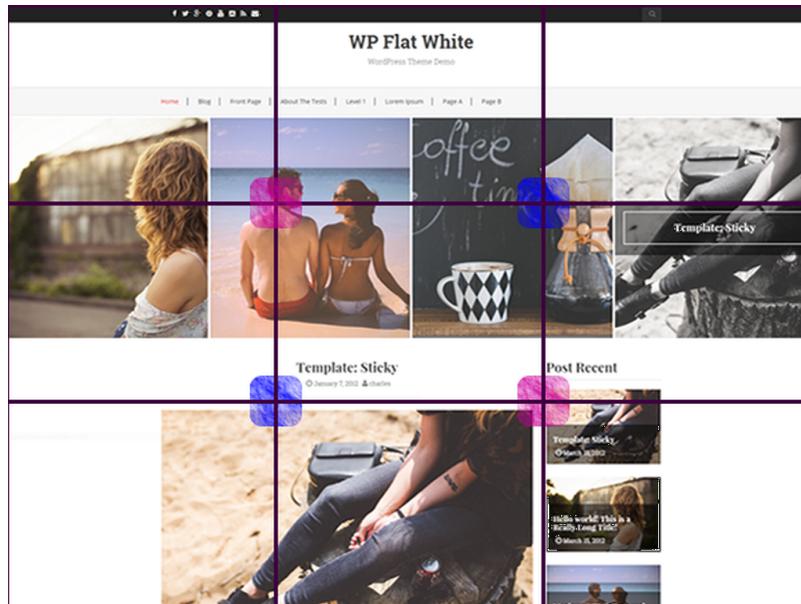
Aquesta regla s'aplica dividint la composició en 9 parts iguals, de tal manera que es divideix la composició mitjançant dues línies horitzontals i dues verticals equidistants. Els quatre punts que formen la intersecció d'aquestes línies es poden

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.

#### 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.

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.

#### 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.

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

- Els **esborrany**s, 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.



Logotip de W3 Consortium.

#### **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.

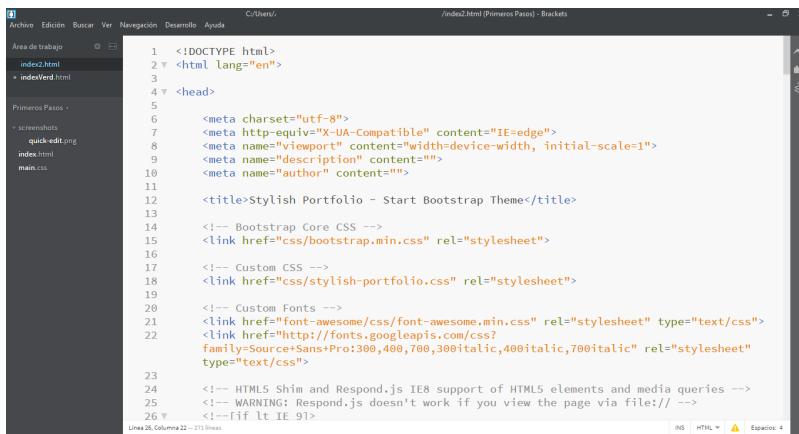
editors d'HTML amb els quals podem treballar actualment són Brackets, Sublime Text, Atom, etc.

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.

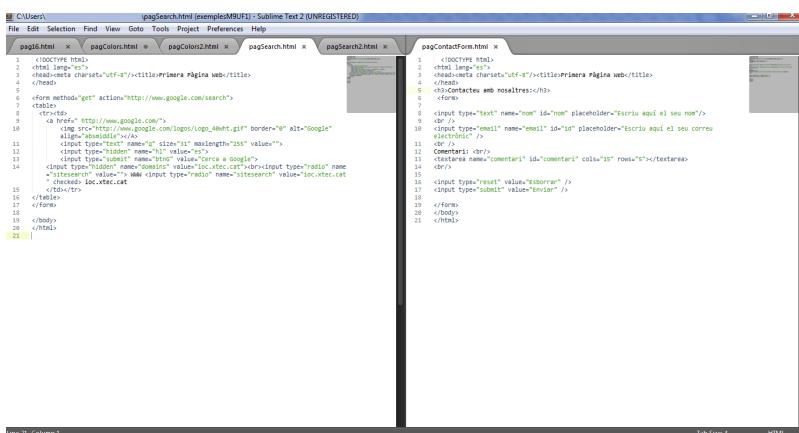
## Github

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.

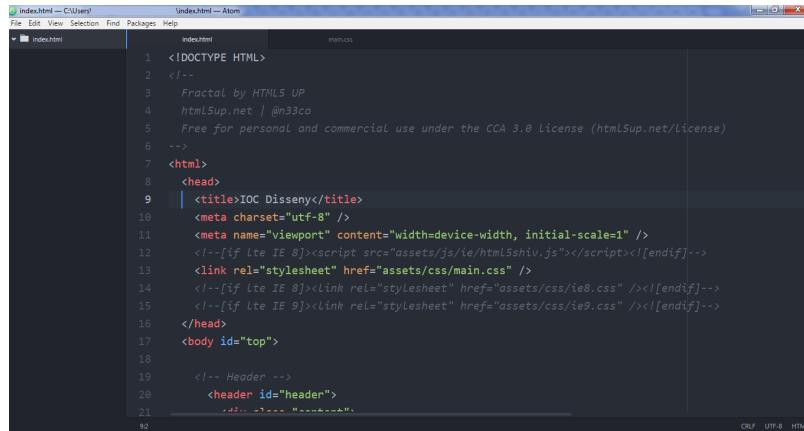
**FIGURA 1.12.** Interfície de treball de l'editor Sublime Text



**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



```

index.html -- C:\Users\1 index.html -- Atom
File Edit View Selection Find Packages Help
index.html main.css
1 <!DOCTYPE HTML>
2 <!--
3   Fractal by HTML5 UP
4   html5up.net | @n33co
5   Free for personal and commercial use under the CCA 3.0 License (html5up.net/License)
6 -->
7 <html>
8   <head>
9     <title>IOC Disseny</title>
10    <meta charset="utf-8" />
11    <meta name="viewport" content="width=device-width, initial-scale=1" />
12    <!--[if lt IE 8]><script src="assets/js/ie/html5shiv.js"></script><!--[endif]-->
13    <link rel="stylesheet" href="assets/css/main.css" />
14    <!--[if lt IE 8]><link rel="stylesheet" href="assets/css/ie8.css" /><!--[endif]-->
15    <!--[if lt IE 9]><link rel="stylesheet" href="assets/css/ie9.css" /><!--[endif]-->
16  </head>
17  <body id="top">
18    <!-- Header -->
19    <header id="header">
20      
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92

```

#### 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

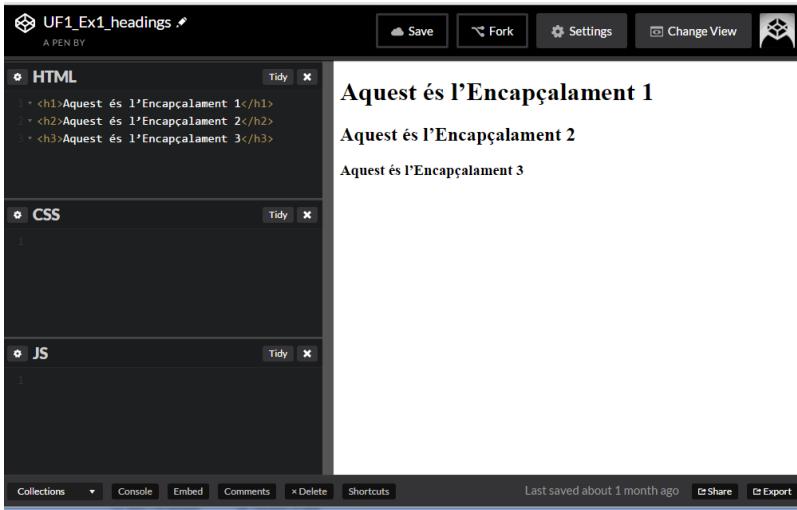
A l'hora de realitzar el disseny d' una pàgina web es pot optar per l'ús d'eines en línia que permeten escriure el codi directament des del navegador, com poden ser Codepen, Online HTML Editor o Real-Time HTML Editor, entre molts altres. Les seves característiques principals són les següents:

**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 intuïtives 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

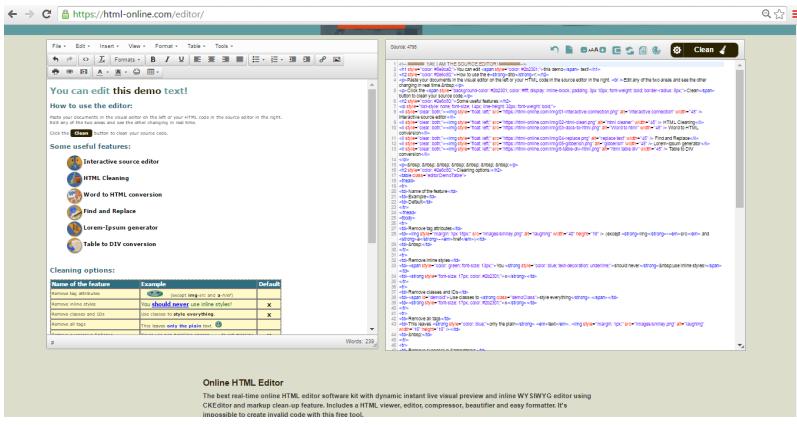


Interfície gràfica de Codepen

**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.

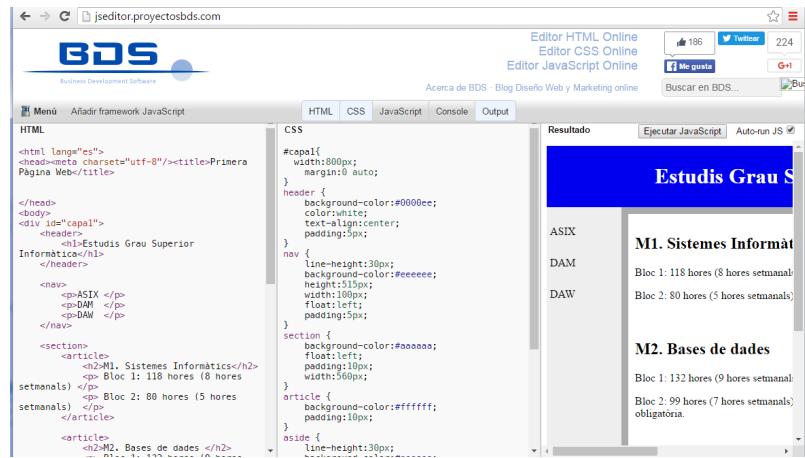
**FIGURA 1.15.** Utilització d'Online HTML Editor



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

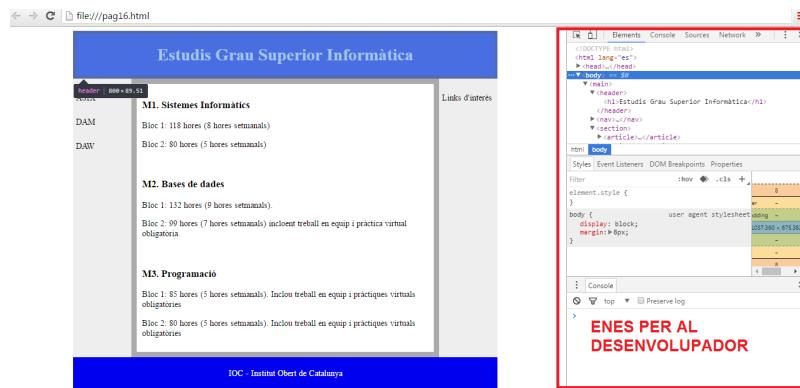
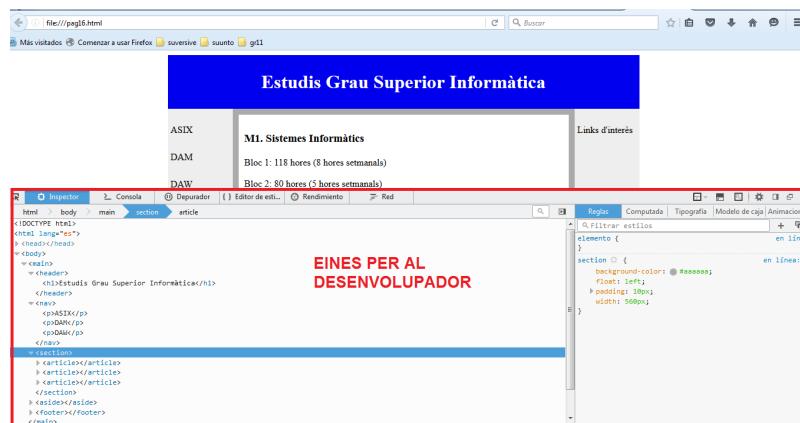
Els navegadors més importants ofereixen una sèrie d'eines que faciliten el desenvolupament d'un lloc web. D'aquesta manera, Chrome ofereix les **Google Chrome Dev Tools**, i Firefox permet incloure l'extensió **Firebug**.

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.

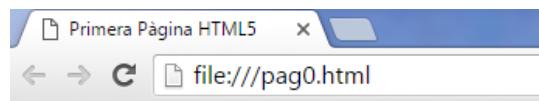
#### Estructura d'una pàgina web en HTML5

```

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 enllot 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.

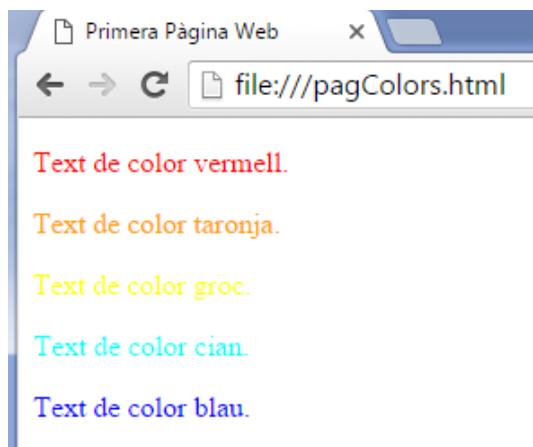
El resultat del següent exemple el podeu visualitzar en la figura 1.22.

#### 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 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;"
```

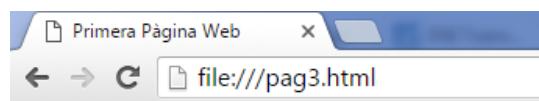
On *proprietat* fa referència a una propietat CSS i *valor* fa referència a un valor CSS.

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

#### 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

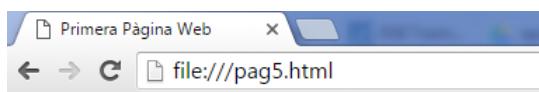
```

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

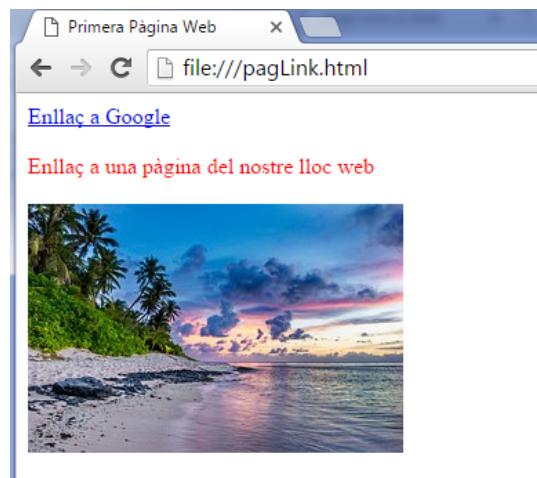
```

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.

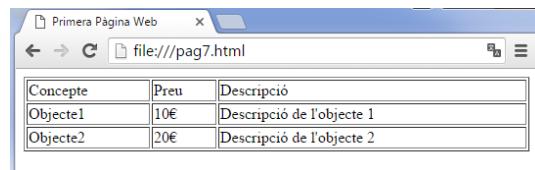
#### Exemple de taula

```

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>
  </tr>
</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

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 Alumne3</td>
32  <td>6</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/ioc-daw-m09/pen/MKBGOy?editors=1000](http://codepen.io/ioc-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.

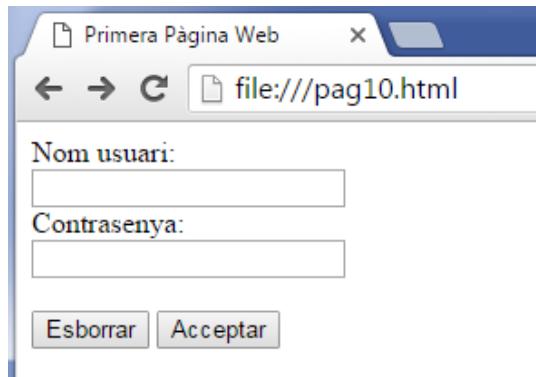
L'element HTML que permet crear formularis és *<form>*, i en el seu contingut s'hi defineixen els diferents camps que en formen part.

### 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>autocomplementar</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".
- Date input field for "Data de naixement" with placeholder "dd / mm / aaaa".
- Gender selection with radio buttons for "Home" (selected) and "Dona".
- Hobbies selection with checkboxes for "Esport", "Cinema", "Lectura", and "Altres".
- Color selection with a "color" input field and a preview box.
- File selection for "Foto" with a button "Tria un fitxer" and a 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

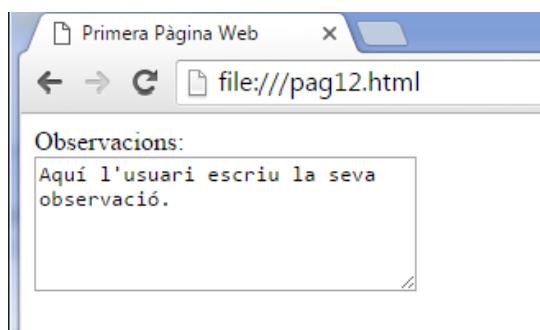
```

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>

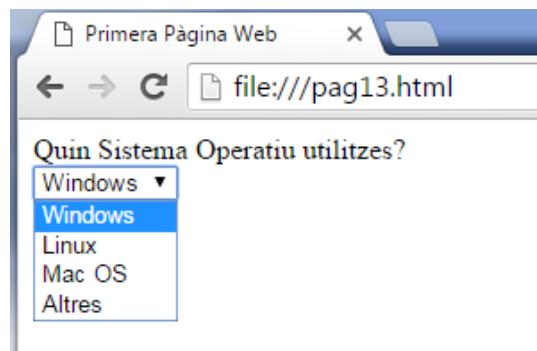
```

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-dawm09/pen/gPdOrg?editors=1000](https://codepen.io/loc-dawm09/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>

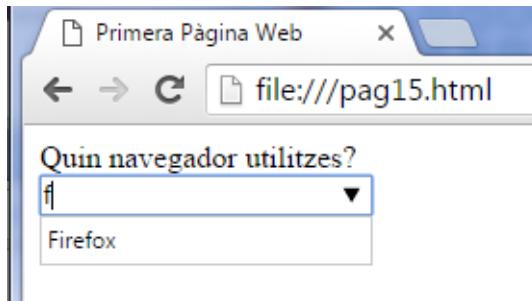
```

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.

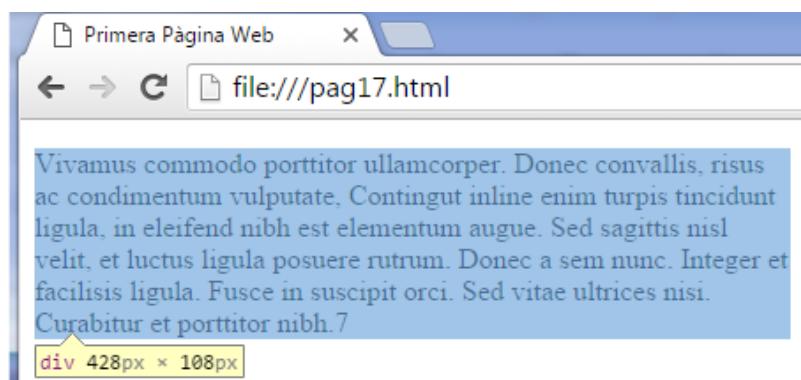
### Exemple d'element "block" i "inline"

```

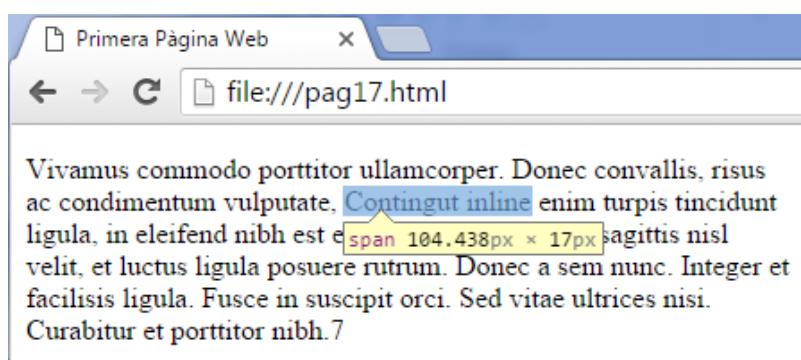
1  <div>
2  <p>
3  Vivamus commodo porttitor ullamcorper. Donec convallis, risus ac
4      condimentum vulputate,
5  <span>Contingut inline</span>
6  enim turpis tincidunt ligula, in eleifend nibh est elementum
7      augue. Sed sagittis nisl velit,
8  et luctus ligula posuere rutrum. Donec a sem nunc. 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.

---

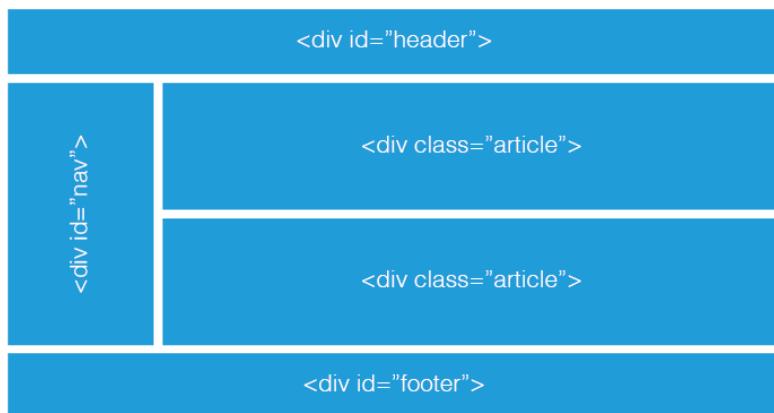
- Element <header>: permet inserir una zona de visualització per a les capçaleres.
- Element <footer>: permet inserir una zona de visualització per als peus de pàgina.
- Element <nav>: permet inserir una zona de visualització per a la navegació amb vincles hipertext.
- Element <section>: permet agrupar 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.

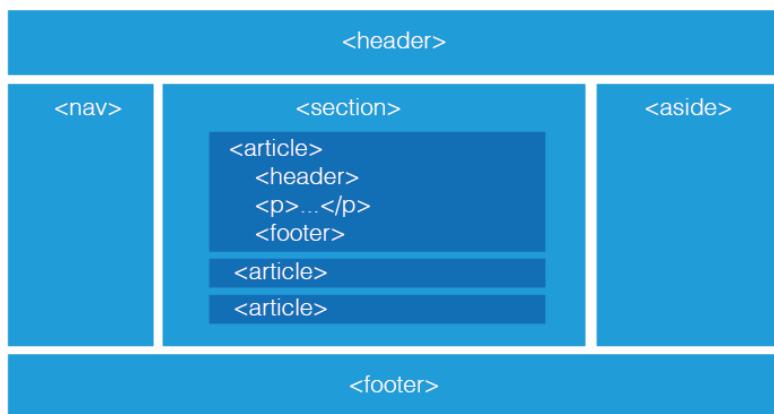
Cal mencionar que en HTML5 continua sent de molta utilitat l'ús de les caixes `<div>`.

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.

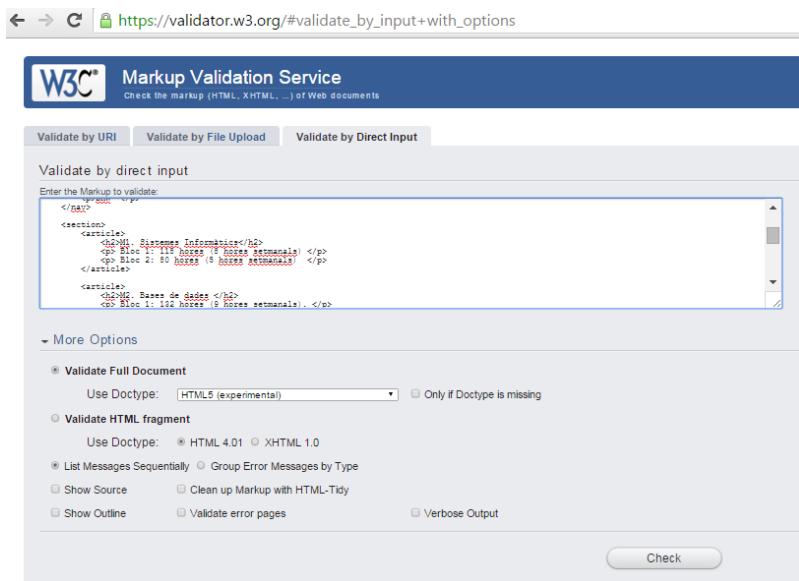
L'eina en línia accessible des de [validator.w3.org](https://validator.w3.org) permet validar el codi HTML de diferents maneres:

- 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).

## 2. Creació i aplicació d'estils a la interfície web. CSS3

Per crear el contingut i donar estructura i semàntica a una web s'utilitza el llenguatge HTML. Tot i això, no tan sols és important el contingut, sinó que també ho són, i molt, el disseny i la presentació de la informació. És a dir, la utilització del color, de la tipografia, de la imatge i dels elements interactius que fan la interfície agradable per a l'usuari que accedeix al web.

Així, per millorar l'aspecte i el manteniment d'un lloc web és indispensable la utilització d'estils en llenguatge CSS, Cascading Style Sheets. Aquest llenguatge controla la presentació dels documents HTML de tal manera que permet separar la presentació del contingut de la pàgina web i li aporta un format al document en HTML molt més ric en disseny. Actualment, CSS està suportat per tots els navegadors web.

### 2.1 Fulls d'estils: CSS3

Els fulls d'estils en cascada o **CSS** corresponen a un conjunt de regles usades per definir i crear la presentació d'un document escrit en HTML. L'organisme encarregat de regular la seva especificació és el World Wide Web Consortium (W3C)

L'estàndard actual de W3C per a l'aplicació d'estils és CSS3, i és totalment compatible amb les seves versions anteriors.

Podeu consultar l'estàndard CSS3 a:  
[www.w3.org/TR/CSS3](http://www.w3.org/TR/CSS3).



Logotip del CSS3

El llenguatge **CSS** indica al navegador com s'ha de visualitzar el contingut de la pàgina web.

Els avantatges que ofereix la utilització de fulls d'estil són:

- Separar l'estructura de les pàgines i el seu contingut (en HTML), del format del text i de la pàgina (en CSS).
- Tenir moltes més possibilitats de format i de presentació d'una pàgina web.
- Unificar el disseny de les pàgines web del lloc web, per així definir un estil una sola vegada i poder aplicar-lo tantes vegades com es vulgui.
- Reutilització dels mateixos fulls d'estil per a diferents documents HTML.
- Estàndard de W3C, que ofereix compatibilitat amb versions anteriors i amb els diferents navegadors.

## 2.1.1 Incloure estils CSS en un document HTML

Per crear un document web amb estils CSS es parteix d'un document HTML amb els continguts etiquetats per designar la funció de cada element dins del document, és a dir, els encapselaments, els paràgrafs, les imatges, els enllaços i altres elements. Seguidament, s'utilitza el llenguatge CSS per definir l'aspecte de cada element HTML, com el color, la mida, el tipus de lletra i la posició.

De fet, els estils es poden crear amb el mateix editor que s'utilitza per crear el document HTML, com pot ser Brackets o Codepen.

A més a més, els estils CSS d'un document web es poden definir en un document de text extern amb l'estensió .css, o bé també es poden definir al mateix document HTML.

### Enllaçar un full d'estils extern

Un **full d'estils extern** és un document de text amb extensió .css en el qual hi ha definits els estils d'un document web.

#### Exemple de contingut d'un document CSS

```
1 /* Comentari del document CSS */
2
3 h1 {
4     color: #FFFFFF;
5     background-color: #999999;
6 }
7
8 p {
9     font-family: Georgia, Helvetica;
10 }
```

Per enllaçar un document CSS a una pàgina web s'utilitza l'etiqueta <link>, la qual s'especifica en la capçalera, <head>, de la pàgina web.

És aconsellable utilitzar un full d'estils extern, ja que permet la seva reutilització i facilita el manteniment del lloc web. Pel fet de tenir els estils centralitzats en un o uns quants documents, qualsevol modificació resulta més senzilla.

#### Exemple d'enllaç d'un full d'estils extern

```
1 <html>
2 <head>
3     <link rel="stylesheet" href=".//css/estils.css" />
4     ...
5 </head>
6 ...
7 </html>
```

Els atributs de l'element link més rellevants es poden consultar en la taula 2.1.

**TAULA 2.1.** Atributs de l'element link

Atribut	Descripció
href	Especifica la ruta del document .css.
rel	Defineix la relació entre els documents .css i .html. Pot ser stylesheet o bé alternate.
type	Indica al navegador el tipus de recurs. En aquest cas: text/css.
title	Especifica un títol al recurs.
media	Especifica els medis en els quals s'aplica en full d'estils.

## Incrustar un full d'estils a HTML

Un full d'estils pot incrustar-se en el document HTML amb l'element <style>, el qual es sol definir a la capçalera (<head>) del document web.

### Exemple d'estils incrustats

```

1 <head>
2 ...
3 <style type="text/css" media="screen">
4   body { background: url(foo.gif) red; color: black; }
5   p em { background-color: yellow; color: black; }
6   .nota { margin-left: 5em; margin-right: 5em; }
7 </style>
8 </head>
```

A l'exemple anterior es pot veure com l'element <style> també admet atributs com **type** i **media**.

S'aconsella fer servir un **estil incrustat** quan un únic document té un únic estil. Però si els mateixos estils s'utilitzen en diferents pàgines web, llavors seria més apropiat utilitzar un full d'estils extern.

## Importar un full d'estils

Un full d'estils extern pot ser importat amb la regla @import de CSS, que es pot especificar tant a l'inici d'un fitxer .css com al principi de l'element <style>.

### Exemple d'importació defulls d'estils

```

1 <style type="text/css" media="screen, projection">
2   @import url(http://elmeudomini.com/css/estils.css);
3   @import url(/css/estils2.css);
4   p { background-color: yellow; color: black; }
5 </style>
```

## Estils en línia

Per especificar estils en línia s'utilitza l'atribut `style`, que pren per valor qualsevol nombre de propietats CSS separades per ";".

### Exemple d'estils en línia

```

1  <p style="color: red; font-family: 'New Century Schoolbook',
           serif">
2  Aquest paràgraf mostra el text vermell i amb la font New Century
   Schoolbook, si està disponible.
3  </p>
```

Els estils en línia perdencen molts dels avantatges dels fulls d'estil en barrejar el contingut amb la presentació. A més a més, els estils en línia s'apliquen a tots els tipus dispositius de sortida (ordinadors, tauletes, mòbils...), la qual cosa provoca que es desquadri un disseny adaptatiu. Així doncs, és per aquests motius que s'haurien d'usar amb moderació.

### 2.1.2 Característiques de CSS: cascada i herència

Dues de les característiques que fan que els fulls d'estil tinguin un ampli ventall de possibilitats són la cascada i l'herència. La **cascada** fa referència a la possible combinació de diferents fulls d'estil, mentre que l'**herència** es refereix a la capacitat que tenen els elements del document HTML d'heretar propietats dels seus elements antecessors.

### Fulls d'estil en cascada

A l'hora de visualitzar una pàgina web, el navegador ha d'interpretar els diferents estils definits per a cada element HTML. Aquests poden estar especificats en diferents llocs:

- Estils predeterminats del navegador.
- Estils específicats per l'usuari final.
- Estils relacionats amb el document especificats pel seu autor. Aquests es poden definir en tres llocs:
  - En un fitxer extern.
  - Al principi del document, a través de l'element `<style>`.
  - En un element HTML específic, a través del seu atribut `style`.

Així, hi ha tres fonts principals d'informació d'estils formant una cascada, de tal manera que els estils de l'usuari modifiquen o són prioritaris als que defineixi per defecte el navegador, i els estils de l'autor del document seran prioritaris als anteriors.

De la mateixa manera, si la definició d'un estil d'un autor entra en conflicte amb la definició d'un altre, s'aplicarà l'últim que estigui definit. És a dir, un estil definit a través de l'atribut `style` serà prioritari respecte als definits a través de l'element `<style>`.

#### Exemple d'aplicació d'estils en cascada

```

1 <!doctype html>
2 <html>
3 <head>
4 ...
5 <style>
6   p {
7     color: red;
8   }
9 </style>
10 </head>
11 <body>
12   <h1>Encapçalament</h1>
13   <p style="color: blue">Un paràgraf</p>
14   <p>Un altre paràgraf</p>
15 </body>
16 </html>

```

En podeu veure el resultat en la figura 2.1.

**FIGURA 2.1.** Figura d'estil en cascada



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

Si us hi fixeu, el text del primer paràgraf és blau, tal com s'indica a l'atribut `style`, enlloc de vermell definit a l'estil per als paràgrafs al `<head>` de la pàgina. Com veieu, doncs, el segon paràgraf ja apareix de color vermell.

#### L'herència als fulls d'estil

Atès que tots els elements d'una pàgina HTML, amb excepció de l'element arrel `<html>`, estan continguts en un altre element, s'ha de tenir en compte que tot element hereta les propietats dels seus antecessors.

Malgrat tot, s'han de tenir en compte les següents consideracions:

- No totes les propietats s'hereten. Aquesta característica es pot consultar a: [www.w3.org/TR/css-2010/#properties](http://www.w3.org/TR/css-2010/#properties).

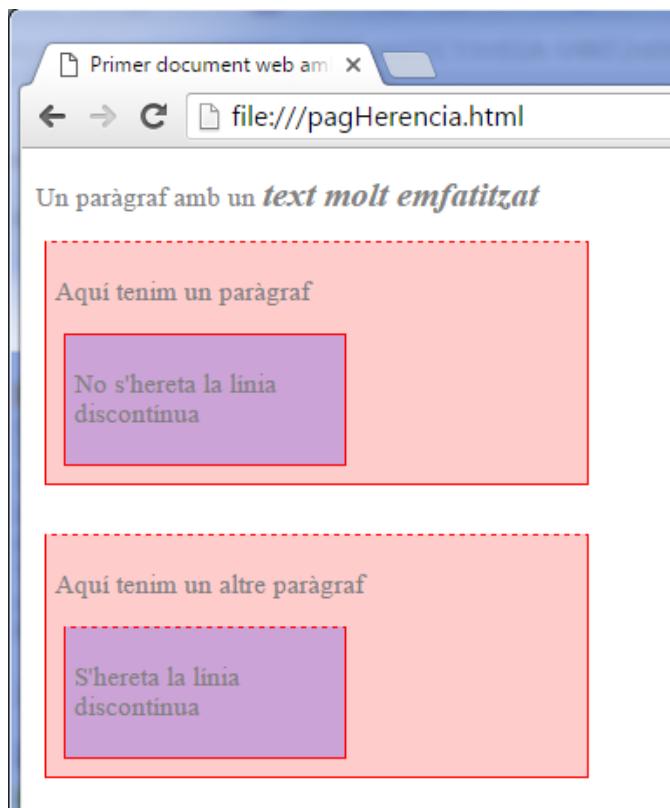
- Si es vol forçar l'herència d'una propietat d'un element que per defecte no hereta es pot utilitzar el valor `inherit`.
- Si s'especifica un valor a una propietat, aquest valor preval sobre el valor heretat.

### Exemple d'utilització de l'herència del llenguatge CSS

```

1  <!doctype html>
2  <html>
3  <head>
4  ...
5  <style>
6  body {
7      color: gray;
8  }
9
10 p {
11     font-size: 15px;
12 }
13
14 strong {
15     font-style: italic;
16     font-size: 20px;
17 }
18
19 div {
20     border-style: solid;
21     border-width: 1px;
22     border-color: red;
23     background-color: rgba(255, 0, 0, 0.2);
24     margin: 5px;
25     width: 300px;
26     padding: 5px;
27 }
28 </style>
29 </head>
30 <body>
31 <p>Un paràgraf amb un <strong>text molt emfatitzat</strong></p>
32
33 <div style="border-top-style: dashed;">
34     <p> Aquí tenim un paràgraf</p>
35     <div style="background-color:rgba(0,0,255,0.2);width:50%;">
36         <p>No s'hereta la línia discontínua</p>
37     </div>
38 </div>
39 <br />
40 <div style="border-top-style: dashed;">
41     <p>Aqui tenim un altre paràgraf</p>
42     <div style="border-top-style: inherit;background-color:rgba
43         (0,0,255,0.2);width:50%;">
44         <p>S'hereta la línia discontínua</p>
45     </div>
46 </div>
47 </body>
</html>
```

Ho podeu veure en la figura 2.2.

**FIGURA 2.2.** Herència a CSS

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

Fixeu-vos com el color del text és gris, ja que el paràgraf, en estar contingut dins de l'element <body>, hereta els seus estils. Alhora, l'element <strong> hereta els estils dels elements <p> i <body>.

Fixeu-vos també com en la primera caixa blava que es troba a dins de la vermella no hereta l'atribut border-top-style, i com a la segona caixa blava s'ha forçat l'herència de la vermella, que la conté amb el valor inherit.

### 2.1.3 Format d'una regla CSS

Per definir un estil específic, CSS fa servir una regla que consisteix en un selector i un bloc on es declaren les diferents propietats que ha de tenir l'estil.

```
selector { propietat: valor; propietat: valor; ... }
```

On *selector* fa referència a l'element que s'aplica l'estil.

On les diferents propietats es declaren en el bloc separades per ";".

#### Exemple de definició d'un estil

```
1 h1 {font-size:10px; color:blue; text-align:center;}
```

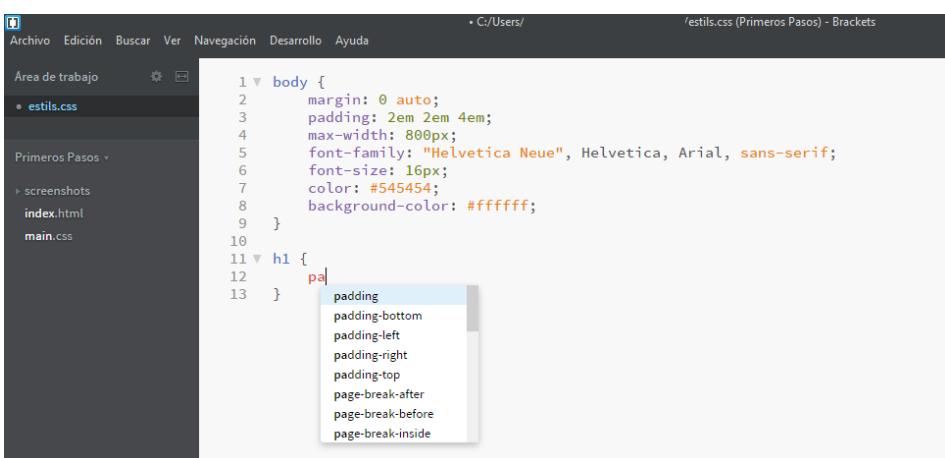
L'estil definit en l'exemple anterior estableix que les capçaleres h1 tindran el següent aspecte:

- La mida de la font serà de 10 píxels.
- El color de la font serà blau.
- El text es veurà centralitzat.

Per definir els estils es pot utilitzar qualsevol editor de text, per exemple el mateix que s'empra per crear els documents HTML, com poden ser Brackets, Codepen, Sublime, etc.

Vegeu en la figura 2.3 com es poden escriure els estils en l'editor Brackets.

**FIGURA 2.3.** Creació d'estils amb l'editor Brackets



The screenshot shows the Brackets IDE interface. The top menu bar includes Archivo, Edición, Buscar, Ver, Navegación, Desarrollo, and Ayuda. The address bar shows the path C:/Users/.../estilos.css (Primeros Pasos) - Brackets. The left sidebar lists files: 'Área de trabajo' contains 'estilos.css'; 'Primeros Pasos' contains 'screenshots', 'index.html', and 'main.css'. The main workspace shows the following CSS code:

```

1  body {
2      margin: 0 auto;
3      padding: 2em 2em 4em;
4      max-width: 800px;
5      font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
6      font-size: 16px;
7      color: #545454;
8      background-color: #ffffff;
9  }
10
11  h1 {
12      pa| // A red placeholder text 'pa' is shown at the cursor position.
13  }

```

A tooltip box appears over the cursor, listing CSS properties: padding, padding-bottom, padding-left, padding-right, padding-top, page-break-after, page-break-before, and page-break-inside. The status bar at the bottom right says 'Veiem com l'editor Brackets ofereix suggeriments a l'hora d'escriure una propietat CSS.'

## 2.1.4 Selectors

### Selectors W3C

Podeu consultar els diferents tipus de selectors a: [goo.gl/9mY7Fw](http://goo.gl/9mY7Fw).

La funció dels selectors és indicar a quins elements s'aplicaran els estils definits. Els diferents tipus de selectors es presenten en la taula 2.2.

**TAULA 2.2.** Selectors bàsics

Sintaxi	Descripció
*	Selector universal. Permet aplicar un estil a qualsevol tipus.
nomElement	Selector de tipus. Permet aplicar un estil a un element HTML concret.
.nomClasse	Selector de classe. Permet aplicar un estil als elements que utilitzin aquesta classe.
#idNom	Selector ID. Permet aplicar un estil a un element que tingui un codi d'identificació únic.
selector[atribut]	Selector d'atribut. Permet aplicar un estil a un element que utilitzi un atribut específic.

Així, per exemple, es poden tenir els següents estils definits en un full d'estils:

#### Exemples d'estils amb diferents selector

```

1 * {
2   font-size: 15px;
3   font-family: Arial;
4 }
5
6 h1.ressaltat {
7   font-weight: bolder;
8   font-size: 25px;
9   color: white;
10  background-color: #ff80c0;
11 }
12
13 p,h2 {
14   text-align:center;
15 }
16
17 .textBlau {
18   color: blue;
19 }
20
21 #estil1 {
22   font-size: larger
23 }
24
25 a[target] {
26   background-color: yellow;
27 }
```

#### Utilització dels estils anteriors

```

1 <h1 class="ressaltat">Títol ressaltat</h1>
2
3 <h2> Els subtítols apareixen centrats</h2>
4
5 <p> Els paràgrafs apareixen centrats igual que els subtítols.</p>
6
7 <p class="textBlau">Aquest paràgraf és blau. </p>
8
9 <p id="estil1">Aquest estil només s'utilitza una vegada en el
10 document html</p>
11
12 <h1>Títol sense ressaltar</h1>
13 <a href="http://ioc.xtex.cat">IOC</a>
14 <br/>
15 <a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>
16 >
```

El resultat es pot veure en la figura 2.4.

**FIGURA 2.4.** Estils selectors

Podeu provar aquest exemple en l'enllaç següent: [codepen.io/ioc-dawm09/pen/pyrBaP](https://codepen.io/ioc-dawm09/pen/pyrBaP).

Algunes consideracions de l'exemple anterior són:

- L'atribut `class` s'utilitza per aplicar un estil de classe a l'element.
- L'atribut `id` s'utilitza per aplicar un estil `id`, únic per a un element.
- Per defecte, el text de tots els elements serà Arial de 15 px.
- Els *links* que tinguin l'atribut `target` apareixen ressaltats.
- Si es vol definir el mateix estil per a diferents selectors es poden agrupar separats per una coma.

En la taula 2.3 es poden consultar els diferents tipus de selectors d'atribut, ja que en aquests s'hi poden especificar quins valors haurien de tenir.

**TAULA 2.3.** Selectors d'atribut

Sintaxi	Descripció
<code>[atribut]</code>	Selecciona l'element amb aquest atribut.
<code>[atribut="valor"]</code>	El valor de l'atribut és valor.
<code>[atribut^="text"]</code>	El valor de l'atribut comença per text.
<code>[atribut\$="text"]</code>	El valor de l'atribut acaba amb text.
<code>[atribut*= "text"]</code>	El valor de l'atribut conté el text.
<code>[atribut~= "valor"]</code>	El valor de l'atribut és una llista de paraules separades per espais, una de les quals és exactament valor.
<code>[atribut = "valor"]</code>	El valor de l'atribut és una llista de paraules separades per guions, començant per valor.

Selecciona l'element amb aquest atribut.

## Combinacions de selectors

Els selectors es poden combinar per afegir especificacions a les regles CSS. En la taula 2.4 es poden visualitzar algunes de les possibles combinacions dels selectors.

**TAULA 2.4.** Combinacions de selectors

Sintaxi	Descripció
<b>A + B</b>	Selectors adjacents. Permet aplicar un estil a l'element que es trobi a continuació (B) d'un element específic (A).
<b>A ~ B</b>	Selectors general de germans. Permet aplicar un estil a l'element (B) que sigui germà de l'element (A).
<b>A &gt; B</b>	Selectors de fills. Permet aplicar un estil al primer element fill (B) d'un element específic (A).
<b>A B</b>	Selectors descendents. Permet aplicar un estil a un element determinat que es trobi a dins d'un element específic.

### Exemple de definició d'estils combinats.

```

1  /* selector general de germans */
2  h1 ~ h2 {
3      color: red;
4  }
5
6  /* selector de germans adjacents */
7  h1 + h2 {
8      color: blue;
9  }
10
11 /* selector de descendents*/
12 div h2 {
13     color: green;
14 }
15
16 /*selector de fills*/
17 div > p {
18     color: orange;
19     font-weight: bold;
20 }
```

### Exemple d'utilització dels estils anteriors:

```

1 <h1>Títol 1</h1>
2 <h2>Subtítol 1 és blau</h2>
3 <p>El subtítol 1 està definit immediatament després del títol 1 i
   també és germà.</p>
4 <hr/>
5 <h1>Títol 2</h1>
6 <p>Aquí tenim un paràgraf qualsevol.</p>
7 <h2>Subtítol 2 és vermell</h2>
8 <p>El subtítol 2 és germà del títol 2 però no és consecutiu.</p>
9 <hr/>
10 <h1>Títol 3</h1>
11 <div>
12   <h2>Subtítol 3 és verd</h2>
13   <p>El subtítol 3 NO és germà del títol 3 anterior. I aquest par
      àgraf és taronja.</p>
14   <p> Aquest paràgraf també és fill de l'etiqueta &lt;div&gt; i é
      s taronja.</p>
15   <span><h2>Subtítol 3.1 és verd</h2></span>
16   <span><p>El subtítol 3.1 també és descendant.</p></span>
```

```

17  <span><p>Aquest paràgraf no és fill de l'etiqueta &lt;div&gt; i
     no es veu taronja.</p></span>
18  </div>
19  <h2>Subtítol 4 és vermell</h2>
20  <p>El subtítol 4 és germà del títol 3 però no és consecutiu.</p>

```

FIGURA 2.5. Combinació de selectors



Podeu provar aquest exemple en l'enllaç: [codepen.io/ioc-dawm09/pen/BKdeKW](https://codepen.io/ioc-dawm09/pen/BKdeKW).

Fixeu-vos en els següents punts:

- El primer subtítol h2 és germà adjacent de h1 (immediatament després del títol h1) i, per tant, es veu de color blau enllloc de vermell. En canvi, el subtítol 2 es veu vermell perquè no està definit immediatament després del títol h1.
- Els subtítols h2 descendents de h1 apareixen de color verd, tal com es veu en el subtítol 3 i 3.1.
- Els paràgrafs que siguin fills directes de l'element div es veuen de color taronja.

## Pseudoclasses

Les pseudoclasses s'afegeixen al selector de manera que es pugui aplicar un format determinat a l'element seleccionat.

La manera de definir estils utilitzant pseudoclasses és:

```

1 selector:pseudoclasse{
2   propietat: valor;
3   ...
4 }
```

En la taula 2.5 s'especifiquen les pseudoclasses per als enllaços.

**TAULA 2.5.** Pseudoclasses d'acció d'usuari i d'enllaç

Pseudoclasse	Descripció
:link	Permet definir l'estil dels enllaços quan encara no s'han visitat.
:visited	Permet definir l'estil dels enllaços visitats.
:active	Permet definir l'estil dels elements quan s'activen (quan es prem el ratolí damunt seu).
:hover	Permet definir l'estil dels elements quan es passa el ratolí damunt seu.
:focus	Permet definir l'estil dels elements quan reben el focus.

### Exemple d'estils per als enllaços

```

1 /* Els enllaços (visitats i no visitats) són de color blau amb el
   fons gris i sense subratllar. */
2 a:link,
3 a:visited {
4   color: blue;
5   background-color: gray;
6   text-decoration: none;
7 }
8
9 /* Quan es passa per damunt tindran el fons blau i text blanc */
10 a:hover {
11   color: white;
12   background-color: blue;
13 }
14
15 /* Quan es fa clic abans de soltar es faran més grans */
16 a:active {
17   font-size: larger;
18   color: white;
19   background-color: blue;
20 }
21
22 /* Quan se situa en un control de formulari <input> tindrà
23 una vora vermella puntejada, i un farciment de 2px */
24 input:focus {
25   border: red 2px dotted;
26   padding: 2px;
27 }
```

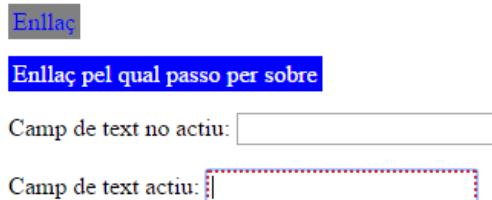
### Utilització d'estils per als enllaços

```

1 <p><a href="http://fsf.org">Enllaç</a></p>
2
3 <p><a href="http://gnu.org">Enllaç pel qual passo per sobre</a></
   p>
4
5 <p>Camp de text no actiu: <input type="text" name="input" /></p>
6
7 <p>Camp de text actiu: <input type="text" name="input" /></p>
```

El resultat que s'obté es pot veure en la figura 2.6.

**FIGURA 2.6.** Estils dels enllaços



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

Podeu provar aquest exemple en el següent enllaç: [codepen.io/ioc-dawm09/pen/MyEWvW](https://codepen.io/ioc-dawm09/pen/MyEWvW).

Existeixen altres pseudoclasses que apliquen els estils fent referència a l'estructura del DOM. Vegeu-les en la taula 2.6.

**TAULA 2.6.** Pseudoclasses estructurals

Pseudoclasse	Descripció
:root	Representa un element que es troba a l'arrel del document.
:nth-child (num)	Selecciona l'element indicat però amb la condició que sigui el fill enèsim del seu pare.
:nth-last-child (num)	Idèntic a l'anterior, però el número indicat es comença a comptar des de l'últim fill.
:nth-of-type (num)	Selecciona l'element indicat però amb la condició que sigui l'enèsim element fill d'aquest tipus.
:nth-last-of-type (num)	Idèntic a l'anterior, però el nombre indicat es comença a comptar des de l'últim fill.
:first-child	Selecciona un element que sigui el primer fill d'un altre element.
:last-child	Selecciona l'element indicat, però amb la condició que sigui l'últim fill del seu element pare.
:first-of-type	Fa referència al primer element d'aquest tipus en l'element pare.
:last-of-type	Fa referència a l'últim element d'aquest tipus en l'element pare.
:only-of-type	Fa referència a aquell element que és l'únic fill del seu pare.
:empty	Selecciona l'element indicat però amb la condició que no tingui cap fill i tampoc pot tenir cap contingut de text.

En la taula 2.7 hi trobareu altres pseudoclasses.

**TAULA 2.7.** Altres pseudoclasses

Pseudoclasse	Descripció
:lang	D'idioma. Permet aplicar estils en funció de l'estil especificat.
:not	De negació. Selecciona tots els elements que no compleixen amb la condició d'un selector.

#### Exemple d'estils fent referència a l'estructura DOM

```

1  /*Selecciona els elements li descendents de les llistes de tipus
   especial */
2  ul.especial li {
3      display: inline;
4      padding: 5px;
5  }
6
7  /* selecciona el primer fill de la llista*/
8  ul.especial li:first-child {
9      color: blue;
10 }
11
12 /*selecciona els elements imparells de la llista */
13 ul.especial li:nth-child(odd) {
14     background-color: grey;
15 }
16
17 /*selecciona els elements imparells de la llista */
18 ul.especial li:nth-child(even) {
19     background-color: pink;
20 }
21
22 /*selecciona el quart fill de la llista*/
23 ul.especial li:nth-of-type(4) {
24     color: white;
25 }
26
27 /*selecciona els paràgrafs que no tenen germans paràgrafs*/
28 p:only-of-type {
29     background-color: grey;
30     color: #ffa0ff;
31     font-weight: bold;
32     font-size: 25px;
33 }
34
35 /*Els paràgrafs en anglès apareixen en itàlic */
36 p:lang(en){
37     font-style:italic;
38 }
39
40 :root {
41     font-family:arial;
42 }
```

#### Exemple d'utilització d'estils que fan referència a l'estructura DOM

```

1 <div>
2     <p>Menú especial:</p>
3
4     <ul class="especial">
5         <li> Element 1</li>
6         <li> Element 2</li>
7         <li> Element 3</li>
```

```

8   <li> Element 4</li>
9   </ul>
10
11  <div>
12    <p>Un paràgraf</p>
13    <p lang="en">This paragraph is in english.</p>
14  </div>
15 </div>

```

El resultat es pot veure en la figura 2.7.

**FIGURA 2.7.** Visualització de l'aplicació d'estils d'estructura DOM

### Menú especial:

Element 1 Element 2 Element 3 Element 4

Un paràgraf

*This paragraph is in english*

Podeu provar l'anterior exemple en l'enllaç següent: [codepen.io/ioc-dawm09/pen/RajWyj](http://codepen.io/ioc-dawm09/pen/RajWyj).

Podeu consultar les pseudoclasses que defineix CSS a: [developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes](https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes).

### Pseudoelements

A diferència de les pseudoclasses, els pseudoelements no descriuen l'estat d'un element, sinó que s'afegeixen a un selector per tal de definir estils a una part concreta del document HTML.

Els pseudoelements solen anar precedits per ":", però per compatibilitat amb versions anteriors també poden anar precedits per ":".

La manera de definir estils utilitzant pseudoelements és:

```

1 selector::pseudoElement{
2   propietat: valor;
3   ...
4 }
```

Podem consultar els pseudoelements que defineix CSS a [developer.mozilla.org/es/docs/Web/CSS/Pseudoelementos](https://developer.mozilla.org/es/docs/Web/CSS/Pseudoelementos). En la taula 2.8 podem veure la descripció d'alguns d'ells.

**TAULA 2.8.** Pseudoelements

Sintaxi	Descripció
::after	Permet introduir contingut al final de l'element. Cal afegir la propietat <code>content</code> amb el valor desitjat.
::before	Permet introduir contingut a l'inici de l'element. Cal afegir la propietat <code>content</code> amb el valor desitjat.
::first-letter	Permet definir l'estil de la primera lletra de l'element.
::first-line	Permet definir l'estil de la primera línia de l'element.

**TAULA 2.8** (continuació)

Sintaxi	Descripció
<code>::selection</code>	Permet definir l'estil d'allò que l'usuari ha seleccionat amb el ratolí.

A continuació, podeu veure un parell d'exemples:

#### Exemple d'estils CSS utilitzant pseudoelements

```

1 div::first-line { color: red; }
2 h1::before { content: "Títol "; }
3 p::after   { content: "."; }
4 p::first-letter { text-transform: uppercase; }
```

#### Exemple d'utilització d'estils utilitzant pseudoelements

```

1 <h1>1</h1>
2 <div>
3   <p>aquest és el primer paràgraf llarg i és vermell</p>
4   <p>aquest és el segon paràgraf i no és vermell.</p>
5 </div>
```

El resultat que obtindreu és el que es pot veure en la figura 2.8.

**FIGURA 2.8.** Visualització de l'aplicació d'estils utilitzant pseudoelements

## Títol 1

Aquest és el primer paràgraf llarg i és vermell.

Aquest és el segon paràgraf i no és vermell..

Aquest exemple el podeu provar en el següent enllaç: [codepen.io/ioc-daw-m09/pen/YqExdd?editors=1100](https://codepen.io/ioc-daw-m09/pen/YqExdd?editors=1100).

### 2.1.5 Unitats de mesura CSS

Algunes propietats CSS indiquen la mida de la lletra, l'amplada, els marges, etc. CSS classifica les unitats de mesura en dos tipus: **absolutes i relatives**.

En la taula 2.9 es poden visualitzar les unitats de mesura absolutes.

**TAULA 2.9.** Unitats de mesura absolutes

Unitat	Descripció
<b>cm</b>	Centímetres
<b>mm</b>	Mil·límetres
<b>in</b>	Polzades, <i>inches</i> . Equival a 2,54 cm
<b>pt</b>	Punts. Mesura que equival a 1/72 d'una polzada
<b>pc</b>	Piques. Mesura que equival a 12 pt

En la taula 2.10 es poden visualitzar les unitats de mesura relatives.

**TAULA 2.10.** Unitats de mesura relatives

Unitat	Descripció
<b>em</b>	Mesura relativa a la mida del tipus de lletra del seu contingut.
<b>rem</b>	<i>Root em.</i> Mesura relativa a la mida del tipus de lletra general. Té com a referència la unitat que fa servir l'element arrel ( <i>root</i> ), i no l'element contingut.
<b>px</b>	Píxels. Mesura relativa a la resolució de la pantalla.
<b>ex</b>	Mesura relativa a l'alçada de la lletra x. Canvia si es canvia la font que fem servir.
<b>ch</b>	Mesura relativa a l'alçada del caràcter numèric "0". Canvia si es canvia la font que fem servir.
<b>%</b>	Percentatge.

També es disposa d'unitats relatives al *viewport*, és a dir, a l'amplitud de la finestra del navegador. En la taula 2.18 es veuen aquestes propietats.

**TAULA 2.11.** Unitats de mesura relatives al 'viewport'

Unitat	Descripció
<b>vw</b>	<i>Viewport width.</i> 1 vw equival a (1/100) de l'amplada de la finestra, és a dir, a l'1%. Per reflectir un valor del 15% hauríem de posar 15 vw.
<b>vh</b>	<i>Viewport height.</i> 1 vh equival a (1/100) de l'alçada de la finestra, és a dir, a l'1%. Per reflectir un valor del 15% hauríem de posar 15 vh.
<b>vmin</b>	Funcionament similar als anteriors, però hi ha unaavaluació prèvia de quin eix, x o y (amplada, alçada),és menor, i selecciona com a referent l'eix que ho sigui.
<b>vmax</b>	Idèntic a l'anterior, però agafant com a referent l'eix que sigui més gran dels dos.

A continuació podeu veure alguns exemples:

#### Estils amb diferents tipus d'unitats de mesura

```

1 .mida1 {font-size: 16pt; }
2 .mida2 {font-size: 16px; }
3 .mida3 {font-size: 1.5em; }
4 .mida4 {font-size: 1.5rem; }
5 .mida5 {font-size: 1.5vw; }
```

#### Utilització d'estils amb diferents tipus d'unitats de mesura

```

1 <p class="mida1">Mida del text en punts (absoluta)</p>
2 <p class="mida2">Mida del text en píxels (relatiu a la resolució)
   </p>
3 <p class="mida3">Mida del text en em (relatiu al contingut) </p>
4 <p class="mida4">paràgraf mida en rem (relatiu a l'element root)<
   /p>
5 <p class="mida5">Mida del text en vw (relativa al viewport)</p>
6 <hr />
7 <div class="mida3">
8   <p class="mida1">Mida del text en punts (absoluta)</p>
```

```

9   <p class="mida2">Mida del text en píxels (relatiu a la resoluci
10  ó)</p>
11  <p class="mida3">Mida del text en em (relatiu al contingut) </
12  >
13  <p class="mida4">Mida del text en rem (relatiu a l'element root
14  )</p>
15  <p class="mida5">Mida del text en vw (relativa al viewport)</p>
16
17 </div>
```

El resultat que obtindreu es pot veure en la figura 2.9.

**FIGURA 2.9.** Visualització de l'aplicació d'estils amb diferents tipus de mesura

Mida del text en punts (absoluta)

Mida del text en pixels (relatiu a la resolució)

Mida del text en em (relatiu al contingut)

paràgraf mida en rem (relatiu a l'element root)

Mida del text en vw (relativa al viewport)

---

Mida del text en punts (absoluta)

Mida del text en pixels (relatiu a la resolució)

## Mida del text en em (relatiu al contingut)

Mida del text en rem (relatiu a l'element root)

Mida del text en vw (relativa al viewport)

L'anterior exemple el podeu provar en el següent enllaç: [codepen.io/ioc-dawm09/pen/KzyXdW](https://codepen.io/ioc-dawm09/pen/KzyXdW).

Encara que sovint s'utilitza la unitat de mesura píxel, una bona opció és “rem”, ja que especifica les mides relatives a l'element *root*, a diferència de “em”, la mida del qual pot costar de controlar perquè pot variar en funció d'on es trobi.

També podeu comprovar que si modifiquem la mida de la finestra del navegador, el text expressat en vw es redimensiona.

### 2.1.6 Propietats CSS

Hi ha múltiples atributs o propietats que es poden definir amb CSS. Per conèixer l'ampli ventall que ofereix CSS es poden consultar les següents webs:

- Developer Mozilla: [developer.mozilla.org/es/docs/Web/CSS/Referencia\\_CSS](https://developer.mozilla.org/es/docs/Web/CSS/Referencia_CSS)
- W3C: [www.w3.org/community/webed/wiki/CSS/Properties](https://www.w3.org/community/webed/wiki/CSS/Properties)

Atès que l'estàndard CSS3 està en contínua evolució, no tots els navegadors han adoptat algunes de les propietats, o si ho han fet, moltes vegades necessiten un

prefix. Aquests prefixos es faran servir quan així sigui necessari i es poden veure en la taula 2.12.

**TAULA 2.12.** Prefixos per als navegadors

Prefix	Navegador
<b>-moz-</b>	Prefix per al navegador Firefox.
<b>-ms-</b>	Prefix per al navegador Internet Explorer.
<b>-webkit-</b>	Prefix per als navegadors Chrome i Safari.
<b>-o-</b>	Prefix per al navegador Opera.

Tot i això, cal mencionar que no tots els navegadors es comporten de la mateixa manera davant del mateix full d'estil, i això es deu al fet que alguns no compleixen els estàndards establerts. Per això es pot recórrer a pàgines com [caniuse.com](#) per veure si determinades característiques ja han estat implementades, i per quins navegadors.

#### 'Shorthand properties'

Les propietats abreujades permeten establir els valors de diverses propietats CSS simultàniament. Aquestes propietats poden fer referència als marges, les vores, els farciments i també al fons del document web.

En la taula 2.13 es pot consultar un recull de les propietats referents als continguts de la pàgina web.

**TAULA 2.13.** Propietats per a la font, els paràgrafs, els fons i les taules

Propietats	
<b>Font</b>	<code>color, font-size, font-family, font-weight, font-style</code>
<b>Paràgrafs</b>	<code>line-height, text-decoration, text-align, text-indent, text-transform, text-shadow, text-overflow, text-wrap, list-style</code>
<b>Fons</b>	<code>background-color, background, background-image, background-origin, background-repeat, background-position, background-attachment</code>
<b>Taules</b>	<code>border-spacing, border-collapse, caption-side, empty-cells</code>

#### Estils amb CSS

```

1  /*Gradient de 180 graus que va de blau a blanc*/
2  .gradient1 {
3      background: linear-gradient(180deg, blue, white);
4  }
5
6  /*Gradient de 90 graus que va de granat a taronja més
    clar, la quantitat de cada color s'especifica en percentatge
    al costat del color.*/
7  .gradient2 {
8      background: linear-gradient(90deg, #660000 10%, #f80 30%, #ffc
9          60%);
10
11 /*Gradient radial circular que va de negre a blanc */
12 .gradient3 {
13     background: radial-gradient(circle, black, white);
14 }
15
16 /*Gradient el·líptic que va de negre a blanc*/
17 .gradient4 {
18     background: radial-gradient(ellipse, black, white);
19 }
```

```

20 /*Els paràgrafs apareixen indentats i tenen una ombra d'un píxel,
   separada 2px i grisa */
21 p {
22   text-indent: 50px;
23   text-shadow: 1px 2px #ccc;
24   padding: 50px;
25 }
26
27 /* Els títols tenen un farciment de 5px*/
28 h1{
29   padding:5px;
30 }
```

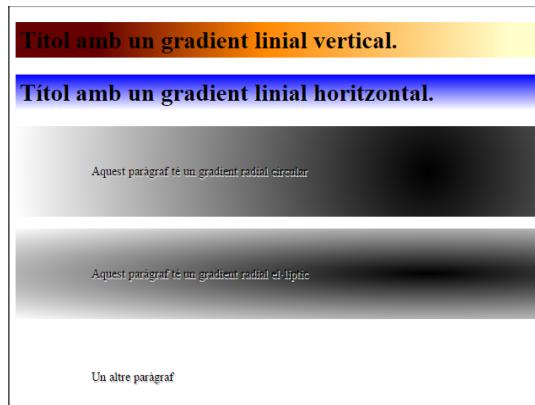
### Utilització d'estils CSS

```

1 <h1 class="gradient2">Títol amb un gradient lineal vertical.</h1>
2 <h1 class="gradient1">Títol amb un gradient lineal horitzontal.</
  h1>
3
4 <p class="gradient3">Aquest paràgraf té un gradient radial
  circular</p>
5
6 <p class="gradient4">Aquest paràgraf té un gradient radial el·líptic</p>
7 <p>Un altre paràgraf</p>
```

El resultat que obtindreu és el que es pot veure en la figura 2.10.

**FIGURA 2.10.** Visualització d'estils de gradient



Podeu provar l'exemple anterior en l'enllaç següent: [codepen.io/ioc-daw-m09/pen/NNwaoa?editors=1100](https://codepen.io/ioc-daw-m09/pen/NNwaoa?editors=1100).

A més a més, cal tenir en compte que a pesar que els navegadors presenten algunes configuracions similars, com ara el tipus de lletra serif de color negre, per exemple. Difereixen bastant en els valors de marges verticals dels encapçalamens (`margin-top` i `margin-bottom`), la tabulació esquerra dels elements d'una llista (`margin-left` o `padding-left`) o bé la interliniat (`line-height`).

### Exemple de visualització d'una pàgina web en diferents navegadors

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="utf-8"/>
5   <title>Primera Pàgina Web</title>
6   <body>
7     <h1>Títol</h1>
```

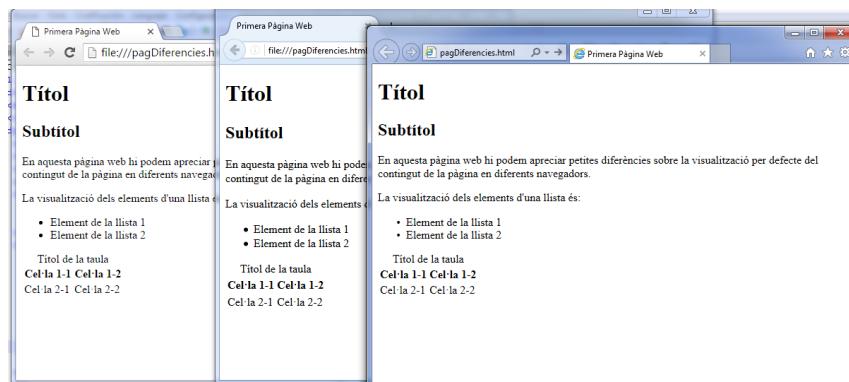
```

8 <h2>Subtítol</h2>
9 <p>En aquesta pàgina web hi podem apreciar petites diferències
   sobre la visualització per defecte del contingut de la pà-
   gina en diferents navegadors. </p>
10 <p>La visualització dels elements d'una llista és:
11 <ul>
12 <li>Element de la llista 1</li>
13 <li>Element de la llista 2</li>
14 </ul>
15 <table>
16 <caption>Títol de la taula</caption>
17 <tr>
18 <th>Cel·la 1-1</th>
19 <th>Cel·la 1-2</th>
20 </tr>
21 <tr>
22 <td>Cel·la 2-1</td>
23 <td>Cel·la 2-2</td>
24 </tr>
25 </table>
26 </body>
27 </html>

```

En la figura 2.11, d'esquerra a dreta, es poden veure les diferències de visualització de la mateixa pàgina web en els navegadors Chrome, Firefox i Internet Explorer.

**FIGURA 2.11.** Visualització en diferents navegadors



### Element "<div>"

Aquest element crea una caixa que per defecte ocupa tota l'amplada del navegador. S'utilitza per definir l'estil d'una secció d'una pàgina web.

### Element "<span>"

Aquest element crea una caixa que per defecte només ocupa l'amplada necessària. S'utilitza per definir l'estil d'una porció de contingut d'una secció d'una pàgina web.

## 2.1.7 El model caixa

Els navegadors, de manera automàtica, creen i col·loquen cada element HTML en una estructura que té format de caixa. Així, per defecte, la caixa dels elements HTML de tipus *block* ocupa tota l'amplada de la finestra, i la dels elements *inline* només agafen l'amplada necessària.

En principi, la caixa de cada element no és visible, ja que no mostra cap color de fons ni vora. No obstant això, es pot modificar amb CSS per canviar les seves dimensions, els colors, la vora i la posició.

Es poden modificar les propietats de la caixa de qualsevol element HTML, però generalment s'utilitza l'element *<div>* per organitzar i donar estructura al disseny de les pàgines web.

## Dimensions de la caixa

Les parts que componen una caixa en el seu ordre de visualització des del punt de vista de l'usuari són:

- **Contingut, content:** es referix al contingut de l'element; per exemple, el text, una imatge, etc.
- **Farciment, padding:** espai lliure opcional existent entre el contingut i la vora.
- **Vora, border:** línia que envolta el contingut i el farciment.
- **Imatge de fons, background image:** imatge que es mostra per darrere del contingut i el farciment.
- **Color de fons, background color:** color que es mostra darrere del contingut i el farciment.
- **Marge, margin:** separació opcional existent entre la caixa i la resta de caixes adjacents.

---

El farciment i el marge que defineix el model de caixa són transparents, per tant, es visualitza la imatge o el color de fons que s'hi hagi definit.

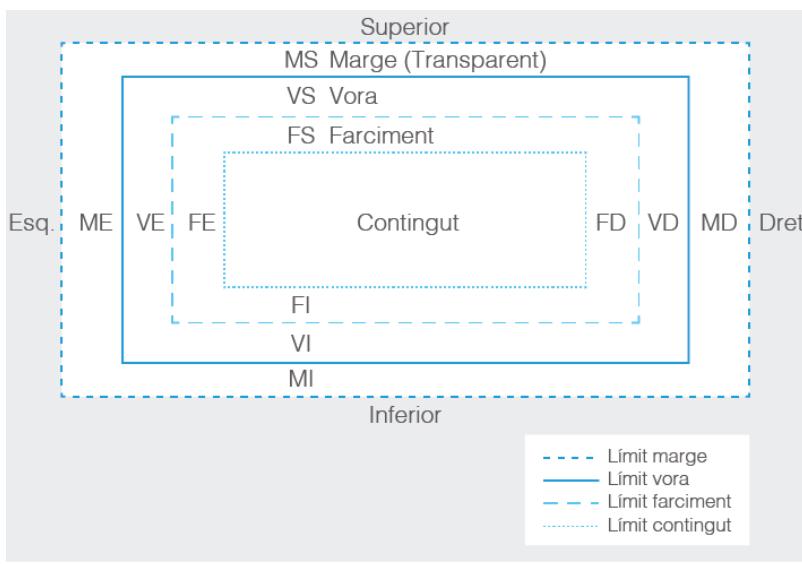
---

### Imatge i color de fons del model de caixa

Si en una caixa es defineixen el color i la imatge de fons, la imatge té més prioritat i es visualitza. Però si la imatge no cobreix totalment la caixa de l'element també es visualitza el color de fons.

Així doncs, cada caixa té una àrea de contingut, i opcionalment l'envolten unes àrees de farciment, vora i marge. La figura 2.12 mostra com es relacionen aquestes àrees.

**FIGURA 2.12.** El model de caixa



## Propietats per a la caixa

Podem consultar en la taula 2.14 un recull de les propietats per a la caixa.

**TAULA 2.14.** Propietats per a la caixa

Propietat	Propietats
<b>Marge de la caixa</b>	margin-left, margin-right, margin-top, margin-bottom, margin
<b>Fons</b>	background-color, background-image
<b>Amplada de la vora</b>	border-top-width, border-right-width, border-bottom-width, border-left-width, border-width
<b>Color de la vora</b>	border-top-color, border-right-color, border-bottom-color, border-left-color, border-color
<b>Estil de la vora</b>	border-top-style, border-right-style, border-bottom-style, border-left-style, border-style
<b>Farciment de la caixa</b>	padding-left, padding-right, padding-top, padding-bottom, padding

La propietat `box-sizing` s'utilitza per modificar l'alçada i l'amplada de la caixa dels elements HTML, ja que per defecte es calcula sense tenir en compte el farciment i l'amplada de la vora.

Així doncs, la propietat `box-sizing` pot prendre per valor `content-box`, el qual és el valor per defecte de les caixes, i no es té en compte la mida del farciment i de la vora per calcular la mida de la caixa. O bé el valor `border-box`, on sí que es té en compte la mida del farciment i de la vora per calcular la mida de la caixa.

#### Estils CSS per a la caixa

```

1 .div1 {
2   width: 300px;
3   height: 100px;
4   border: 1px solid blue;
5 }
6
7 .div2 {
8   width: 300px;
9   height: 100px;
10  padding: 50px;
11  border: 1px solid red;
12 }
13
14 .div3 {
15   width: 300px;
16   height: 100px;
17   border: 1px solid blue;
18   box-sizing: border-box;
19 }
20
21 .div4 {
22   width: 300px;
23   height: 100px;
24   padding: 50px;
25   border: 1px solid red;
26   box-sizing: border-box;
27 }
```

#### Utilització dels estils anteriors

```

1 <h2>Caixes sense box-sizing</h2>
2 <div class="div1">
```

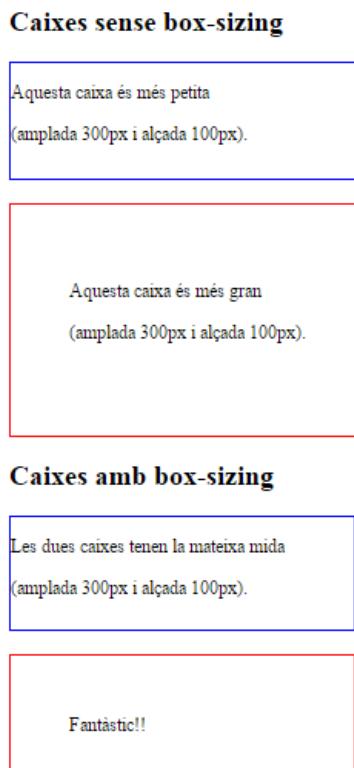
```

3 <p>Aquesta caixa és més petita </p>
4 <p>(amplada 300px i alçada 100px).</p>
5 </div>
6 <br>
7 <div class="div2">
8   <p>Aquesta caixa és més gran</p>
9   <p>(amplada 300px i alçada 100px).</p>
10 </div>
11
12 <h2>Caixes amb box-sizing</h2>
13 <div class="div3">
14   <p>Les dues caixes tenen la mateixa mida</p>
15   <p>(amplada 300px i alçada 100px).</p>
16 </div>
17 <br>
18 <div class="div4">Fantàstic!!</div>

```

El resultat que s'obté es pot veure en la figura 2.13.

**FIGURA 2.13.** Model de caixa



L'exemple anterior el podeu provar en següent enllaç: [codepen.io/ioc-daw-m09/pen/BKmmrx](https://codepen.io/ioc-daw-m09/pen/BKmmrx).

## Posicionament

Els navegadors creen i posicioneuen de manera automàtica totes les caixes dels elements que formen cada pàgina HTML seguint el flux normal d'aquesta. No obstant això, CSS permet modificar la posició on es mostra la caixa, per així poder aconseguir un disseny determinat.

Per indicar els diferents tipus de posicionament de la caixa s'utilitza la propietat CSS `position`, i els valors que pot prendre són `static`, `relative`, `absolute` o `fixed`.

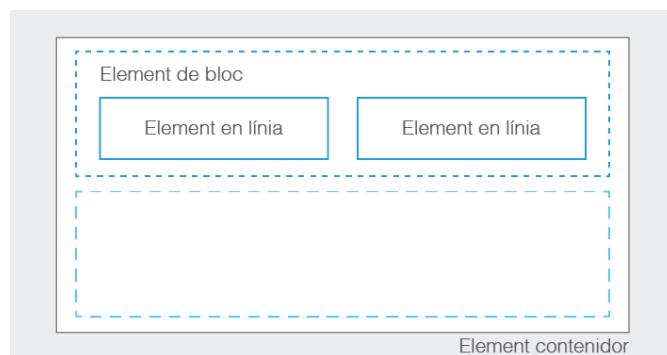
Els diferents tipus de posicionament de les caixes són:

### 1. Posicionament estàtic o normal

El posicionament de les caixes és estàtic si no s'indica el contrari. En aquest model només es té en compte si l'element és en bloc o en línia, les propietats `width` i `height` i el seu contingut.

En la figura 2.14 es pot veure com quedarien les caixes amb aquest model.

**FIGURA 2.14.** Posicionament normal



Visualització sobre les caixes dels elements HTML que s'han posicionat de manera normal.

Així doncs, veieu que per defecte els elements en bloc es mostren un a sota de l'altre, i que els elements en línia es mostren un darrere de l'altre. I si un element es troba dins d'un altre, l'element pare s'anomena *element contenidor* i determina tant la posició com la mida de les caixes del seu interior.

### 2. Posicionament relatiu

Model que consisteix a posicionar una caixa amb el seu posicionament normal i després desplaçar-la.

El desplaçament de la caixa es controla amb les propietats `top`, `bottom`, `right` i `left`. Així doncs, el valor d'aquestes propietats s'utilitza per moure les caixes de manera descendent, ascendent, cap a l'esquerra i cap a la dreta, respectivament. Si s'utilitzen valors negatius el seu efecte serà l'invers. Fixeu-vos, doncs, que aquest comportament és poc intuïtiu i pot causar errors si no s'hi està acostumat.

#### Estils amb diferents tipus de posicionament

```

1  div {
2      position: relative;
3      width: 250px;
4      height: 180px;
5      border: 1px solid black;
6      margin: 2px;
7      box-sizing: border-box;
8      padding: 2px;
9      background-color: rgba(255, 0, 0, 0.2);
10 }
11

```

```

12 .caixaInline {
13   height: 20px;
14   display: inline;
15 }
16
17 .caixaBlock {
18   width: 142px;
19   height: 20px;
20   margin-top: 8px;
21 }
22
23 .caixaRel {
24   position: relative;
25   top: 10px;
26   left: 5px;
27   width: 50px;
28   height: 20px;
29   display: inline;
30 }
31
32 .caixaAbs {
33   position: absolute;
34   top: 70px;
35   left: 100px;
36   width: 50px;
37   height: 20px;
38   display: inline;
39 }
40
41 .caixaFixa {
42   position: fixed;
43   top: 180px;
44   left: 100px;
45   width: 50px;
46   height: 20px;
47   display: inline;
48 }
49
50 .caixaFlotantDreta{
51   float: right;
52   width: 50px;
53   height: 20px;
54 }
55
56 .caixaFlotantEsquerra{
57   float: left;
58   width: 50px;
59   height: 20px;
60 }
61
62 body {
63   font-family: arial;
64   font-size: 0.8rem;
65 }
```

**Utilització dels estils de posicionament anteriors per comprovar la posició relativa respecte a la posició normal.**

```

1 <div>
2   <p>Caixa contenidora: les caixes estan en posició estàtica.</p>
3   <div class="caixaInline">caixa1</div>
4   <div class="caixaInline">caixa2</div>
5   <div class="caixaInline">caixa3</div>
6   <div class="caixaBlock">caixa4</div>
7 </div>
8 <div>
9   <p>Caixa contenidora: les caixes estan en posició estàtica,
10    excepte la caixa2, que està en posició relativa.</p>
11   <div class="caixaInline">caixa1</div>
```

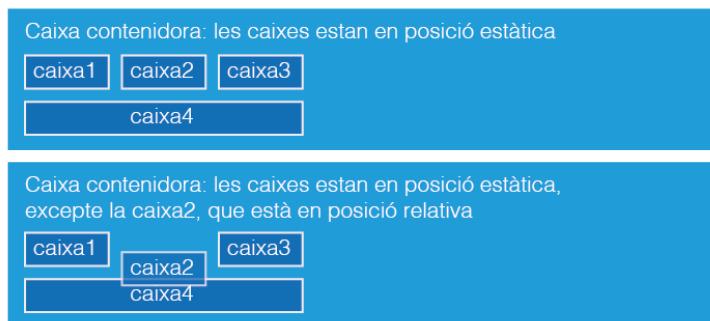
```

11   <div class="caixaRel">caixa2</div>
12   <div class="caixaInline">caixa3</div>
13   <div class="caixaBlock">caixa4</div>
14 </div>

```

El desplaçament relatiu d'una caixa no afecta la resta de caixes adjacents, les quals es mostren a la mateixa posició (vegeu la figura 2.15).

**FIGURA 2.15.** Posició estàtica



Podeu provar aquest exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/xVPpXX](http://codepen.io/ioc-daw-m09/pen/xVPpXX).

### 3. Posicionament absolut

En aquest model, la posició de la caixa s'estableix de manera absoluta respecte del seu element contenidor. L'element contenidor de referència serà el que estigui posicionat de qualsevol manera diferent de `position:static`.

Si no hi ha cap element que pugui fer de referència, llavors ho serà la finestra del navegador, que no s'ha de confondre amb l'element `<body>`.

La nova posició de la caixa s'indica mitjançant les propietats `top`, `right`, `bottom` i `left`, de tal manera que:

- El valor de la propietat `top` indica el desplaçament des de la vora superior de la caixa fins a la vora superior de l'element contenidor de referència.
- El valor de la propietat `right` indica el desplaçament des de la vora dreta de la caixa fins a la vora dreta de l'element contenidor de referència.
- El valor de la propietat `bottom` indica el desplaçament des de la vora inferior de la caixa fins a la vora inferior de l'element contenidor de referència.
- El valor de la propietat `left` indica el desplaçament des de la vora esquerra de la caixa fins a la vora esquerra de l'element contenidor de referència.

Les caixes posicionades de manera absoluta **surten del flux normal de la pàgina**, fet que provoca que la resta d'elements es moguin.

**Utilització dels estils de posicionament anteriors per comprovar la posició absoluta respecte de la posició normal**

```

1 <div>
2   <p>Caixa contenidora: les caixes estan en posició estàtica.</p>
3   <div class="caixaInline">caixa1</div>

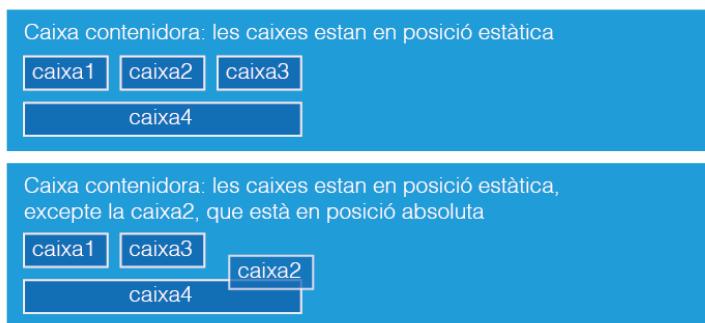
```

```

4 <div class="caixaInline">caixa2</div>
5 <div class="caixaInline">caixa3</div>
6 <div class="caixaBlock">caixa4</div>
7 </div>
8 <div>
9   <p>Caixa contenidora: les caixes estan en posició estàtica,
    excepte la caixa2 que està en posició absoluta.</p>
10  <div class="caixaInline">caixa1</div>
11  <div class="caixaAbs">caixa2</div>
12  <div class="caixaInline">caixa3</div>
13  <div class="caixaBlock">caixa4</div>
14 </div>
```

La caixa 2 està posicionada de manera absoluta, fet que provoca que la resta d'elements de la pàgina modifiquin la seva posició i ocupin el seu lloc (vegeu la figura 2.16).

**FIGURA 2.16.** Posició absoluta



Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/aNVEVY](https://codepen.io/ioc-daw-m09/pen/aNVEVY).

#### 4. Posicionament fix

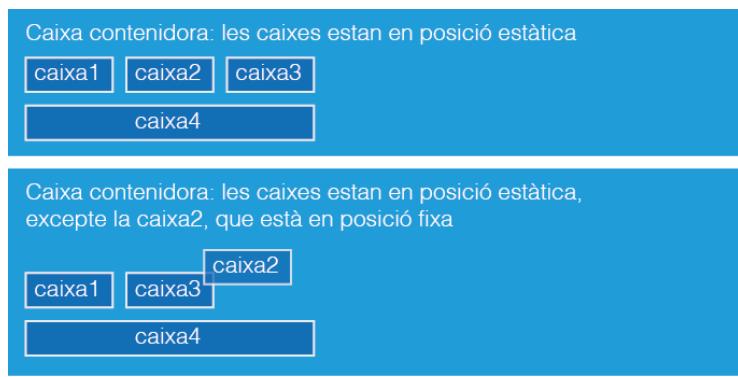
En aquest model, la caixa es converteix en un element inamovible, de manera que la seva posició a la pantalla sempre és la mateixa independentment dels altres elements i de si l'usuari puja o baixa la finestra del navegador. És un cas particular de posicionament absolut.

**Utilització dels estils de posicionament anteriors per comprovar la posició fixa respecte a la posició normal**

```

1 <div>
2   <p>Caixa contenidora: les caixes estan en posició estàtica.</p>
3   <div class="caixaInline">caixa1</div>
4   <div class="caixaInline">caixa2</div>
5   <div class="caixaInline">caixa3</div>
6   <div class="caixaBlock">caixa4</div>
7 </div>
8 <div>
9   <p>Caixa contenidora: les caixes estan en posició estàtica,
    excepte la caixa2 que està en posició fixa.</p>
10  <div class="caixaInline">caixa1</div>
11  <div class="caixaFixa">caixa2</div>
12  <div class="caixaInline">caixa3</div>
13  <div class="caixaBlock">caixa4</div>
14 </div>
```

El resultat que obtindreu es pot veure en la figura 2.17.

**FIGURA 2.17.** Posició fixa

Podeu provar l'anterior exemple en l'enllaç següent: [codepen.io/ioc-daw-m09/pen/PNOERg](https://codepen.io/ioc-daw-m09/pen/PNOERg).

## 5. Posicionament flotant

En aquest model es desplaça la caixa tot el possible a la dreta o a l'esquerra de la línia en la qual es troba, i llavors aquesta caixa deixa de formar part del flux normal de la pàgina, de manera que la resta de caixes ocupen el seu lloc.

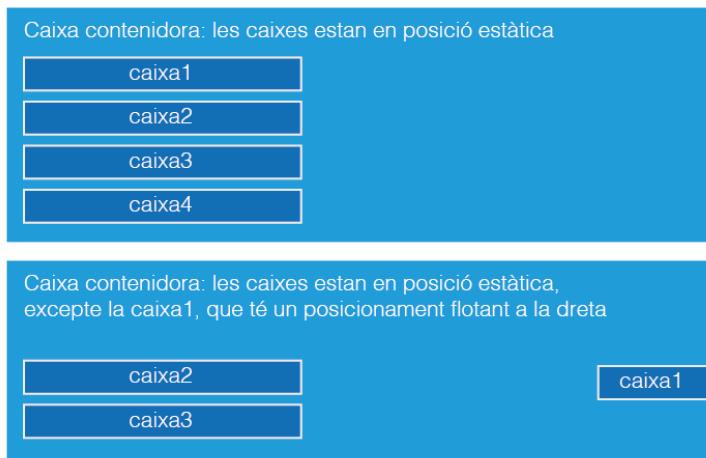
La propietat CSS que permet posicionar una caixa de forma flotant s'anomena `float`, i pot prendre per valors `left`, `right`, `none` i `inherit`.

**Utilització dels estils de posicionament anteriors per comprovar la posició flotant respecte a la posició normal**

```

1 <div>
2   <p>Caixa contenidora. Les es caixes estan en posició estàtica.<
3     /p>
4   <div class="caixaBlock">caixa1</div>
5   <div class="caixaBlock">caixa2</div>
6   <div class="caixaBlock">caixa3</div>
7   <div class="caixaBlock">caixa4</div>
8 </div>
9 <div>
10  <p>Caixa contenidora. Les caixes estan en posició estàtica,
11    excepte la caixa1 que té un posicionament flotant a la
12    dreta.</p>
13  <div class="caixaFlotantDreta">caixa1</div>
14  <div class="caixaBlock">caixa2</div>
15  <div class="caixaBlock">caixa3</div>
16 </div>
```

El resultat que obtindreu es pot veure en la figura 2.18.

**FIGURA 2.18.** Posició flotant

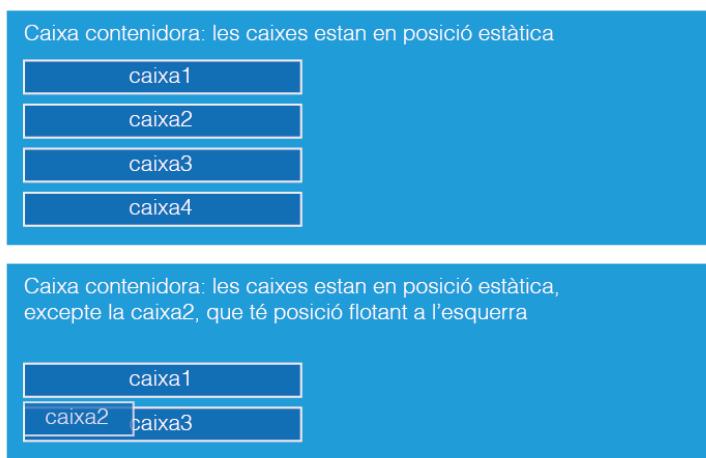
Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-daw-m09/pen/oxopQM](https://codepen.io/ioc-daw-m09/pen/oxopQM).

**Utilització dels estils de posicionament anteriors per comprovar la posició flotant respecte a la posició normal**

```

1 <div>
2   <p>Caixa contenidora. Les caixes estan en posició estàtica.</p>
3   <div class="caixaBlock">caixa1</div>
4   <div class="caixaBlock">caixa2</div>
5   <div class="caixaBlock">caixa3</div>
6   <div class="caixaBlock">caixa4</div>
7 </div>
8 <div>
9   <p>Caixa contenidora. Les caixes estan en posició estàtica,
10    excepte la caixa2 que té posició flotant a l'esquerra.</p>
11   <div class="caixaBlock">caixa1</div>
12   <div class="caixaFlotantEsquerra">caixa2</div>
13   <div class="caixaBlock">caixa3</div>
14 </div>
```

El resultat que obtindreu es pot veure en la figura 2.19.

**FIGURA 2.19.** Posició flotant respecte de la normal

Podeu provar l'exemple anterior en el següent enllaç: [codepen.io/ioc-daw-m09/pen/QNOaoN](https://codepen.io/ioc-daw-m09/pen/QNOaoN).

### Utilització dels estils de posicionament anteriors per comprovar la posició flotant respecte a la posició normal

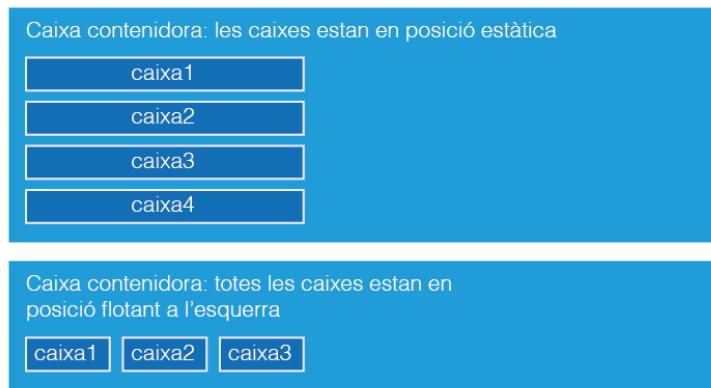
```

1  <div>
2      <p>Caixa contenidora. Les caixes estan en posició estàtica.</p>
3      <div class="caixaBlock">caixa1</div>
4      <div class="caixaBlock">caixa2</div>
5      <div class="caixaBlock">caixa3</div>
6      <div class="caixaBlock">caixa4</div>
7  </div>
8  <div>
9      <p>Caixa contenidora. Totes les caixes estan en posició flotant
10     a l'esquerra.</p>
11     <div class="caixaFlotantEsquerda">caixa1</div>
12     <div class="caixaFlotantEsquerda">caixa2</div>
13     <div class="caixaFlotantEsquerda">caixa3</div>
14 </div>

```

El resultat que obtindreu es pot veure en la figura 2.20.

**FIGURA 2.20.** Posició flotant respecte a l'esquerra



Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-dawm09/pen/KzyZLN](https://codepen.io/ioc-dawm09/pen/KzyZLN).

Es poden utilitzar tots els tipus de posicionament, ja que en funció del que vulguem aconseguir ens interessarà posicionar la caixa d'un element utilitzant un tipus o un altre. Tot i això, el més utilitzat és el flotant.

A més a més, la propietat CSS `clear` permet especificar si un element pot estar al costat dels elements flotants que el precedeixen o s'ha de moure cap avall per sota d'ells. Aquesta propietat es pot aplicar tant als elements flotants com els no flotants.

### Exemple d'utilització de la propietat clear

```

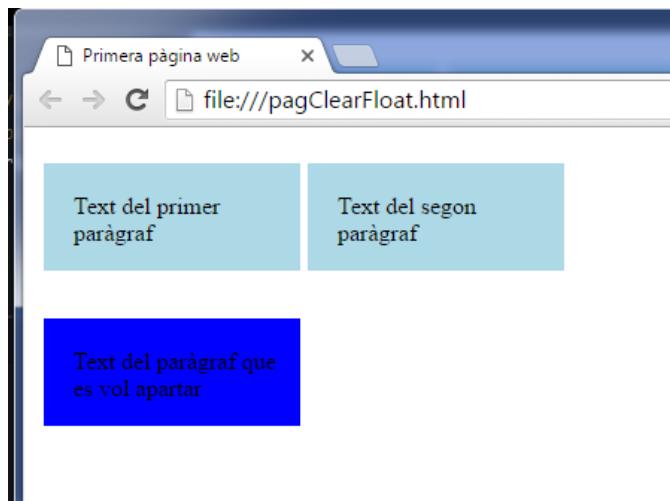
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <title>Primera pàgina web</title>
6      <style>
7          p {
8              float: left;
9              max-width: 150px;
10             min-height: 50px;
11             background-color: lightblue;

```

```

12 margin-left: 5px;
13 padding: 20px 2px 2px 20px;
14
15 }
16
17 p.clear {
18   clear: both;
19   background-color: blue;
20 }
21 </style>
22 </head>
23 <body>
24 <p>Text del primer paràgraf</p>
25 <p>Text del segon paràgraf</p>
26 <p class="clear">Text del paràgraf que es vol apartar</p>
27
28
29 </body>
30 </html>
```

**FIGURA 2.21.** Visualització d'elements amb posicionament flotant



Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/loc-daw-m09/pen/RRxjy](https://codepen.io/loc-daw-m09/pen/RRxjy).

## Visualització al navegador

Quan el navegador visualitza una pàgina web crea una caixa per a cada element HTML, i aquesta es pot modificar amb CSS. Per tant, els factors que té en compte a l'hora de generar cada caixa són:

- La mida de la caixa, si és que s'ha establert a través de les propietats `width` i `height`.
- El tipus d'element HTML, és a dir, `block` o `inline`.
- El posicionament de la caixa.
- Les relacions entre elements (on es troba cada element, elements descendents, etc).
- Altres tipus d'informació, com per exemple la mida de les imatges i de la finestra del navegador.

A més a més, CSS defineix propietats per controlar la visualització de les caixes. Aquestes propietats són `display`, `visibility`, `overflow` i `z-index`.

En la taula 2.15 apareixen els valors que poden prendre aquestes propietats.

**TAULA 2.15.** Valors de les propietats

Propietat	Valors
<code>display</code>	<code>inline</code> , <code>block</code> , <code>none</code> , <code>list-item</code> , <code>run-in</code> , <code>inline-block</code> , <code>table</code> , <code>inline-table</code> , <code>table-row-group</code> , <code>table-header-group</code> , <code>table-footer-group</code> , <code>table-row</code> , <code>table-column-group</code> , <code>table-column</code> , <code>table-cell</code> , <code>table-caption</code> , <code>inherit</code>
<code>visibility</code>	<code>visible</code> , <code>hidden</code> , <code>collapse</code> , <code>inherit</code>
<code>overflow</code>	<code>visible</code> , <code>hidden</code> , <code>scroll</code> , <code>auto</code> , <code>inherit</code>
<code>z-index</code>	<code>auto</code> , <code>numero</code> , <code>inherit</code>

Les propietats `display` i `visibility` permeten amagar qualsevol element de la pàgina. La propietat `display` permet amagar un element perquè desaparegui de la pàgina i així els altres elements puguin ocupar el seu lloc. En canvi, `visibility` permet fer invisible un element; el navegador crearia la caixa, però no la mostraria.

De fet, la propietat `display` ofereix moltes més possibilitats que amagar o mostrar un element, i en realitat modifica la manera com es visualitza un element. Així doncs, els valors més utilitzats d'aquesta propietat són:

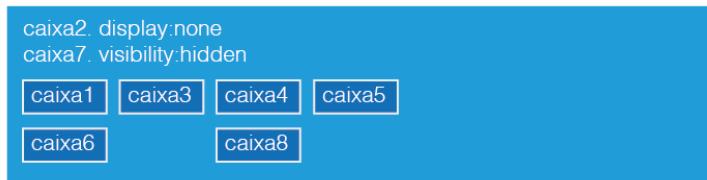
- `inline`: mostra l'element com si fos en línia, independentment de si ho és o no.
- `block`: mostra l'element com si fos en bloc, independentment de si ho és o no.
- `none`: oculta l'element i fa que desaparegui de la pàgina.

#### Visualització de les propietats "display" i "visibility"

```

1  <div>
2    <p>Caixa2. display:none</p>
3    <p>Caixa7. visibility:hidden</p>
4    <div class="caixaFlotantEsquerra"> Caixa1 </div>
5    <div class="caixaFlotantEsquerra" style="display:none"> Caixa2
6      </div>
7    <div class="caixaFlotantEsquerra"> Caixa3 </div>
8    <div class="caixaFlotantEsquerra"> Caixa4 </div>
9    <div class="caixaFlotantEsquerra"> Caixa4 </div>
10   <div class="caixaFlotantEsquerra"> Caixa6 </div>
11   <div class="caixaFlotantEsquerra" style="visibility:hidden">
12     Caixa7 </div>
13   <div class="caixaFlotantEsquerra"> Caixa8 </div>
14 </div>
```

El resultat es pot veure en la figura 2.22.

**FIGURA 2.22.** Visualització de les propietats 'display' i 'visibility'

En general, quan s'oculta un element no es vol que segueixi ocupant lloc. Així, per aconseguir aquest efecte és millor usar 'display' que 'visibility'.

Podeu provar l'exemple anterior en el següent enllaç: [codepen.io/ioc-daw-m09/pen/ONoqMg](https://codepen.io/ioc-daw-m09/pen/ONoqMg).

La propietat `overflow` s'utilitza per controlar la manera com es visualitzen els continguts que sobresurten de la caixa que té definida.

#### Visualització de la propietat "overflow"

```

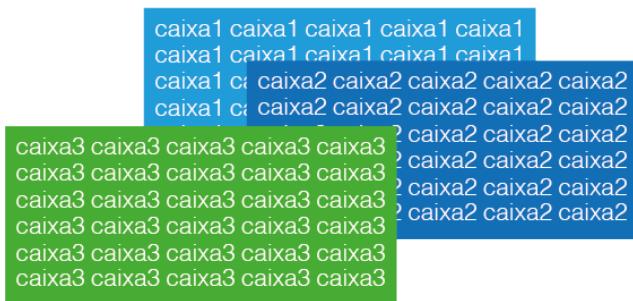
1  <div>
2    <p>Caixa1. overflow:hidden</p>
3    <p>Caixa4. overflow:visible</p>
4    <p>Caixa6. overflow:scroll</p>
5    <div class="caixaFlotantEsquerra" style="overflow:hidden">
6      Caixa1 caixa1 caixa1</div>
7    <div class="caixaFlotantEsquerra"> Caixa2 </div>
8    <div class="caixaFlotantEsquerra"> Caixa3 </div>
9    <div class="caixaFlotantEsquerra"> Caixa4 </div>
10   <div class="caixaFlotantEsquerra" style="overflow:visible">
11     Caixa4 caixa4 caixa4 </div>
12   <div class="caixaFlotantEsquerra" style="height: 50px; overflow
13     :scroll"> Caixa6 caixa6 caixa6</div>
14   <div class="caixaFlotantEsquerra"> Caixa7 </div>
15   <div class="caixaFlotantEsquerra"> Caixa8 </div>
16 </div>
```

El resultat que obtindreu es pot veure en la figura 2.23.

**FIGURA 2.23.** Visualització de la propietat 'overflow'

L'anterior exemple el podeu provar en el següent enllaç: [codepen.io/ioc-daw-m09/pen/Rajvdr](https://codepen.io/ioc-daw-m09/pen/Rajvdr).

La propietat `z-index` controla la posició tridimensional d'un element que s'estableix sobre el tercer eix Z, i així permet establir quina caixa es veu per damunt d'una altra, tal com es veu en la figura 2.24.

**FIGURA 2.24.** Posició de les caixes

## 2.1.8 Regles arrova

A més de les regles CSS que es defineixen a través del seu selector, hi ha una sèrie de regles anomenades *regles arrova* que tenen usos molt diversos.

Les regles arrova comencen sempre amb el símbol ”@” seguit d'un nom clau; es poden declarar en un full extern o incrustat. Algunes es poden consultar en la taula 2.16.

**TAULA 2.16.** Regles @

Regla	Descripció
@import	Agrega els estils CSS d'un document extern.
@media	Aplica les regles que conté al tipus de dispositiu que s'especifica.
@font-face	Especifica una font no inclosa en el navegador que l'usuari es descarregarà.
@charset	Especifica quin és el joc de caràcters que farem servir dins del fitxer CSS.
@page	Estableix les dimensions, l'orientació i els marges del quadre d'un document.
@supports	Especifica una o diverses condicions, que en cas de complir-se s'aplicaran els estils definits.

A continuació teniu alguns exemples d'utilització d'aquestes regles:

**Regla @media**

```

1  /* Definició de diferents estils per diferents tipus de medis.*/
2  @media print {
3      body{ font-size: 10pt }
4  }
5  @media screen {
6      body { font-size: 12pt }
7  }
8  @media screen, print {
9      body { line-height: 1.2 }
10 }
```

**Regla @font-face**

```

1  /* Inclusió d'una font que preveiem que el navegador no tindrà
   instal·lada, per així utilitzar-la després en la definició de
```

```

1   l'estil per als paràgrafs. */
2 @font-face {
3   font-family: DeliciousRoman;
4   src: url("Delicious-Roman.otf");
5 }
6
7 p {
8   font-family: DeliciousRoman, Helvetica, Arial, sans-serif;
9 }
```

#### Regla @charset

```

1 /* activa el joc de caràcters per al full d'estil a Unicode UTF-8
   */
2 @charset "UTF-8";
3 /* activa el joc de caràcters per al full d'estil Latin-9 (llengü
   es de l'oest d'Europa, amb el símbol de l'euro) */
4 @charset 'iso-8859-15';
```

#### Regla @page

```

1 /* definim els marges d'impressió de la pàgina. */
2 @page {margin-left: 1cm; margin-right: 0.5cm;}
```

#### Regla @supports

```

1 /*si la propietat display pren per valor flex, s'aplicaran els
   següents estils:*/
2 @supports (display:flex) {
3   section { display: flex }
4 ...
5 }
```

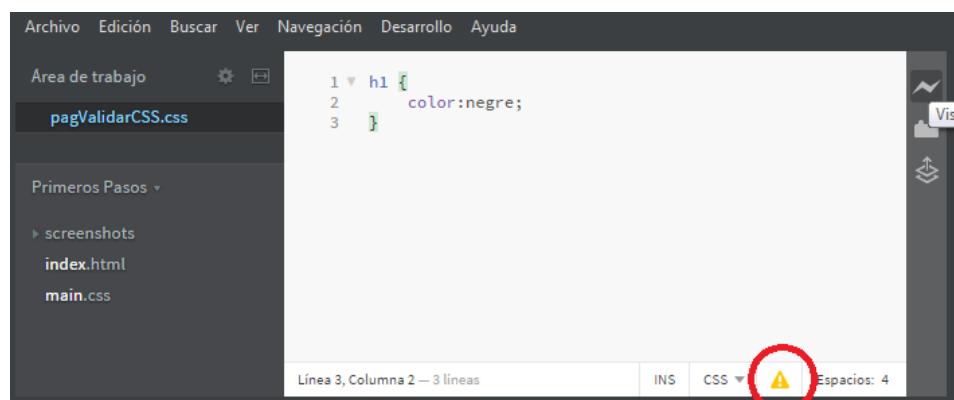
### 2.1.9 Test i validació CSS

Per verificar que el codi CSS és correcte, la gran majoria dels editors d'HTML i CSS contenen unes eines que permeten fer una sèrie de proves per validar-lo.

Si es vol fer aquest test amb l'editor Brackets, el que es farà serà utilitzar l'extensió CSSLint.

Aquesta extensió afegeix una icona a la barra d'estat, de tal manera que cada vegada que es guarda el full d'estils es valida.

Tal com es mostra en la figura 2.25, en el cas que no hi hagi cap error es mostra una icona de color verd. En cas que sí que n'hi hagi, tal com podeu veure en la figura 2.26, es mostra un triangle groc.

**FIGURA 2.25.** Visualització del validador CSSLint a l'editor Brackets**FIGURA 2.26.** Visualització del validador CSSLint a l'editor Brackets

Aquest validador no tan sols detecta els errors, sinó que també proporciona consells per millorar els fulls d'estils. Tal com es mostra en la figura 2.27, fent clic en el triangle taronja podeu veure els errors detectats.

**FIGURA 2.27.** Visualització dels errors del validador CSSLint

Alternativament, també es pot verificar que el codi en CSS és correcte mitjançant la web [jigsaw.w3.org/css-validator](http://jigsaw.w3.org/css-validator).

La validació es pot dur a terme de diverses maneres:

- Indicant l'URL complet de la pàgina que desitgem validar.
- Pujant el document CSS.
- Escrivint el codi CSS directament.

Llavors seleccionem *More Options* i allí ens assegurem que a *Perfil* està selecciónat *CSS Versión 3*. Finalment, polsem el botó *Check*, tal com es mostra en la figura 2.28.

**FIGURA 2.28.** Pantalla del validador de W3

The screenshot shows the W3C CSS Validation Service interface. At the top, there are language selection buttons for various languages like English, Spanish, French, etc. Below that is the W3C logo and the title "CSS Validation Service". A sub-header says "Check Cascading Style Sheets (CSS) and (X)HTML documents with style sheets". There are three input methods: "By URI", "By file upload", and "By direct input". The "By direct input" tab is selected, showing a text area where the following CSS code is entered:

```
h1 {
    color:negre;
}
```

Below the text area, there's a "More Options" section with dropdown menus for "Profile" (set to "CSS level 3"), "Medium" (set to "All"), "Type" (set to "CSS"), and "Warnings" (set to "Normal report"). To the right of these dropdowns is a "Vendor Extension" link. At the bottom of the "More Options" section is a "Check" button.

En la figura 2.29 veiem la pantalla d'informació dels errors CSS trobats.

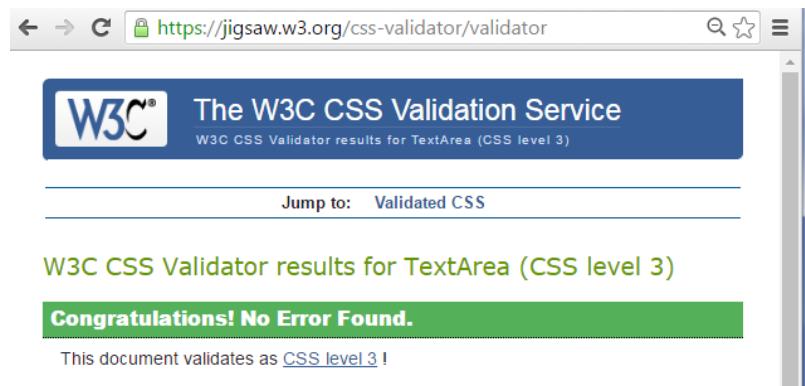
**FIGURA 2.29.** Pantalla d'informació de l'error CSS trobat

The screenshot shows the results of a CSS validation. At the top, it says "The W3C CSS Validation Service" and "W3C Validator results for TextArea (CSS level 3)". Below that is a red banner with the text "Sorry! We found the following errors (1)". A table lists the error: "URI : TextArea" and "2 h1 Value Error : color negre is not a color value : negre".

At the bottom of the error table, there's a message from HP: "The W3C validators are hosted on server technology donated by HP, and supported by community donations. [Donate](#) and help us build better tools for a better web." There are also links for "Home", "About", "Documentation", "Download", "Feedback", and "Credits".

En el cas que no hi hagués cap error CSS, es mostraria una pantalla com la de la figura 2.30.

**FIGURA 2.30.** Pantalla que indica que no hi ha hagut cap error



## 2.2 Disseny web adaptatiu

---

El contingut d'un web es podrà presentar per pantalla en una columna en un telèfon mòbil o en dues columnes en una tauleta, i en tres columnes en un ordinador.

---

El creixent ús de dispositius mòbils i tauletes per accedir a Internet en substitució de l'ordinador fa palès que molts llocs web no estan optimitzats per visualitzar-se correctament en aquests dispositius. Així, molts dels dispositius solen estar limitats per la mida de la pantalla i requereixen un enfocament diferent sobre com presentar el contingut.

En conseqüència, tal com es pot apreciar en la figura 2.31, hi ha diferents mides de pantalla en telèfons mòbils, tauletes, ordinadors de sobretaula, consoles de jocs, televisors, etc. Les mides de pantalla seran sempre canviants, per la qual cosa és important que el lloc web pugui adaptar-se a qualsevol mida de pantalla en l'actualitat o en el futur.

**FIGURA 2.31.** Mida de les pantalles



Es defineix el **disseny adaptatiu** o, en anglès, Responsive Web Design, com una tècnica de disseny i desenvolupament web que permet que un lloc web es vegi correctament en tots els tipus de dispositius: ordinadors de sobretaula, portàtils, tauletes, telèfons mòbils, televisors, etc.

El disseny adaptatiu s'aconsegueix amb l'ús dels fulls d'estil CSS aplicats als elements HTML de les pàgines web. Així, es pot canviar la mida, amagar, encongrir, engrandir o moure el contingut de la pàgina per fer-lo visible i adequat a qualsevol pantalla.

### 2.2.1 Tècniques per crear un disseny adaptatiu

Per tal que el lloc web sigui adaptatiu és important que disposi de diverses versions perquè es pugui visualitzar correctament des de qualsevol dispositiu amb diferents mides de pantalla i resolucions. D'aquesta manera, almenys caldria disposar de:

- Una versió per a telèfon mòbil.
- Una versió per mòbil en mode apaïsat.
- Una versió per a la tauleta.
- Una versió per a ordinador d'escriptori.

No s'ha d'oblidar que l'experiència d'ús és molt diferent amb un dispositiu o amb un altre, ja que la mida de la pantalla d'un mòbil acostuma a ser més petita, a pesar que la seva resolució pot ser més gran que la d'un monitor de 17”.

La resolució de pantalla de l'iPhone5S és de 1.136 x 640 px, mentre que la del Nexus5 és 1.920 x 1.080 px.

Així que no tan sols és necessari adaptar el contingut a la pantalla, sinó que també cal tenir en compte les necessitats de l'usuari i les capacitats del dispositiu. Per això, també caldrà considerar el que vol fer l'usuari en primer lloc i mostrar-li la informació en funció d'aquest criteri. Tampoc s'ha de suposar que l'usuari amb un telèfon mòbil no voldrà accedir a tota la informació del lloc, i per tant caldrà idear la manera d'accendir a la informació, com es mostra i en quin ordre.

Les tècniques que s'utilitzaran per crear el disseny adaptatiu són:

- **Media queries** que permeten saber quin tipus de dispositiu i quines són les dimensions de la pantalla de l'usuari.
- **Un grid flexible**, és a dir, un disseny de pàgina basat en una quadrícula que pugui adaptar-se a la resolució de la pantalla.
- **Continguts i imatges flexibles** als quals, utilitzant estils CSS, s'apliquen canvis d'escala dinàmics per tal que s'adaptin a la mida de la pantalla.

### Disseny web fluid 'vs.' adaptatiu

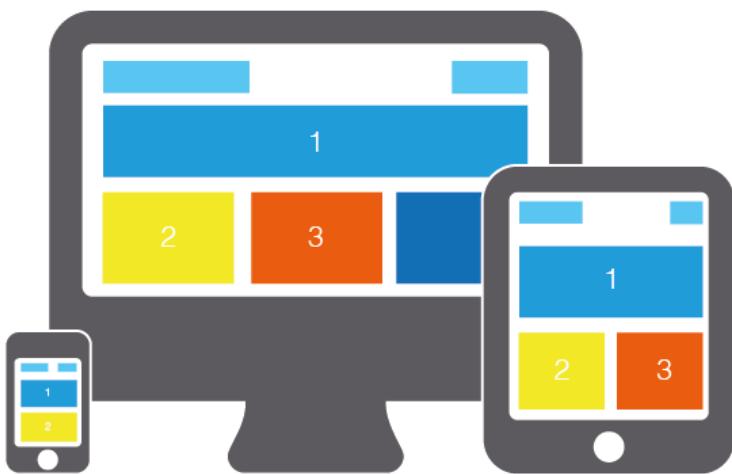
Una web té un **disseny líquid** o **fluid** quan el contingut s'ajusta horizontalment a la mida de la pantalla sense necessitat de la barra de desplaçament horitzontal (*scroll*).

El problema sorgeix tant en pantalles molt grans com en pantalles molt petites, ja que el disseny es desajusta, bé perquè apareixen grans espais en blanc o bé perquè les imatges es miniaturitzen i els textos es tornen il·legibles.

Per obviar aquests problemes, els dissenyadors poden crear diverses versions en CSS segons la pantalla on s'hagi de visualitzar, suportant així una càrrega de treball addicional sobre el manteniment de les diferents versions.

En un **disseny adaptatiu** o **responsive**, el disseny es basa en una quadrícula flexible i els blocs s'ordenen i es jerarquitzen de tal manera que els elements es mostren en una o diverses columnes, en funció de la mida de la pantalla (figura 2.32).

**FIGURA 2.32.** Visualització dels elements d'una pàgina web en diferents tipus de dispositiu



## 2.2.2 'Media queries'

Una **media query** és una consulta que dóna informació del tipus de medi en què es visualitza la pàgina web, per així poder limitar els estils que s'hi aplicaran. El resultat d'aquesta consulta serà un booleà, així que quan sigui verdader s'aplicaran les regles d'estil corresponents, seguint les normes de cascada.

Vegem amb un exemple la sintaxi d'una *media query*:

Especifiquem la **consulta** al <head> de la pàgina HTML a través de l'element <link>.

```
1  <link rel="stylesheet" media="screen and (width:780px)" href="estils780.css" />
```

I la mateixa consulta es podria especificar en un document CSS a través d'una regla @media:

```
1  @media screen and (width:780px) {  
2      ...
```

3 }

Els diferents tipus de medis que podem especificar són:

- *screen* (valor per defecte): per a presentació en pantalles d'ordinadors no paginats.
- *print*: per a la sortida per una impressora.
- *projection*: per a presentació en projectors.
- *handheld*: per a dispositius de mà (típicament petita pantalla, d'amplada limitada).
- *aural*: per a sintetitzadors de veu.
- *braille*: per a presentació en dispositius Braille.
- *tty*: per a pantalla en cel·les de caràcters.
- *tv*: per a presentació en televisors.
- *all*: per a tots els dispositius de sortida.

Vegeu en la taula 2.17 els criteris que podem usar en les consultes.

**TAULA 2.17.** Criteris per a les 'media queries'

Criteri	Valor	Mín./Màx.	Descripció
<code>width</code>	Llargada	Sí	Per examinar l'amplada de la zona de visualització del navegador.
<code>height</code>	Llargada	Sí	Per examinar l'altura de la zona de visualització del navegador
<code>device-width</code>	Llargada	Sí	Per examinar l'amplada física de la pantalla de difusió.
<code>device-height</code>	Llargada	Sí	Per examinar l'altura física de la pantalla de difusió.
<code>orientation</code>	<i>Landscape o portrait</i>	No	Per examinar si l'usuari utilitza tauleta tàctil verticalment, <i>portrait</i> o horitzontalment, <i>landscape</i> .
<code>aspect-ratio</code>	Ràtio	Sí	Per examinar el coeficient ample/alt.
<code>device-aspect-ratio</code>	Ràtio	Sí	Per examinar el coeficient físic ample/alt de la pantalla.
<code>color</code>		Sí	Per examinar si el suport de difusió utilitza el color (valor per defecte en cas que no s'hagi especificat), o el blanc i negre o una escala de grisos.
<code>color-index</code>	Número	Sí	Per examinar el número de colors de la taula de colors.
<code>monochrome</code>	Número	Sí	Per examinar el número de nivells de gris per als dispositius monocroms.
<code>resolution</code>	DPI	Sí	Per examinar la resolució de la pantalla de visualització expressada en DPI.
<code>scan</code>	<i>Progressive o interlace</i>	No	Per examinar el tipus d'exploració de les pantalles de televisió.

**TAULA 2.17** (continuació)

Criteri	Valor	Mín./Màx.	Descripció
grid	Número	No	Per examinar si la pantalla de difusió utilitza una quadrícula amb una única mida de font.

Es poden redactar consultes utilitzant operadors lògics con són ***not***, ***and*** i ***only***. A més a més, les llistes separades per comes (”,”) es comporten com l’operador ***or***.

Vegem alguns exemples per conèixer el funcionament de diversos criteris de cerca:

#### Exemples de ‘media queries’

Per examinar si la finestra té una amplada de 700 px o més i la pantalla està en horitzontal:

```
1 @media (min-width: 700px) and (orientation: landscape) { ... }
```

Per examinar si la finestra té una amplada de 700 px o més, la pantalla està en horitzontal i el dispositiu és un televisor:

```
1 @media tv and (min-width: 700px) and (orientation: landscape) {
    ...
}
```

Per examinar si l’amplada mínima és 700 px o si el dispositiu està en horitzontal:

```
1 @media (min-width: 700px), handheld and (orientation: landscape)
    { ... }
```

Per examinar pantalles que no tinguin una resolució de 780 px:

```
1
2 @media screen and (not width:780px) {
3     ...
4 }
```

Podeu consultar la següent web per obtenir més informació sobre les *media queries*: [developer.mozilla.org/en-US/docs/Web/CSS/Media\\_Queries /Using\\_media\\_queries](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries /Using_media_queries).

#### El ‘viewport’

Pràcticament tots els navegadors de dispositius mòbils, en entrar en un lloc web, analitzen la seva mida i l’escalen perquè es mostri completament a la pantalla; en conseqüència, segons com estigui optimitzat el web, és possible que el resultat sigui molt petit.

El ***viewport*** és l’àrea visible del navegador. En els navegadors per a mòbils no es correspon a la mida real de la pantalla en píxels, sinó a l’espai que la pantalla està emulant que té.

Per exemple, un iPhone que tingui la pantalla en vertical té unes dimensions de 320 px, però en realitat el dispositiu està emulant 980 px.

Per solucionar aquesta situació es pot alterar el *viewport* que està configurat en el navegador a través de l’element HTML `<meta>`, que ubicarem en el `<head>` de la pàgina web.

Les característiques o els paràmetres que podem modificar del *viewport* s'especifiquen a l'atribut *content* separades per coma. Vegeu en la taula 2.18 les diferents característiques que podem configurar.

**TAULA 2.18.** Paràmetres del 'viewport'

Paràmetre	Descripció
<code>width</code>	Amplada del <i>viewport</i> .
<code>height</code>	Altura del <i>viewport</i> .
<code>initial-scale</code>	Escala inicial del document.
<code>minimum-scale</code>	Escala mínima configurable del document.
<code>maximum-scale</code>	Escala màxima configurable del document.
<code>user-scalable</code>	Si es permet o no a l'usuari fer zoom.

D'aquesta manera, podem fixar l'amplada i l'alçada del *viewport* amb píxels (320 px, 480 px, etc.), o bé també podem usar dues constants: *device-width* i *device-height*, referents a l'amplada i a l'alçada del dispositiu mòbil.

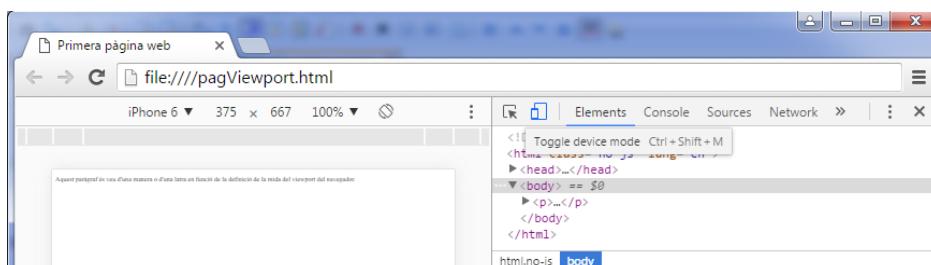
#### Exemple de definició del 'viewport'

```

1 <head>
2 ...
3   <meta name="viewport" content="width=device-width, initial-
      scale=1.0" />
4 ...
5 </head>
```

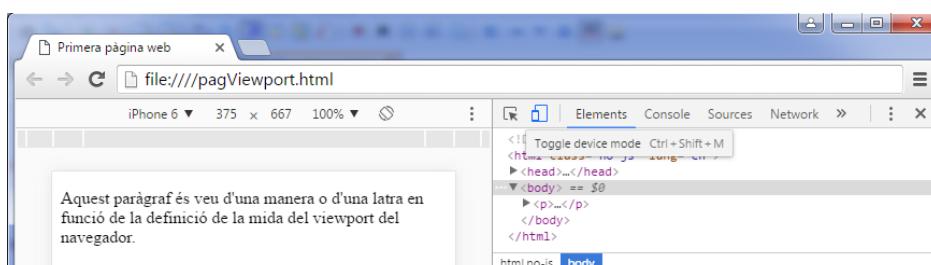
A la figura 2.33 i a la figura 2.34 hi veiem com es visualitza el contingut de la pàgina web en un telèfon mòbil si no s'hi ha definit el *viewport*, i com es veu si el *viewport* està definit.

**FIGURA 2.33.** Visualització en un iPhone 6 d'una pàgina web sense el 'viewport' definit



Visualització mitjançant l'opció 'toggle device mode' de les eines del desenvolupador que inclou el navegador Chrome

**FIGURA 2.34.** Visualització en un iPhone 6 d'una pàgina web amb el 'viewport' definit



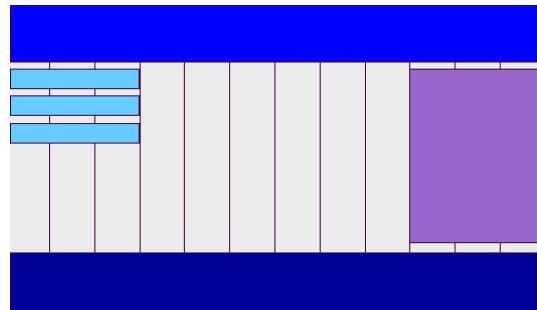
Visualització mitjançant l'opció //toggle device mode// de les eines del desenvolupador que inclou el navegador Chrome

Així, veieu que en la primera imatge el *viewport* no està definit i el paràgraf es veu molt petit, i que en la segona sí està definit i per tant el paràgraf es veu correctament.

### **2.2.3 Desenvolupar un 'grid' fluid**

Hi ha pàgines web que estan estructurades en el que s'anomena *grid*, que significa que la pàgina està dividida en columnes. D'aquesta manera és molt senzill col·locar-hi els elements, tal com s'aprecia en la figura 2.35.

**FIGURA 2.35.** Col·locació dels elements en el 'grid'



Visualització sobre la ubicació dels elements d'una pàgina web

Aquest disseny s'aconsegueix definint per a cada fila, *row*, la distribució del nombre de columnes, *col*, que ens interessa. Vegem en l'exemple següent el codi CSS per a la creació d'un *grid* fluid de 12 columnes i l'aplicació d'aquest *grid* en HTML.

### Estils CSS que defineixen un 'grid' fluid

```
1 * {
2     box-sizing: border-box;
3 }
4 @media only screen and (max-width: 768px) {
5     /* Per a telèfons mòbils: */
6     [class*="col_"] {
7         width: 100%;
8     }
9 }
10 .row:after {
11     content: "";
12     clear: both;
13     display: block;
14 }
15 [class*="col_"] {
16     float: left;
17     padding: 15px;
18 }
19 @media only screen and (min-width: 768px) {
20     .col_1 {width: 8.33%;}
21     .col_2 {width: 16.66%;}
22     .col_3 {width: 25%;}
23     .col_4 {width: 33.33%;}
24     .col_5 {width: 41.66%;}
25     .col_6 {width: 50%;}
26     .col_7 {width: 58.33%;}
27     .col_8 {width: 66.66%;}
28     .col_9 {width: 75%;}
29     .col_10 {width: 83.33%;}
30     .col_11 {width: 91.66%;}
31     .col_12 {width: 100%;}
32 }
33 html {
34     font-family: "Lucida Sans", sans-serif;
35 }
36 .header {
37     background-color: #9933cc;
38     color: #ffffff;
39     padding: 15px;
40 }
41 .footer {
42     background-color: #9933cc;
43     color: #ffffff;
44     padding: 15px;
45     text-align: center;
46 }
47 .menu ul {
48     list-style-type: none;
49     margin: 0;
50     padding: 0;
51 }
52 .menu li {
53     padding: 8px;
54     margin-bottom: 7px;
55     background-color: #33b5e5;
56     color: #ffffff;
57     box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24);
58 }
59 .menu li:hover {
60     background-color: #00099cc;
61 }
62 .aside p {
63     height: 100px;
64     padding: 8px;
65     margin: 0;
66     background-color: #33b5e5;
```

```
67     color: #ffffff;
68     box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba
69     (0,0,0,0.24);
```

### Exemple d'utilització del 'grid' definit en els estils anterior

```
1  <!doctype html>
2  <html>
3  <head>
4  ...
5  <meta name="viewport" content="width=device-width, initial-scale
6  =1.0">
7  ...
8  </head>
9  <body>
10 <header>
11   <div class="header">
12     <h1>Estudis de grau superior d'Informàtica</h1>
13   </div>
14 </header>
15 <content>
16   <div class="row">
17     <div class="col-9">
18       <div class="row">
19         <nav>
20           <div class="col-3 menu">
21             <ul>
22               <li>ASIX</li>
23               <li>DAM</li>
24               <li>DAW</li>
25             </ul>
26           </div>
27         </nav>
28         <div class="col-9">
29           <article>
30             <h2>M1. Sistemes informàtics</h2>
31             <p>Bloc 1. 118 hores (8 hores setmanals)</p>
32             <p>Bloc 2. 80 hores (5 hores setmanals)</p>
33           </article>
34         </div>
35       </div>
36       <div class="row">
37         <div class="col-3"></div>
38         <div class="col-9">
39           <article>
40             <h2>M2. Bases de dades</h2>
41             <p>Bloc 1. 132 hores (9 hores setmanals)</p>
42             <p>Bloc 2. 99 hores (7 hores setmanals). Inclou treball
43               en equip i pràctica virtual obligatòria.</p>
44           </article>
45         </div>
46       <div class="row">
47         <div class="col-3"></div>
48         <div class="col-9">
49           <article>
50             <h2>M3. Programació</h2>
51             <p>Bloc 1. 85 hores (5 hores setmanals). Inclou treball
52               en equip i pràctica virtual obligatòria.</p>
53             <p>Bloc 2. 80 hores (5 hores setmanals). Inclou treball
54               en equip i pràctica virtual obligatòria.</p>
55           </article>
56         </div>
57       </div>
58     <div class="col-3">
```

```
59 <div class="row">
60   <div class="col-12">
61     <aside>
62       <div class="aside">
63         <p> links d'interès</p>
64       </div>
65     </aside>
66   </div>
67 </div>
68 </div>

69
70 </div>
71 </content>

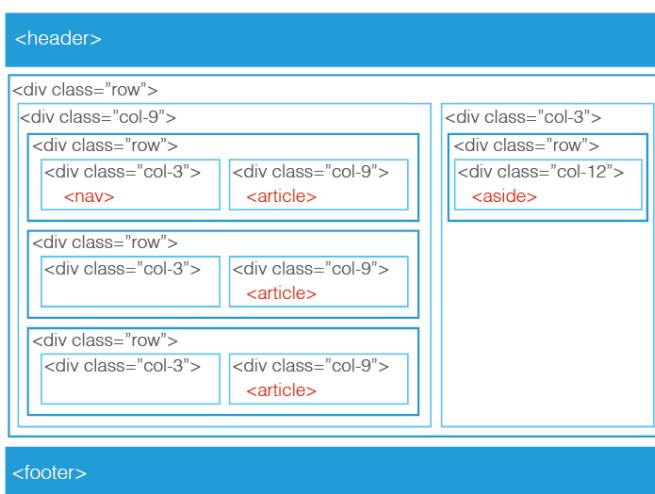
72
73 <footer>
74   <div class="footer">
75     <p>IOC – Institut Obert de Catalunya</p>
76   </div>
77 </footer>
78 </body>
79 </html>
```

El resultat apareix en la figura 2.36 i la divisió de l'espai en la figura 2.37.

**FIGURA 2.36.** Visualització de la pàgina web

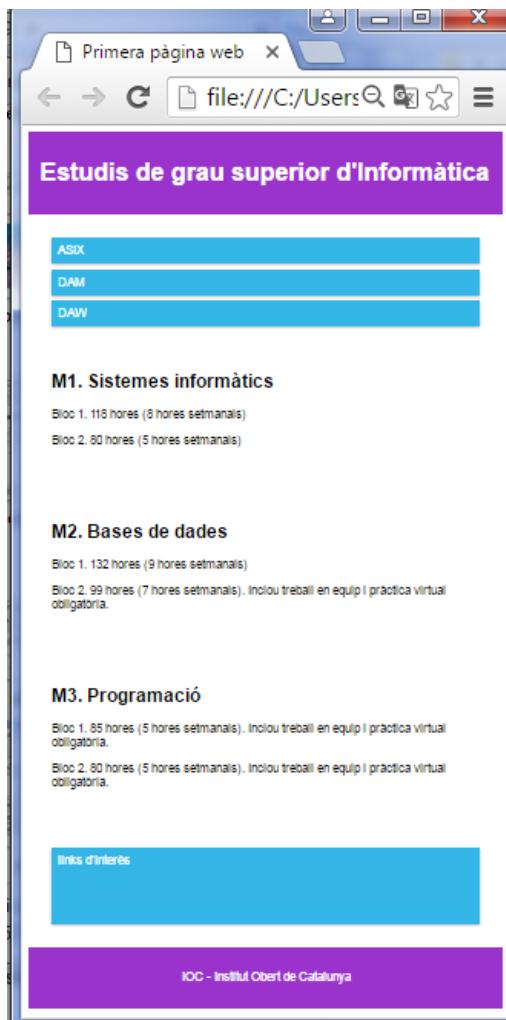


**FIGURA 2.37.** Divisió de l'espai



I en la figura 2.38 es visualitza com es veuria en una pantalla petita.

**FIGURA 2.38.** Visualització en pantalla petita



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

Fixeu-vos en l'exemple anterior, com en una fila (*row*) només hi pot haver columnes (*col*) i que el contingut sempre se situa dins de les columnes. Alhora teniu que dins d'una columna hi pot haver una altra fila, la qual també es dividirà en 12 columnes.

Amb aquest disseny de *grid* es veu que els blocs es posicionen en un lloc o un altre en funció de la mida de la pantalla. Aquest funcionament és degut al fet que la posició de les columnes és flotant, i que per a mides de pantalla inferiors a 768 px l'amplada de la columna és del 100%.

Cal remarcar la dificultat que hi pot haver en aquest tipus de *grid* si es vol canviar la disposició d'alguns dels blocs quan es canvia la mida de la pantalla, com per exemple fer que el menú de navegació (els botons *ASIX*, *DAM* i *DAW*) apareguin damunt del bloc *links d'interès* enlloc de després del *header*.

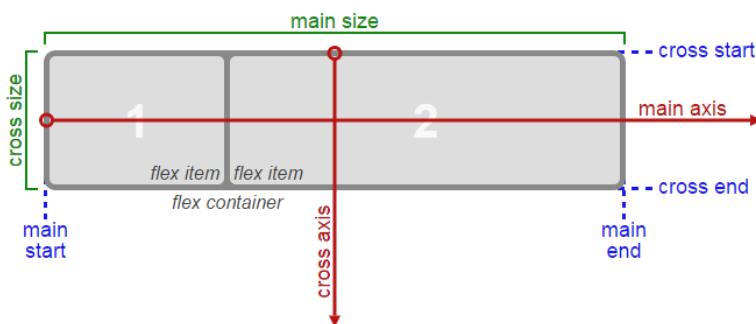
## 2.2.4 Desenvolupar un 'grid' flexible

El desenvolupament d'un *grid* flexible es basa en el nou model de caixa, ***flexbox***. Així doncs, podem crear una caixa o contingidor flexible a través de la propietat CSS `display: flex`. D'aquesta manera, la declaració d'aquesta propietat converteix de manera automàtica els seus elements fills directes en flexibles.

Un contingidor flexible té un eix principal, *main axis*, que és la direcció en la qual es posicioneuen els elements flexibles, i també té un eix transversal, perpendicular a l'eix principal. Aquests dos eixos tenen una sèrie de propietats que controlen com es posiciona cada element flexible en relació amb els altres.

Vegeu en la figura 2.39 la disposició dels dos eixos.

**FIGURA 2.39.** Model de caixa flex



Font: <https://www.w3.org/TR/css-flexbox-1/>

Vegem en la taula 2.19 les diferents propietats CSS que ens permetran crear el contingidor flexible.

**TAULA 2.19.** Propietats CSS per al contingidor flexible

Propietat	Descripció
<code>flex-direction</code>	Especifica com se situen els elements dins del contingidor. Els valors que pot prendre són <code>row</code> , <code>row-reverse</code> , <code>column</code> , <code>column-reverse</code> .
<code>flex-wrap</code>	Especifica si el contingidor té una o diverses línies. Els valors que pot prendre són: <code>no-wrap</code> , <code>wrap</code> , <code>wrap-reverse</code> .
<code>flex-flow</code>	Propietat abreujada de les propietats anteriors. <code>flex-flow: row wrap</code> .

### Exemple de caixa flexible

```

1 #caixaFlex {
2   display: flex;
3   display: -webkit-flex;
4   flex-direction: row;
5   -webkit-flex-direction: row;
6   flex-wrap: wrap;
7   -webkit-flex-wrap: wrap;
8 }
```

## 2.2.5 Continguts i imatges flexibles

Els continguts i les imatges que formaran part de la pàgina web també han de ser flexibles. No és objectiu d'aquesta unitat com mostrar imatges flexibles; per tant, ens limitarem a tractar el contingut flexible.

En la taula 2.20 es poden consultar les propietats CSS que es poden assignar als elements que estan dins del contingut.

**TAULA 2.20.** Propietats CSS per al contingut flexible

Propietat	Descripció
<code>order</code>	Per establir l'ordre en què apareixen els elements d'una caixa flexible. Per defecte és 0.
<code>flex-grow</code>	Especifica el factor de creixement, és a dir, quant creix un element en relació amb els altres quan hi ha espai disponible en el contingut. Per defecte és 0.
<code>flex-shrink</code>	Especifica el factor de reducció, és a dir, quant decreixerà un element en relació amb els altres quan no hi hagi espai disponible en el contingut. Per defecte és 1.
<code>flex-basis</code>	Pren el mateix valor que la propietat <code>width</code> . Especifica la mida inicial de l'element abans de distribuir l'espai lliure amb les propietats anterior. Per defecte és <code>main-size (auto)</code> .
<code>flex</code>	Propietat abreujada de les anteriors.

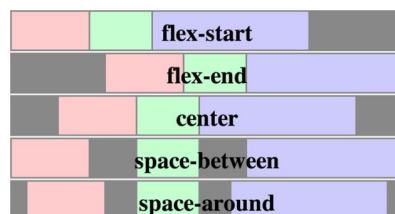
Les propietats per alinear els elements flexibles es poden consultar en la taula 2.21.

**TAULA 2.21.** Propietats CSS per alinear els elements flexibles

Propietat	Descripció
<code>justify-content</code>	Permet alinear els elements en l'eix principal. Pren per valor <code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>space-between</code> , <code>space-around</code> .
<code>align-items</code>	Permet alinear tots els elements en l'eix transversal. Pren per valor: <code>auto</code> , <code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>baseline</code> , <code>stretch</code> .
<code>align-self</code>	Permet alinear els elements individuals en l'eix transversal; així doncs, sobreescriu la propietat anterior.

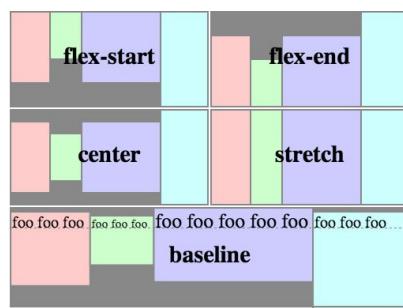
En la figura figura 2.40 i la figura 2.41 es pot apreciar com s'alineen els elements en funció de les propietats anterior.

**FIGURA 2.40.** Alineament sobre l'eix principal



Font: <https://www.w3.org/TR/css-flexbox-1/>

**FIGURA 2.41.** Alineament sobre l'eix transversal



Font: <https://www.w3.org/TR/css-flexbox-1/>

Vegem un exemple sobre com definir un contenidor i elements flexibles:

#### Estils per a un disseny flexible

```

1  body {
2      font: 18px Arial;
3      background: #eeeeee;
4  }
5
6  #main {
7      min-height: 450px;
8      margin: 0px;
9      padding: 0px;
10     display: -webkit-flex;
11     display: flex;
12     -webkit-flex-flow: row;
13     flex-flow: row;
14 }
15
16 #main > article {
17     margin: 4px;
18     padding: 5px;
19     border: 1px solid #ccc;
20     border-radius: 2pt;
21     background: #white;
22     -webkit-flex: 3 1 60%;
23     flex: 3 1 60%;
24     -webkit-order: 2;
25     order: 2;
26 }
27
28 #main > nav {
29     margin: 4px;
30     padding: 5px;
31     border: 1px solid #8888bb;
32     border-radius: 2pt;
33     background: #33b5e5;
34     -webkit-flex: 1 6 20%;
35     flex: 1 6 20%;
36     -webkit-order: 1;
37     order: 1;
38 }
39
40 #main > aside {
41     margin: 4px;
42     padding: 5px;
43     border: 1px solid #8888bb;
44     border-radius: 2pt;
45     background: #33b5e5;
46     -webkit-flex: 1 6 20%;
47     flex: 1 6 20%;
```

```
48     -webkit-order: 3;
49     order: 3;
50 }
51
52 header,
53 footer {
54   display: block;
55   margin: 2px;
56   padding: 15px;
57   min-height: 50px;
58   border: 1px solid #771199;
59   border-radius: 2pt;
60   background: #9933cc;
61   color: #ffffff;
62   text-shadow: 1px 1px #aaa;
63 }
64
65 @media all and (max-width: 640px) {
66   #main,
67   #page {
68     -webkit-flex-flow: column;
69     flex-flow: column;
70   }
71   #main > article,
72   #main > nav,
73   #main > aside {
74     -webkit-order: 0;
75     order: 0;
76   }
77   #main > nav,
78   #main > aside,
79   header,
80   footer {
81     min-height: 50px;
82   }
83 }
```

### Exemple d'utilització dels estils anteriors

```
1 <header>
2   <h1>Estudis de grau superior d'Informàtica</h1></header>
3 <div id='main'>
4   <article>article</article>
5   <nav>menú de navegació</nav>
6   <aside>Links d'interès</aside>
7 </div>
8 <footer>IOC – Institut Obert de Catalunya</footer>
```

El resultat es pot visualitzar en la figura 2.42.

**FIGURA 2.42.** Elements flexibles

Podeu provar l'anterior exemple en l'enllaç següent: [codepen.io/ioc-daw-m09/pen/YqrGYB](https://codepen.io/ioc-daw-m09/pen/YqrGYB).

Heu vist a l'exemple anterior la utilització de les propietats `display`, `flex` i `order`, i també com el contingut de la pàgina en HTML es simplifica moltíssim.

Fixeu-vos també com, per assegurar la compatibilitat amb altres navegadors, es repeteix la mateixa propietat CSS amb el seu prefix `-webkit-`.

## 2.3 Bootstrap 'framework'

En moltes ocasions, per crear una interfície web es pot partir d'una infraestructura o esquelet que ja conté molts dels elements necessaris per a la implementació d'una aplicació web.

Un **framework**, en general, és una estructura conceptual de suport definida, amb mòduls concrets de *software* que poden ser utilitzats com a base per a l'organització i el desenvolupament de *software*.

L'objectiu dels *frameworks* és proporcionar una estructura comuna perquè els desenvolupadors no ho hagin de fer des de zero i puguin reutilitzar el codi proporcionat.

Actualment existeixen diversos *frameworks* per poder desenvolupar el nostre lloc web, com poden ser: Bootstrap ([getbootstrap.com](https://getbootstrap.com)), Angular ([angularjs.org](https://angularjs.org)), Foundation ([foundation.zurb.com](https://foundation.zurb.com)), Leaf ([getleaf.com](https://getleaf.com)) o Materialize ([materializecss.com](https://materializecss.com)), entre molts d'altres.

En aquest apartat s'utilitzarà el **framework Bootstrap 3**. Les seves característiques més rellevants són:

- És un *framework* fàcil, ràpid i gratuït per al desenvolupament de la interfície web.

- Inclou plantilles basades en HTML i CSS, així com *plugins* JavaScript.
- Proporciona la capacitat de crear dissenys adaptatius.

### 2.3.1 Com obtenir i instal·lar Bootstrap

Hi ha dues formes d'iniciar l'ús de *Bootstrap* en el nostre lloc web:

1. Descarregar Bootstrap a [getbootstrap.com](http://getbootstrap.com) i seguir les instruccions que s'hi proporcionen.
2. Incloure Bootstrap d'una CDN (Content Delivery Network).

Una vegada descarregat, el descomprimirem i obtindrem els fitxers necessaris per utilitzar Bootstrap:

- **Carpeta css:** aquí hi trobem els fitxers bootstrap.css i bootstrap-theme.css. El primer és el que hem de vincular en els nostres projectes. I el segon és opcional, conté estils prefabricats per a botons, barres de navegació, etc. Els fitxers .min són els mateixos; la diferència és la mida i la lectura, ja que a la versió mínima s'han tret les tabulacions i salts de línia.
- **Carpeta js:** aquí hi trobem els fitxers bootstrap.js i la seva versió reduïda bootstrap.min.js. Aquest fitxer conté totes les llibreries JavaScript, rutines que fan que Bootstrap sigui més dinàmic i interactiu. Per exemple, menús desplegables i altres funcionalitats.
- **Carpeta fonts:** hi trobem les fonts *glyphicon* Bootstrap, que ens proporcionen l'opció d'utilitzar fonts enlloc d'imatges per mostrar icones bàsiques. Amb això aconseguim que el lloc sigui molt més ràpid de carregar i que consumeixi menys ample de banda. Tots els fitxers d'aquesta carpeta responen a la mateixa font, amb diferents formats que s'adequen als diferents navegadors.

Per vincular el *framework* Bootstrap descarregat a la nostra pàgina web:

#### Inclusió del 'framework' Bootstrap descarregat

```

1 <head>
2 ...
3 <link rel="stylesheet" href="css/bootstrap.min.css" />
4 </head>
5
6 <script src="js/bootstrap.min.js"></script>
```

De manera alternativa, podem utilitzar els estils Bootstrap des d'una CDN. Inclourem en el codi:

### Inclusió del 'framework' Bootstrap des d'una CDN

```

1 <head>
2 ...
3
4 <!-- estils CSS -->
5 <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/
   bootstrap/3.3.6/css/bootstrap.min.css" />
6
7 <!-- llibreria JQuery -->
8 <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/
   jquery.min.js"></script>
9
10 <!-- plugins JavaScript -->
11 <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/
   bootstrap.min.js"></script>
12
13 </head>
```

### 2.3.2 El contenidor

Bootstrap 3 requereix d'un element contenidor per embolcallar els continguts del lloc i col·locar el *grid bootstrap*. Podem triar dos tipus de contenidors:

- **.container**: defineix un contenidor adaptable d'amplada fixa.
- **.container-fluid**: defineix un contenidor adaptable d'ample total, és a dir, la seva amplada es correspon a l'amplada del *viewport*.

#### Definició d'un contenidor fluid

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 ...
5   <meta name="viewport" content="width=device-width, initial-
      scale=1">
6 </head>
7 <body>
8   <div class="container-fluid">
9     ...
10    </div>
11 </body>
12 </html>
```

### 2.3.3 Bootstrap 'grid'

Els sistema de graella, *grid*, s'utilitza per crear dissenys de pàgina a través d'una sèrie de files i columnes que acullen el contingut. Bootstrap inclou un sistema *grid responsive fluid* i *mobile first* de 12 columnes com a màxim.

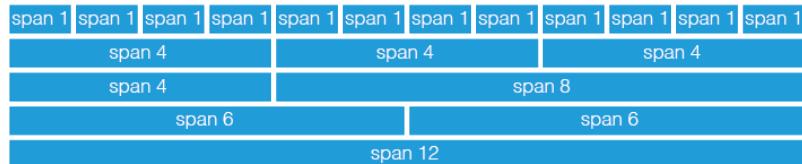
Aquest sistema funciona de la següent manera:

Els preprocessadors SASS i LESS són eines que permeten escriure pseudocodi CSS (variables, condicions, bucles, funcions) que serà convertit en CSS real.

- Les files, *rows*, han d'anar dins d'un contenidor d'amplada fixa o fluida. D'aquesta manera, s'estableix el seu alineament i farciment.
- El contingut es col·loca dins de les columnes.

En la figura 2.43 hi podem veure una representació del *grid* de Bootstrap en 12 columnes i com es poden combinar entre si.

**FIGURA 2.43.** Representació del Bootstrap 'grid'



Per obtenir més informació sobre els estils que conformen el *grid* de Bootstrap podeu consultar la web: [getbootstrap.com/css/#grid](http://getbootstrap.com/css/#grid).

El sistema Bootstrap *grid* ofereix quatre mides de quadrícula per als diferents tipus de dispositius. Així doncs, la classe que s'utilitza per definir cada fila de la quadrícula és `.row`, i les classes que s'utilitzen per definir les columnes són:

- `.col-xs-(num_columnes)`: per a telèfons mòbils ( $<768$  px).
- `.col-sm-(num_columnes)`: per a tauletes ( $\geq 768$  px).
- `.col-md-(num_columnes)`: per a ordinadors ( $\geq 992$  px).
- `.col-lg-(num_columnes)`: per a pantalles grans ( $\geq 1200$  px).

#### Exemple utilització de diferents mides de columnes

```

1 <div class="container">
2   <div class="row">
3     <div class="col-xs-12">
4       <p>Dispositius molt petits (Telèfons, amplada més petita de
768px)</p>
5     </div>
6   </div>
7   <div class="row">
8     <div class="col-xs-6" style="background-color:purple;height:40
px;">col-xs-4</div>
9     <div class="col-xs-6" style="background-color:pink;height:40px;
">col-xs-4</div>
10  </div>
11
12  <div class="row">
13    <div class="col-sm-12">
14      <p>Dispositius petits (tauletes, amplada més gran o igual a
768px)</p>
15    </div>
16  </div>
17  <div class="row">
18    <div class="col-sm-6" style="background-color:purple;height:40
px;">col-sm-4</div>
19    <div class="col-sm-6" style="background-color:pink;height:40px;
">col-sm-4</div>
20  </div>
21
22  <div class="row">
23    <div class="col-md-12">
```

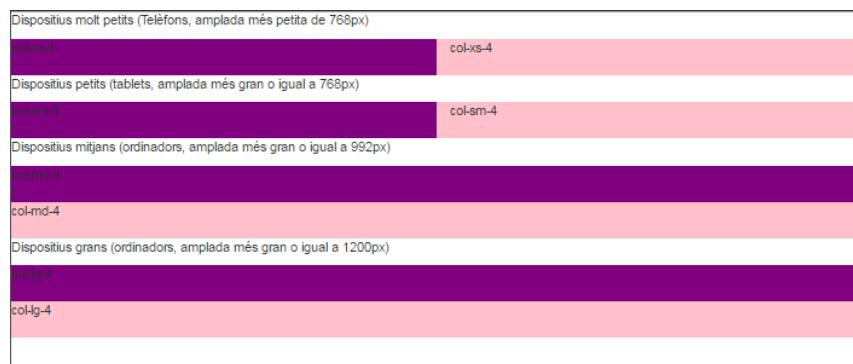
```

24   <p>Dispositius mitjans (ordinadors, amplada més gran o igual
25     a 992px)</p>
26   </div>
27 </div>
28 <div class="row">
29   <div class="col-md-6" style="background-color:purple;height:40
30     px;">col-md-4</div>
31   <div class="col-md-6" style="background-color:pink;height:40px;
32     ">col-md-4</div>
33 </div>
34 <div class="row">
35   <div class="col-lg-12">
36     <p>Dispositius grans (ordinadors, amplada més gran o igual a
37       1200px)</p>
38   </div>
39 </div>
40 <div class="row">
41   <div class="col-lg-6" style="background-color:purple;height:40
42     px;">col-lg-4</div>
43   <div class="col-lg-6" style="background-color:pink;height:40px;
44     ">col-lg-4</div>
45 </div>
46 </div>

```

El resultat serà el que es pot veure en la figura 2.44.

**FIGURA 2.44.** Representació de les mides de columnes



En la figura 2.44 podeu veure com totes les files es parteixen en dues columnes horizontals, però quan modifiquem la mida del navegador cada fila s'acomoda verticalment en funció de la classe i de la resolució. Així doncs, els *divs* col-sm-6 mantindran la seva estructura en dues columnes alineades horitzontalment, però quan la mida del *viewport* sigui menor a 768 px aquests s'acomodaran verticalment. Igualment passa amb les altres.

Podeu provar l'exemple anterior en el següent enllaç: [codepen.io/ioc-dawm09/pen/EKQPdZ](https://codepen.io/ioc-dawm09/pen/EKQPdZ).

A més a més, Bootstrap proporciona columnes d'amplada mixtes. Suposem que volem distribuir l'espai en dues columnes amb el 50% cadascuna, de tal manera que les dues columnes no ocupin sempre el 50%, sinó que en dispositius més grans de 1.200 px aquests *divs* ocupin el primer un 33% i el segon un 66%. Farem:

#### Exemple d'utilització de columnes d'amplada mixta

```

1 <div class="container">
2   <div class="row">
3     <div class="col-md-12">
4       <p>Divisió 50% i 50% </p>
5     </div>

```

```

6   </div>
7   <div class="row">
8     <div class="col-md-6" style="background-color:blue;">col-md-6</
9       div>
10    <div class="col-md-6" style="background-color:lightblue;">col-
11      md-6</div>
12    </div>
13    <div class="row">
14      <div class="col-md-12">
15        <p>Divisió 50% 50% per dispositius mitjans, 33% 66% per a
16          grans</p>
17      </div>
18    </div>
19    <div class="row">
20      <div class="col-md-6 col-lg-4" style="background-color:blue;">
21        col-md-6</div>
22      <div class="col-md-6 col-lg-8" style="background-color:
23        lightblue;">col-md-6</div>
24    </div>
25    <div class="row">
26      <div class="col-md-12">
27        <p>Divisió 33% 66% dispositius petits, 50% 50% mitjans i 25%
28          75% per a grans. </p>
29    </div>

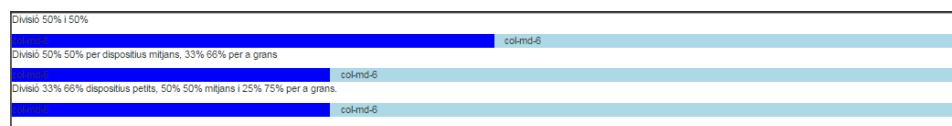
```

Divisió 50%/50% per a dispositius mitjans, 33%/66% per a grans; Divisió 33%/66% per a dispositius petits, 50%/50% per a mitjans i 25%/75% per a grans.

Podeu provar l'anterior exemple en l'enllaç: [codepen.io/ioc-dawm09/pen/GZQoVP](https://codepen.io/ioc-dawm09/pen/GZQoVP).

Vegeu a la figura 2.45 com la mida de les columnes es redimensiona en funció de l'amplada i la resolució del *viewport*.

**FIGURA 2.45.** Representació de columnes d'amplada mixta



De la mateixa manera, en Bootstrap també disposem d'utilitats per ocultar i mostrar un element basant-nos en la mida del dispositiu i la resolució del *viewport*.

Així doncs, podem usar les classes següents per mostrar elements *div*: *.visible-xs*, *.visible-sm*, *.visible-md* i *.visible-lg*. I per amagar elements *div* basant-nos en la mida del *viewport*: *.hidden-xs*, *.hidden-sm*, *.hidden-md* i *.hidden-lg*.

## Netejar columnes

A causa dels quatre nivells de quadrícula disponibles, ens trobem amb certs problemes quan canviem la disposició en les diferents resolucions, ja que les columnes poden no tenir el mateix contingut i algunes són més altes que d'altres. Per solucionar-ho utilitzem la classe `.clearfix`.

### Exemple sense "clearfix"

```

1 <div class="container">
2 <div class="row">
3   <div class="col-xs-6 col-sm-3" style="background-color:red;
4     height:40px;">.col-xs-6 .col-sm-3</div>
5   <div class="col-xs-6 col-sm-3" style="background-color:blue;">
6     col-xs-6 .col-sm-3</div>
7   <div class="col-xs-6 col-sm-3" style="background-color:green;">
8     .col-xs-6 .col-sm-3 (left)</div>
9   <div class="col-xs-6 col-sm-3" style="background-color:yellow;">
10    >.col-xs-6 .col-sm-3 (right)</div>
11 </div>
12 </div>
```

El resultat que obtindreu és el que es pot veure en la figura 2.46.

**FIGURA 2.46.** Visualització sense "clearfix"



L'exemple el podeu provar en el següent enllaç: [codepen.io/ioc-dawm09/pen/ONQXax](https://codepen.io/ioc-dawm09/pen/ONQXax).

Vegeu a l'exemple anterior on es posiciona la columna verda en canviar la mida del *viewport*.

### Exemple amb "clearfix"

```

1 <div class="container">
2 <div class="row">
3   <div class="col-xs-6 col-sm-3" style="background-color:red;
4     height:40px;">.col-xs-6 .col-sm-3</div>
5   <div class="col-xs-6 col-sm-3" style="background-color:blue;">
6     col-xs-6 .col-sm-3</div>
7   <div class="clearfix visible-xs"></div>
8   <div class="col-xs-6 col-sm-3" style="background-color:green;">
9     .col-xs-6 .col-sm-3 (left)</div>
10  <div class="col-xs-6 col-sm-3" style="background-color:yellow;">
11    >.col-xs-6 .col-sm-3 (right)</div>
12 </div>
13 </div>
```

El resultat és el que es pot veure en la figura 2.47.

**FIGURA 2.47.** Visualització amb "clearfix"



Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-dawm09/pen/WwMxYJ](https://codepen.io/ioc-dawm09/pen/WwMxYJ).

Vegeu en l'exemple anterior com, amb la utilització de la classe `clearfix`, la columna verda s'alinea amb la columna de la seva esquerra.

## Desplaçar columnes

Utilitzem la classe `-offset-` per desplaçar qualsevol columna cap a la dreta. Aquesta classe augmenta la mida del marge esquerre de la columna en una quantitat equivalent a aquest nombre de columnes. Per exemple, la classe `.col-md-offset-6` desplaça la columna cap a la dreta o dóna marges a l'esquerra d'una amplada equivalent a sis columnes.

### Exemple de desplaçament de columnes

```

1  <div class="container">
2    <div class="row">
3      <div class="col-md-4" style="background-color: red;">.col-md-4</
4        div>
5        <div class="col-md-4 col-md-offset-4" style="background-color:
6          blue;">.col-md-4 .col-md-offset-4</div>
7      </div>
8
9      <div class="row">
10       <div class="col-md-3 col-md-offset-3" style="background-color:
11         green;">.col-md-3 .col-md-offset-3</div>
12       <div class="col-md-3 col-md-offset-3" style="background-color:
13         yellow;">.col-md-3 .col-md-offset-3</div>
14     </div>
15   </div>
```

El resultat és el que es pot veure en la figura 2.48.

**FIGURA 2.48.** Exemple de desplaçament de columnes



L'anterior exemple el podeu provar en el següent enllaç: [codepen.io/ioc-dawm09/pen/wGyWNy](https://codepen.io/ioc-dawm09/pen/wGyWNy).

Vegeu com entre la columna vermella i blava hi ha un offset igual a la mida de tres columnes, i com les columnes verda, groga i també la vermella tenen un offset de tres columnes.

## Reordenar elements

Amb les classes `-pull-`, el que fem és estirar l'element a la seva esquerra el nombre de columnes que especificuem. I amb les classes `-push-`, el que fem és empènyer l'element a la dreta el nombre de columnes que especificuem.

### Exemple de reordenació de columnes

```

1 <div class="container">
2 <div class="row">
3   <div class="col-md-9 col-md-push-3" style="background-color:red
      ;">.col-md-9 .col-md-push-3</div>
4   <div class="col-md-3 col-md-pull-9" style="background-color:
      blue;">.col-md-3 .col-md-pull-9</div>
5 </div>
6 </div>
```

El resultat es pot veure en la figura figura 2.49.

**FIGURA 2.49.** Ordenació d'elements



Podeu provar l'exemple anterior en el següent enllaç: [codepen.io/ioc-daw-m09/pen/WwMxWy](http://codepen.io/ioc-daw-m09/pen/WwMxWy).

Vegeu que la columna vermella l'hem empès cap a la dreta, i que la columna blava l'hem estirat cap a l'esquerra, i així les veiem en l'ordre invers.

### 2.3.4 Elements

El *framework* Bootstrap 3 proporciona estils per representar múltiples elements, com són la tipografia, les taules, les imatges, el formularis, els botons, etc. Totes les seves propietats es poden consultar a: [getbootstrap.com/css](http://getbootstrap.com/css).

Vegeu en la taula 2.22 algunes classes definides per Bootstrap referents a la tipografia.

**TAULA 2.22.** Classes Bootstrap 3 per a la tipografia

Classe	Descripció
.lead	Per destacar un text.
.text-left	Alinea el text a l'esquerra.
.text-center	Centra el text.
.text-right	Alinea el text a la dreta.
.list-inline	Mostra els elements de la llista horitzontalment.
.list-unstyled	Per definir una llista sense vinyeta.

Vegeu en la taula 2.23 algunes classes definides per Bootstrap referents a les taules.

**TAULA 2.23.** Classes Bootstrap 3 per a les taules

Classe	Descripció
.table	Aplica els estils bàsics de Bootstrap3 a les taules.
.table-striped	Altera el color de fons per millorar la llegibilitat dels continguts.
.table-bordered	Mostra les quatre vores de cada cel·la de la taula.

**TAULA 2.23** (continuació)

Classe	Descripció
.table-hover	Modifica lleugerament l'aspecte de les files quan l'usuari passa el ratolí per damunt d'aquestes.
.table-condensed	Mostra els continguts de la taula de manera més compacta.

Vegeu en la taula 2.24 algunes classes definides per Bootstrap per decorar les imatges.

**TAULA 2.24.** Classes Bootstrap 3 per a les imatges

Classe	Descripció
.img-rounded	Arrodoneix els vèrtexs de la imatge.
.img-thumbnail	Mostra la imatge amb un farciment blanc i una vora fina. Simulant l'aspecte de fotografies antigues.
.img-circle	Converteix la imatge en un cercle.

Vegeu en la taula 2.25 algunes classes definides per Bootstrap per als formularis.

**TAULA 2.25.** Classes Bootstrap 3 per als formularis

Classe	Descripció
.form-control	Estableix amplada <i>width</i> : 100% als elements <input>, <textarea> i <select>.
.form-group	Agrupa el camp del formulari amb el seu <label>.
.form-horizontal	Agrupa el camp del formulari amb el seu <label> horitzontalment.

A més a més, també inclou components CSS que cobreixen les necessitats més habituals: icones, menús desplegables, barres de navegació, barres de progrés, objectes multimèdia, etc.

Podeu consultar com utilitzar i incloure aquests components a: [getbootstrap.com/components](http://getbootstrap.com/components).

#### Exemple de menú desplegable utilitzant Bootstrap 3

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5    <meta charset="utf-8">
6    <title>Primera pàgina web</title>
7    <meta name="viewport" content="width=device-width, initial-
       scale=1">
8    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/
       bootstrap/3.3.6/css/bootstrap.min.css">
9    <script src="https://ajax.googleapis.com/ajax/libs/jquery/
       1.12.0/jquery.min.js"></script>
10   <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/
       bootstrap.min.js"></script>
11   </head>
12
13  <body>
14
15    <div class="container">
```

```

16 <h2>Menú desplegable</h2>
17 <div class="dropdown">
18   <button class="btn btn-info dropdown-toggle" type="button"
19     data-toggle="dropdown">Exemple botó 1
20   <span class="caret"></span></button>
21   <ul class="dropdown-menu">
22     <li><a href="#">HTML</a></li>
23     <li><a href="#">CSS</a></li>
24     <li><a href="#">JavaScript</a></li>
25     <li class="divider"></li>
26     <li><a href="#">Qui som</a></li>
27   </ul>
28 </div>
29 </div>
30 <div class="container">
31   <div class="row">
32     <div class="jumbotron">
33       <h1>Text destacat: programació web</h1>
34       <p>Utilitza els principals llenguatges de programació:
35         HTML, CSS i JavaScript.</p>
36       <p><a class="btn btn-primary btn-lg" role="button">Més...
37         </a></p>
38     </div>
39   </div>
40 </body>
41 </html>

```

El resultat serà el que es pot veure en la figura 2.50.

**FIGURA 2.50.** Menú desplegable i jumbotron



L'anterior exemple el podeu provar en el següent enllaç: [codepen.io/ioc-dawm09/pen/LNJvKo](https://codepen.io/ioc-dawm09/pen/LNJvKo).

Cal tenir en compte les consideracions següents:

- Per crear un menú s'utilitza la classe `.dropdown` sobre l'etiqueta `<div>`.
- El botó `button` serà l'element que obre el menú. També podria ser un enllaç.
- Per donar estil al botó s'han utilitzat les classes `.btn` i `.btn-info`, les quals fan que el botó es vegi amb aquesta forma i color.
- Perquè el botó obri el menú cal utilitzar la classe `.dropdown-toggle` i l'atribut `data-toggle="dropdown"`.
- Perquè es vegi la fletxa que indica que s'obrirà un menú en fer clic al botó s'utilitza la classe `.caret`.

- Per crear el menú s'utilitza una llista <ul> amb la classe .dropdown-menu.
- Cada element del menú es representarà amb l'element <li>.
- Per crear el text destacat s'utilitza la classe que ofereix Bootstrap, anomenada .jumbotron.

Vegeu un exemple de formulari amb Bootstrap per donar d'alta un usuari:

#### Exemple de formulari utilitzant Bootstrap 3

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Primera pàgina web</title>
6      <meta name="viewport" content="width=device-width, initial-
    scale=1">
7      <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery
    /1.12.0/jquery.min.js"></script>
9      <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/
    bootstrap.min.js"></script>
10 </head>
11
12 <body>
13
14     <div class="container">
15         <div class="row text-left">
16             <div class="col-xs-6 ">
17                 <h2>Alta Usuari:</h2>
18                 <form role="form">
19                     <div class="form-group">
20                         <label for="nom_1">Nom:</label>
21                         <input type="text" class="form-control" id="nom_1" placeholder="Escriu el teu nom">
22                     </div>
23                     <div class="form-group">
24                         <label for="email_1">Correu electrònic:</label>
25                         <input type="email" class="form-control" id="email_1"
                            placeholder="Escriu el teu correu electrònic">
26                     </div>
27                     <div class="form-group">
28                         <label for="contrasenya_1">Contrasenya:</label>
29                         <input type="password" class="form-control" id="contrasenya_1"
                            rows="4" placeholder="Escriu la contrasenya"/>
30                     </div>
31                     <button type="submit" class="btn btn-default">Enviar</button>
32                 </form>
33             </div>
34         </div>
35     </div>
36 </body>
37 </html>
```

El resultat es pot veure en la figura 2.51.

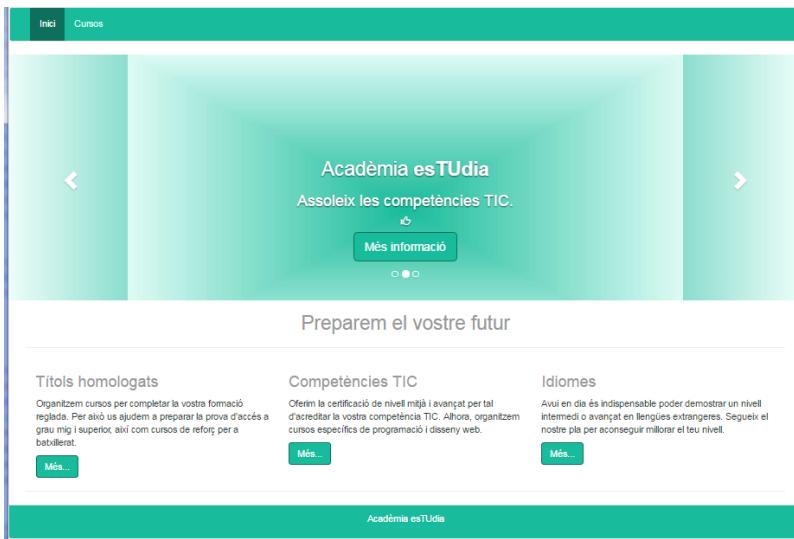
**FIGURA 2.51.** Formulari amb Bootstrap

The screenshot shows a web browser window with a blue header bar. The title bar says "Primera página web" and the address bar says "file:///pagFormulariBS.html". The main content area has a heading "Alta Usuari:". Below it are three input fields with placeholder text: "Escríu el teu nom" for Name, "Escríu el teu correu electrònic" for Email, and "Escríu la contrasenya" for Password. At the bottom is a "Enviar" button.

## 2.4 Cas pràctic amb Bootstrap: Acadèmia esTUDia

En aquest cas pràctic creareu un document web utilitzant el *framework* Bootstrap per a una acadèmia que fa cursos. L'aparença que ha de tenir és la que es mostra en la figura 2.52.

**FIGURA 2.52.** Aparència del lloc web d'esTUDia



Visualització en el navegador de la pàgina web inicial

Primerament, creareu la capçalera amb el menú de navegació utilitzant les classes `.navbar` i `.navbar-default`. Aquesta barra té l'identificador `bs-example-navbar-collapse-1` i mostrarà els ítems “Inici” i “Cursos”, però si la mida de la pantalla es redueix es reduirà en un sol botó.

El codi HTML serà com el de l'exemple:

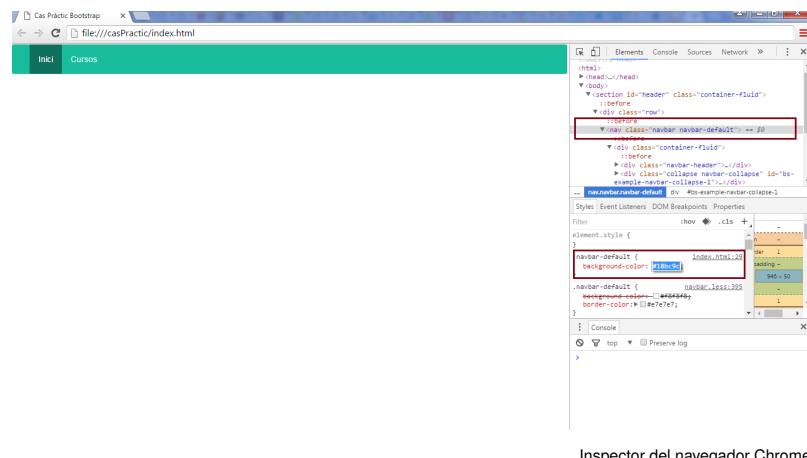
#### Codi HTML per a la barra de navegació

```

1 <section id="header" class="container-fluid">
2   <div class="row">
3     <nav class="navbar navbar-default">
4       <div class="container-fluid">
5         <div class="navbar-header">
6           <button type="button" class="navbar-toggle" data-
7             toggle="collapse" data-target="#bs-example-navbar-
8             collapse-1">
9             <span class="sr-only">Menú</span>
10            <span class="icon-bar"></span>
11            <span class="icon-bar"></span>
12            <span class="icon-bar"></span>
13           </button>
14         </div>
15         <div class="collapse navbar-collapse" id="bs-example-
16           navbar-collapse-1">
17           <ul class="nav navbar-nav">
18             <li class="active"><a href="#">Inici</a></li>
19             <li><a href="#seccio2">Cursos</a></li>
20           </ul>
21         </div>
22       </div>
23     </nav>
24   </div>
25 </section>
```

A més a més, també serà necessari modificar alguns estils CSS del *framework* Bootstrap perquè es vegi amb els colors escollits. Per tant, utilitzareu l'inspector del navegador, tal com es mostra en la figura 2.53, per tal de localitzar els estils que cal modificar. Atès que no es recomana modificar el *framework*, serà necessari crear un estil amb el mateix nom i s'hi introduiran les propietats que es volen modificar o afegir.

**FIGURA 2.53.** Visualització de l'inspector per cercar els estils



Inspector del navegador Chrome

Els estils CSS referents la barra de navegació que cal modificar per tal que la seva aparença sigui la desitjada són:

#### Estil per a la barra de navegació

```

1 .navbar-default {
2   background-color: #18bc9c;
```

```

3 }
4 .navbar-default .navbar-nav > li > a {
5   color: #DBFBF4;
6 }
7 .navbar-default .navbar-nav > li > a:hover,
8 .navbar-default .navbar-nav > li > a:focus {
9   color: #fff;
10 }
11 .navbar-default .navbar-nav > .active > a,
12 .navbar-default .navbar-nav > .active > a:hover,
13 .navbar-default .navbar-nav > .active > a:focus {
14   color: #fff;
15   background-color: #0E705C;
16 }

```

Seguidament, per crear el carrusel amb les imatges caldrà tenir primer les imatges que mostrarà. En aquest cas sempre mostra la mateixa imatge amb el fons verd de 900 x 400 px.

Les classes que proporciona Bootstrap per crear el carrusel es poden veure en la taula 2.26.

**TAULA 2.26.** Classes que proporciona Bootstrap per crear un carrusel

Classe	Descripció
.slide	Afegeix una transició CSS i efecte d'animació quan es passa d'un element al següent.
.carousel-indicator	Afegeix indicadors per al carrusel. Aquests són els petits punts a la part inferior de cada diapositiva.
.carousel-inner	Afegeix les diapositives al carrusel.
.item	Especifica el contingut de cada diapositiva.
.left carousel-control	Afegeix un botó esquerre al carrusel , que permet a l'usuari tornar entre les diapositives.
.right carousel-control	Afegeix un botó dret al carrusel, que permet a l'usuari tornar entre les diapositives.
.carousel-caption	Permet especificar un títol.

Així doncs, el codi per inserir el carrusel és:

#### Codi HTML per inserir un carrusel

```

1 <section id="secciol" class="container-fluid">
2   <div class="row">
3     <div id="myCarousel" class="carousel slide" data-ride="carousel">
4       <!-- Indicadors -->
5       <ol class="carousel-indicators">
6         <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
7         <li data-target="#myCarousel" data-slide-to="1"></li>
8         <li data-target="#myCarousel" data-slide-to="2"></li>
9       </ol>
10
11      <!-- slides -->
12      <div class="carousel-inner" role="listbox">
13        <div class="item active">
14          
16          <div class="container">
17            <div class="carousel-caption">

```

```

17          <h2>Acadèmia <strong>esTUdia</strong></h2>
18          <h3>Us facilitem l'obtenció de títols oficials.</h3>
19          >
20          <p class="glyphicon glyphicon-thumbs-up"></p>
21          <p><a class="btn btn-lg btn-primary" href="#seccio2
22              " role="button">Més informació</a></p>
23          </div>
24      </div>
25      <div class="item">
26          
28          <div class="container">
29              <div class="carousel-caption">
30                  <h2>Acadèmia <strong>esTUdia</strong></h2>
31                  <h3>Assoleix les competències TIC.</h3>
32                  <p class="glyphicon glyphicon-thumbs-up"></p>
33                  <p><a class="btn btn-lg btn-primary" href="#seccio2
34                      " role="button">Més informació</a></p>
35          </div>
36      </div>
37      <div class="item">
38          
40          <div class="container">
41              <div class="carousel-caption">
42                  <h2>Acadèmia <strong>esTUdia</strong></h2>
43                  <h3>Millora la teva fluïdesa en llengües
44                      extrangeres.</h3>
45                  <p class="glyphicon glyphicon-thumbs-up"></p>
46                  <p><a class="btn btn-lg btn-primary" href="#seccio2
47                      " role="button">Més informació</a></p>
48          </div>
49      </div>
50
51      <!-- controls a dreta i esquerra -->
52      <a class="left carousel-control" href="#myCarousel" role=
53          "button" data-slide="prev">
54          <span class="glyphicon glyphicon-chevron-left" aria-
55              hidden="true"></span>
56          <span class="sr-only">Anterior</span>
57      </a>
58      <a class="right carousel-control" href="#myCarousel" role
59          ="button" data-slide="next">
60          <span class="glyphicon glyphicon-chevron-right" aria-
61              hidden="true"></span>
62          <span class="sr-only">Següent</span>
63      </a>
64  </div>
65 </div>
66 </div>
67 </section>

```

També es poden modificar els estils CSS de Bootstrap per canviar l'aspecte del carrusel. Per a això es poden crear aquests estils:

#### Estils per al carrusel

```

1 .carousel-inner > .item > img,
2 .carousel-inner > .item > a > img {
3     width: 70%;

```

```

4     margin: auto;
5   }
6   .carousel-control.right {
7     background-image: linear-gradient(to right, #18bc9c 0%, #DBFBF4
8       100%);
9   }
10  .carousel-control.left {
11    background-image: linear-gradient(to left, #18bc9c 0%, #DBFBF4
12      100%);
13  }

```

Després cal crear els articles, per a la qual cosa serà necessari crear un *grid* de dues files. En la primera s'hi localitza el títol, i en la segona els diferents articles, els quals, en funció de la mida de la pantalla del navegador, es mostraran en una, dues o tres columnes.

El codi HTML per inserir els articles del lloc web és:

#### Codi HTML per inserir els articles

```

1 <section id="seccio2" class="container">
2   <div class="row text-center">
3     <h2>Preparem el vostre futur</h2>
4     <hr class="small" />
5   </div>
6
7   <div class="row">
8     <div class="col-xs-12 col-sm-6 col-md-4">
9       <h3>Títols homologats</h3>
10      <figure></figure>
11      <p>Organitzem cursos per completar la vostra formació
12        reglada. Per això us ajudem a preparar la prova d'accés
13        a grau mig i superior, així com cursos de reforç per a
14        batxillerat.</p>
15      <a href="" class="btn btn-primary">Més...</a>
16    </div>
17
18    <div class="col-xs-12 col-sm-6 col-md-4">
19      <h3>Competències TIC</h3>
20      <figure></figure>
21      <p>Oferim la certificació de nivell mitjà i avançat per tal
22        d'acreditar la vostra competència TIC. Alhora,
23        organitzem cursos específics de programació i disseny
24        web. </p>
25      <a href="" class="btn btn-primary">Més...</a>
26    </div>
27
28    <div class="col-xs-12 col-sm-6 col-md-4">
29      <h3>Idiomes</h3>
30      <figure></figure>
31      <p>Avui en dia és indispensable poder demostrar un nivell
32        intermedi o avançat en llengües estrangeres. Segueix el
33        nostre pla per aconseguir millorar el teu nivell.</p>
34      <a href="" class="btn btn-primary">Més...</a>
35    </div>
36  </div>
37
38  <div class="row text-center">
39    <hr class="small" />
40  </div>
41
42 </section>

```

I els estils que cal modificar perquè el botons i el text es vegin del color adient són:

### Estils per als botons i text del lloc web

```

1 .btn-primary {
2   background-color:#18bc9c;
3   border-color: #0E705C
4 }
5
6 .btn-primary:hover, .btn-primary:active , .btn-primary:focus, .
7   btn-primary:active:focus{
8   background-color:#0E705C;
9   border-color: #041E18;
10 }
11 div[class*="row"] > h2, div[class*="col-"] > h3 {
12   color:#999999;
13 }
```

Finalment, es crea el peu de pàgina utilitzant l'estil `.panel-footer`.

### Codi HTML per inserir el peu de pàgina

```

1 <section class="container-fluid">
2   <div class="row">
3     <footer class="panel-footer">
4       <p class="text-center">Acadèmia esTUDia</p>
5     </footer>
6   </div>
7 </section>
```

I els estils que cal modificar són:

### Estil per modificar el color del peu de pàgina

```

1 .panel-footer {
2   color: #fff;
3   background-color: #18bc9c;
4 }
```

Així doncs, el codi resultant del document web utilitzant el *framework* Bootstrap serà:

### Codi del lloc web

```

1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-
6     scale=1">
7   <title>Cas Pràctic Bootstrap</title>
8
9   <!-- Inclusió del framework Bootstrap -->
10  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
11  <script src="https://ajax.googleapis.com/ajax/libs/jquery/
12    1.12.2/jquery.min.js"></script>
13  <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/
14    bootstrap.min.js"></script>
15
16  <!-- Estils que modifiquen i/o amplien el framework -->
17  <style>
18    .carousel-inner > .item > img,
19    .carousel-inner > .item > a > img {
```

```
20 .carousel-control.right {
21   background-image: linear-gradient(to right, #18bc9c 0%, #
22   DBFBF4 100%);
23 }
24 .carousel-control.left {
25   background-image: linear-gradient(to left, #18bc9c 0%, #
26   DBFBF4 100%);
27 }
28 .navbar-default {
29   background-color: #18bc9c;
30 }
31 .navbar-default .navbar-nav > li > a {
32   color: #DBFBF4;
33 }
34 .navbar-default .navbar-nav > li > a:hover,
35 .navbar-default .navbar-nav > li > a:focus {
36   color: #fff;
37 }
38 .navbar-default .navbar-nav > .active > a,
39 .navbar-default .navbar-nav > .active > a:hover,
40 .navbar-default .navbar-nav > .active > a:focus {
41   color: #fff;
42   background-color: #0E705C;
43 }
44 .btn-primary {
45   background-color:#18bc9c;
46   border-color: #0E705C
47 }
48 .btn-primary:hover, .btn-primary:active , .btn-primary:focus,
49   .btn-primary:active:focus{
50   background-color:#0E705C;
51   border-color: #041E18;
52 }
53 .panel-footer {
54   color: #fff;
55   background-color: #18bc9c;
56 }
57 div[class*="row"] > h2, div[class*="col-"] > h3 {
58   color:#999999;
59 }
60 }
61 </style>
62 </head>
63 <body>
64
65 <section id="header" class="container-fluid">
66 <div class="row">
67   <nav class="navbar navbar-default">
68     <div class="container-fluid">
69       <div class="navbar-header">
70         <button type="button" class="navbar-toggle" data-toggle
71           ="collapse" data-target="#bs-example-navbar-
72           collapse-1">
73           <span class="sr-only">Menú</span>
74           <span class="icon-bar"></span>
75           <span class="icon-bar"></span>
76           <span class="icon-bar"></span>
77         </button>
78       </div>
79       <div class="collapse navbar-collapse" id="bs-example-
80           navbar-collapse-1">
81         <ul class="nav navbar-nav">
82           <li class="active"><a href="#">Inici</a></li>
```

```
81             <li><a href="#seccio2">Cursos</a></li>
82         </ul>
83     </div>
84     </div>
85     </nav>
86 </div>
87 </section>
88
89 <section id="seccio1" class="container-fluid">
90     <div class="row">
91         <div id="myCarousel" class="carousel slide" data-ride="carousel">
92             <!-- Indicadors -->
93             <ol class="carousel-indicators">
94                 <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
95                 <li data-target="#myCarousel" data-slide-to="1"></li>
96                 <li data-target="#myCarousel" data-slide-to="2"></li>
97             </ol>
98
99             <!-- slides -->
100            <div class="carousel-inner" role="listbox">
101                <div class="item active">
102                    
103                    <div class="container">
104                        <div class="carousel-caption">
105                            <h2>Acadèmia <strong>esTUDia</strong></h2>
106                            <h3>Us facilitem l'obtenció de títols oficials.</h3>
107                            >
108                            <p class="glyphicon glyphicon-thumbs-up"></p>
109                            <p><a class="btn btn-lg btn-primary" href="#seccio2" role="button">Més informació</a></p>
110                        </div>
111                    </div>
112
113                <div class="item">
114                    
115                    <div class="container">
116                        <div class="carousel-caption">
117                            <h2>Acadèmia <strong>esTUDia</strong></h2>
118                            <h3>Assoleix les competències TIC.</h3>
119                            <p class="glyphicon glyphicon-thumbs-up"></p>
120                            <p><a class="btn btn-lg btn-primary" href="#seccio2" role="button">Més informació</a></p>
121
122                </div>
123            </div>
124
125            <div class="item">
126                
127                <div class="container">
128                    <div class="carousel-caption">
129                        <h2>Acadèmia <strong>esTUDia</strong></h2>
130                        <h3>Millora la teva fluència en llengües
131                            estrangeres.</h3>
132                        <p class="glyphicon glyphicon-thumbs-up"></p>
133                        <p><a class="btn btn-lg btn-primary" href="#seccio2" role="button">Més informació</a></p>
134
135                </div>
136            </div>

```

```
137
138     <!-- Controls a dreta i esquerra -->
139     <a class="left carousel-control" href="#myCarousel" role=
140         "button" data-slide="prev">
141         <span class="glyphicon glyphicon-chevron-left" aria-
142             hidden="true"></span>
143         <span class="sr-only">Anterior</span>
144     </a>
145     <a class="right carousel-control" href="#myCarousel" role
146         ="button" data-slide="next">
147         <span class="glyphicon glyphicon-chevron-right" aria-
148             hidden="true"></span>
149         <span class="sr-only">Següent</span>
150     </a>
151     </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 <section id="seccio2" class="container">
159     <div class="row text-center">
160         <h2>Preparem el vostre futur</h2>
161         <hr class="small" />
162     </div>
163     <div class="row">
164         <div class="col-xs-12 col-sm-6 col-md-4">
165             <h3>Títols homologats</h3>
166             <figure></figure>
167             <p>Organitzem cursos per completar la vostra formació
168                 reglada. Per això us ajudem a preparar la prova d'accés
169                 a grau mig i superior, així com cursos de reforç per a
170                 batxillerat.</p>
171             <a href="" class="btn btn-primary">Més...</a>
172         </div>
173         <div class="col-xs-12 col-sm-6 col-md-4">
174             <h3>Competències TIC</h3>
175             <figure></figure>
176             <p>Oferim la certificació de nivell mitjà i avançat per tal
177                 d'acreditar la vostra competència TIC. Alhora,
178                 organitzem cursos específics de programació i disseny
179                 web. </p>
180             <a href="" class="btn btn-primary">Més...</a>
181         </div>
182         <div class="col-xs-12 col-sm-6 col-md-4">
183             <h3>Idiomes</h3>
184             <figure></figure>
185             <p>Avui en dia és indispensable poder demostrar un nivell
186                 intermedi o avançat en llengües estrangeres. Segueix el
187                 nostre pla per aconseguir millorar el teu nivell.</p>
188             <a href="" class="btn btn-primary">Més...</a>
189         </div>
190     </div>
191 </div>
```

Podeu provar l'anterior cas pràctic en el següent enllaç: [codepen.io/ioc-dawm09/pen/bpzZoR](https://codepen.io/ioc-dawm09/pen/bpzZoR).

### 3. Generació de llocs web: prototips i plantilles

Per fer el disseny i el desenvolupament d'un lloc web s'ha de tenir en compte quin és el contingut, l'objectiu i qui serà el perfil dels usuaris finals.

Primerament, tenir aquesta informació és cabdal no tan sols per planificar i definir el seu disseny i les eines a utilitzar, sinó també per esbrinar les tècniques posteriors de posicionament.

En segon lloc, s'ha de fer el disseny del lloc web, alhora que cal pensar quines imatges, continguts i components en formaran part. També cal decidir en quin ordre apareixeran els diferents elements i la disposició de la informació en la pàgina.

Finalment, cal decidir com s'implementa el disseny, és a dir, quins llenguatges i eines s'utilitzaran per a cada component que s'hi vulgui incloure, així com avaluar si es comença el lloc des de zero o s'utilitza un *framework* com pot ser Bootstrap, o bé si el disseny s'adaptaria fàcilment a una plantilla web.

#### 3.1 Fases d'un projecte web

El procés de desenvolupament d'un projecte web es porta a terme en diferents fases per tal d'assegurar-ne l'èxit. Les fases estàndard de creació d'un lloc web acostumen a ser planificació, disseny, desenvolupament, publicació i manteniment.

Cadascuna d'aquestes fases executen unes tasques concretes de manera iterativa, entenent així que una fase no comença fins que no ha acabat l'anterior; no obstant això, en funció de l'envergadura del projecte es posa més èmfasi en una fase que en una altra.

##### 3.1.1 Planificació

Quan iniciem un projecte per a la creació d'un nou lloc web és important tenir present una sèrie de factors abans d'iniciar qualsevol disseny, com són:

- L'objectiu per al qual es crea aquest lloc web.
- Els usuaris i les seves necessitats.
- El contingut que s'hi vol publicar.
- El nom del lloc web, comprovar que el domini està lliure i registrar-lo.

- Investigar llocs similars, la competència.

D'aquesta manera, el client ens ha de fer saber l'objectiu que voldrà aconseguir amb la pàgina web, per a així poder plantejar un disseny adequat i adaptat a les seves necessitats. Aquesta informació l'obtindrem mitjançant entrevistes o qüestionaris.

És important saber quins continguts s'hi volen publicar, per així poder adaptar el disseny al contingut i no a l'inrevés. Tampoc és necessari tenir ja tots els textos i les imatges, però sí una idea clara del que s'ha de publicar.

També és interessant estudiar altres llocs web similars i com estan posicionats. Això ens ajudarà a recollir idees per fer el disseny posterior.

En definitiva, aquesta fase està centrada en la recollida de dades i en l'anàlisi de la informació a incloure en funció dels objectius i usuaris finals, així com també tots aquells requeriments tècnics per la interfície o *front-end* i per al *back-end*, com ara llenguatges i aplicacions de desenvolupament.

### 3.1.2 Disseny

En aquesta fase cal establir els criteris de disseny i el model d'interfície. En primer lloc, caldrà clarificar i estructurar visualment com s'organitza el contingut de cada pàgina que formarà part del lloc web. Per això també serà necessari establir l'ordre de prioritat dels continguts i identificar els missatges més importants que compondran cada pàgina amb independència del dispositiu en el qual es visualitzi la web.

Posteriorment, s'hi inclourà tot allò que permeti visualitzar l'estruatura de la informació, és a dir, els colors que s'utilitzaran, els diferents tipus de lletra, les imatges de fons, els formularis, els botons, etc.

En finalitzar aquesta fase s'ha d'obtenir un prototip on es pugui visualitzar correctament la interfície de totes les planes del lloc web en els diferents tipus de dispositius (ordinadors, tauletes, mòbils) i que permeti avaluar la usabilitat, l'accressibilitat i la navegabilitat del lloc web.

### 3.1.3 Desenvolupament

Quan ja s'ha obtingut el disseny aprovat pel client comença la fase de desenvolupament. És quan es codifica i s'implementen les funcionalitats que ha d'oferir el lloc web.

Cal destacar que aquesta fase es pot fer des de zero, utilitzant una plantilla, o bé emprant un CMS. En tot cas, es parteix del disseny del prototip obtingut en la fase

#### Content Management System, CMS

Plataformes *software* que permeten desenvolupar un lloc web destinat a la gestió de continguts des d'un panell d'administració.

de disseny i en funció de la definició d'aquest és possible que ja es disposi de la plantilla de la web amb HTML5; si no fos així, s'hauria de fer.

Llavors s'hi incorporen els diferents estils en CSS3 que s'hagin determinat al final de la fase anterior i es desenvolupa el codi dels elements interactius, generalment en JavaScript. També es crearan altres components necessaris per al projecte. Per exemple, un blog, zones de registre d'usuaris, una botiga *online* o altres tipus de components, segons els requeriments de la web.

Finalment, també cal comprovar la visualització del lloc web en diferents navegadors i dispositius mòbils.

### 3.1.4 Publicació

Una vegada comprovat que tot funciona correctament, s'ha de publicar el lloc web. Per a això cal pujar-lo al servidor on estigui allotjat el domini registrat per a aquest.

Tanmateix, usualment el nom del domini del lloc web ja s'ha registrat en la fase de planificació, i per al seu allotjament es pot optar per algun servei de *hosting* que es pot contractar a [ca.dinahosting.com](http://ca.dinahosting.com) o a [www.cdmon.com/ca](http://www.cdmon.com/ca).

Per tant, caldrà contractar un servei de *hosting*. Hi ha diverses empreses que l'ofereixen, com pot ser dinahosting.com, cdmon.com o dondominio.com.

Finalment, s'ha de revisar que el lloc web sigui visible als principals cercadors, i també pot resultar interessant instal·lar l'eina de seguiment estadístic Google Analytics per obtenir informes per fer-ne el seguiment.

#### Allotjament web, 'hosting'

Servei que ofereix al client un espai en uns servidors amb connexions permanents fiables per poder publicar-hi un lloc web i que estigui disponible a Internet. També inclou altres serveis com correu electrònic, bases de dades, etc.

### 3.1.5 Manteniment

Cal que entenguem el lloc web com una cosa “viva”, que segueix creixent i canviant al llarg del temps. Així que el client també ens pot demanar que efectuem el manteniment del seu lloc web, per així adaptar-lo a noves tecnologies o afegir-hi i modificar-ne funcionalitats.

En conseqüència, entenem que el procés de creació d'un lloc web no s'acaba mai, ja que sempre caldrà actualitzar-ne continguts i revisar que els seus components són els adients.

## 3.2 Elaboració d'un prototipus web

La **fase de disseny** d'un projecte web correspon a l'elaboració d'un esbós del que seria la interfície del web, que es pot dur a terme mitjançant l'elaboració d'un prototip.

Un **prototip** és una representació limitada d'un lloc web que permet a les parts (clients i desenvolupadors) explorar i provar-ne l'ús.

De manera general, els passos a seguir per fer un prototipus són:

1. Pas 1. Obtenir la inspiració.
2. Pas 2. Creació d'un *wireframe*.
3. Pas 3. Creació d'un *mockup*.
4. Pas 4. Elaborar el prototipus.

En funció del tipus de projecte, se'n pot obviar algun o insistir més en un altre. Cal destacar que el prototip final que s'obtindrà no necessàriament serà un disseny acabat, sinó un document de treball.

### 3.2.1 Pas 1: obtenir la inspiració

En la fase inicial de disseny del lloc web cal una tasca de “pluja d’idees” per part del dissenyador (i, si s’escau, del sol·licitant del lloc web) per tal d’esbrinar i planificar quins elements hi han d’aparèixer. Per això caldrà començar deixant via lliure a la imaginació i jugar amb els diferents elements de disseny que en poden formar part, com per exemple si han d’aparèixer menús i de quina manera, quina informació es vol que aparegui a la part central o la temàtica de les imatges que s’hi volen incloure.

D’aquesta manera, el procés de creació de l’espíritu inicial no té un treball conceptual ni tècnic; preval la creativitat, l’experiència, l’habilitat i, en moltes ocasions, els gustos subjectius del dissenyador.

En definitiva, l’espíritu ha de reflectir les idees generals sobre el projecte i, per tant, ha de poder respondre qüestions com:

- Quins són els objectius finals de la web?
- A quin públic va dirigida?
- Quin servei es vol oferir?
- Quins són els elements més característics i on es posaran? (Logos, informació de contacte...)
- On estarà la zona de navegació?
- Tindrà serveis de xarxes socials? Quines?
- Quines seran les àrees amb continguts?

L'elaboració de l'esbós no requereix necessàriament de cap eina informàtica, fins i tot s'acostuma a fer-lo a llapis i paper i s'hi plasmen les idees de com es vol que sigui el lloc web. Així doncs, és una primera representació de la distribució dels elements que ha de contenir. També s'hi poden incloure anotacions als laterals o petits comentaris amb una mica d'informació sobre el que es vol fer.

### 3.2.2 Pas 2: creació del 'wireframe'

El següent pas en la creació d'un lloc web és la realització formal amb eines informàtiques de l'esbós elaborat en la fase anterior, que s'anomena *wireframe*.

Un *wireframe* és una il·lustració de la interfície d'un document web que se centra en l'assignació de l'espai i la prioritació del contingut, així com les funcionalitats disponibles i els comportaments desitjats.

Així doncs, els *wireframes* ajuden, entre altres coses, a:

- Donar prioritat als continguts determinant la quantitat i la localització de l'espai que se'ls ha d'assignar.
- Determinar la funcionalitat prevista de la interfície de la web.
- Clarificar els espais i les formes de visualització de determinats tipus de continguts sobre la interfície d'usuari.
- Mostrar les connexions entre pàgines.

És per aquest motiu que els *wireframes* no tenen estil tipogràfic, color o imatges, ja que el seu principal objectiu és establir la funcionalitat, el comportament i la jerarquia dels continguts.

Així que primer caldrà estructurar la informació en caixes i jerarquitzar els continguts. Llavors caldrà establir l'ordre de la informació que es vol presentar als usuaris que accedeixin al lloc web.

D'aquesta manera, l'estructura bàsica més comú és dividir la pàgina en tres zones: capçalera, cos i peu. Així doncs, en primer lloc situarem els continguts de dalt a baix, i en segon lloc, d'esquerra a dreta.

En la figura 3.1 podeu veure una proposta de *wireframe* per a la pàgina principal d'una companyia d'aviació.

**FIGURA 3.1.** 'Wireframe' per a la pàgina d'entrada d'una companyia d'aviació

Gliffy / wireframe1, v6



Per fer aquest 'wireframe' s'ha utilitzat l'eina en línia Gliffy.

Existeixen multitud d'eines en línia gratuïtes o de pagament per fer *wireframes*, com poden ser:

- Gliffy: disponible a [www.gliffy.com](http://www.gliffy.com)
- Cacoo: disponible a [cacoo.com](http://cacoo.com)
- Mockingbird: disponible a [gomockingbird.com](http://gomockingbird.com)
- Lumzy: disponible a [www.lumzy.com](http://www.lumzy.com)

### 'Wireframes' adaptatius

Avui dia, l'accés a qualsevol lloc web es fan tant des d'un ordinador o un telèfon mòbil com des d'una tauleta, i per aquest motiu cal definir l'estructura i l'ordre dels continguts per a múltiples mides i resolucions de pantalla.

Molts autors apunten que per crear un *wireframe* adaptatiu convé començar pel telèfon mòbil, *mobile first*. D'aquesta manera es garanteix que el contingut pot servir en els dispositius més petits i després ja s'anirà ampliant el disseny progressivament.

Per desenvolupar un disseny *mobile first* cal deixar de banda algunes idees comunes que s'apliquen al disseny per a ordinadors, ja que en una pàgina web per a mòbil en realitat hi ha dues parts importants, la capçalera i el peu de pàgina, les quals hauran d'incloure les opcions de navegació més importants.

D'aquesta manera, el disseny per a mòbils sovint tindrà una sola columna, fet que també provoca una visualització lineal dels continguts i característiques del lloc web.

#### **'Mobile first'**

Consisteix a dissenyar el lloc web pensant primer en els dispositius mòbils, ja que són més petits. Així, per exemple, si es pensa en un disseny per a un mòbil resumirem l'article que volem explicar per així aconseguir-ne una versió optimitzada.

En la figura 3.2 podeu veure un exemple de *wireframe responsive* per a la pàgina principal d'una companyia d'aviació.

**FIGURA 3.2.** 'Wireframe' per a la pàgina d'entrada d'una companyia d'aviació



A l'esquerra veiem el 'wireframe' per a un ordinador i a la dreta el mateix per a un mòbil.

## Mapes de navegació

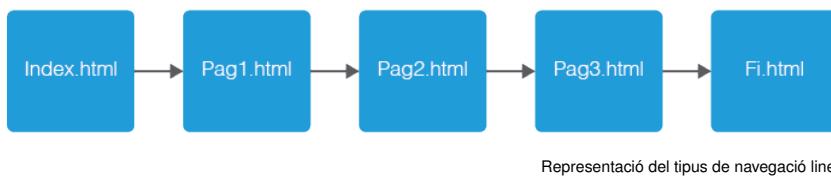
Una de les tasques principals en el moment de dissenyar un lloc web és l'elecció dels continguts i, per això, cal crear el que es diu un mapa del lloc, és a dir, una llista de pàgines web estructurades amb una arquitectura d'enllaços adequada per a una correcta navegació de l'usuari i un millor posicionament en els motors de cerca.

Als *wireframes* es defineixen els mapes de navegació que proporcionen una representació esquemàtica de les pàgines que formen el lloc web i mostren des de quina pàgina o pàgines es pot accedir a les altres.

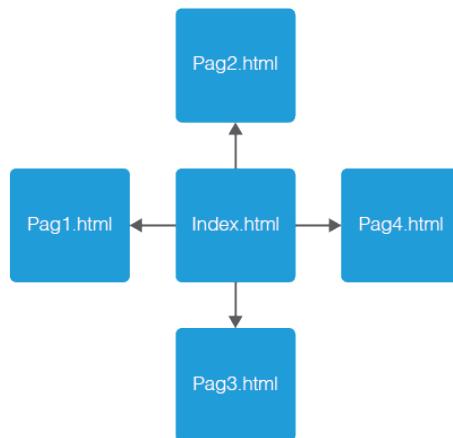
Existeixen sis tipus de mapes de navegació:

- **Navegació linial:** permet que l'usuari rebi la informació en l'ordre adequat, únicament amb les opcions d'anar endavant o endarrere. En la figura 3.3 podeu apreciar l'esquema que descriu aquest tipus de navegació.

**FIGURA 3.3.** Mapa de navegació linial

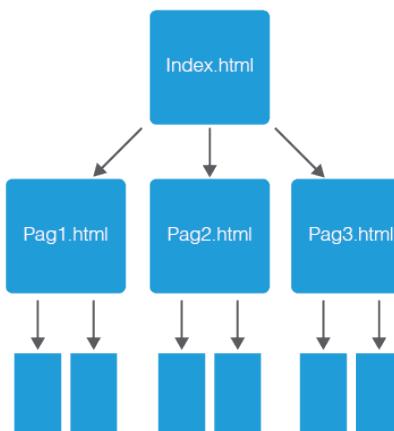


- **Navegació linial en estrella:** sistema de navegació similar al linial, però es va i es torna a l'inici. En la figura 3.4 podeu apreciar l'esquema que descriu aquest tipus de navegació.

**FIGURA 3.4.** Mapa de navegació en estrella

Representació del tipus de navegació en estrella

- **Navegació jeràrquica o d'arbre:** aquest sistema comença per una pàgina principal en la qual es presenten diverses opcions que permeten anar visualitzant pàgines específiques. En la figura 3.5 podeu apreciar l'esquema que descriu aquest tipus de navegació.

**FIGURA 3.5.** Mapa de navegació jeràrquica

Representació del tipus de navegació jeràrquica

- **Navegació no linial:** permet marcar un camí general de navegació, però dóna cabuda a petites variacions.
- **Navegació composta:** la navegació composta és la que inclou diferents sistemes de navegació.
- **Navegació múltiple:** aquesta estructura de navegació permet que cada pàgina estigui vinculada a totes les altres.

Els mapes de navegació sovint també s'incorporen com una funcionalitat més per a l'usuari de la pàgina web. Així, l'usuari disposa de tots els enllaços a qualsevol pàgina del lloc web des d'un mateix punt.

Vegeu en la figura 3.6 el mapa web de la Xarxa Telemàtica Educativa de Catalunya (XTEC).

**FIGURA 3.6.** Mapa web de la Xarxa Telemàtica Educativa de Catalunya (XTEC)

The screenshot shows a computer screen displaying the 'Mapa web' (Navigation menu) of the XTEC website. The menu is organized into several main categories:

- Inici**
- Recursos**: Includes sections like Ciències de la naturalesa (Astronomia, Medi ambient, Botànica, Zoologia), Biologia humana (Salut, Geologia, Fisica, Quimica), En general (Edumet), and Ciències socials.
- Formació**: Includes sections like Formació permanent del professorat (Línies de formació, Plans de formació de zona, Cercador d'activitats), Formació general del professorat (Acció tutorial, orientació educativa i clima de centre, Actualització científica i didàctica, Atenció a la diversitat i la inclusió, Capacitats, competències i currículum), and Escola, família i entorn.
- Serveis educatius**: Includes sections like Serveis que són (Adreces, Tot Catalunya, Serveis Territorials, Tipus de Serveis), Serveis educatius de zona (Centres de recursos pedagògics (CRP), Equips d'assessorament i orientació psicopedagògica (EAP), Equips d'assessorament en llengua i cohesió social (ELIC)), and Serveis educatius específics (Centres d'aprenentatge).

Font: <http://xtec.gencat.cat/ca/MapaWeb/>

### 3.2.3 Pas 3: creació del 'mockup'

Una vegada es té clara l'estructura dels elements que apareixeran a la interfície web cal fer el que s'anomena *mockup*, que servirà per demostrar i provar el disseny.

El **mockup** és una composició gràfica completa que utilitza el *wireframe* com a plantilla i introduceix tots els elements gràfics i visuals.

El *mockup* ha de ser un mitjà per representar l'aparença del producte i mostrar el fonaments de la seva funcionalitat, i inclou els detalls visuals com són els colors, la tipografia, les icones, les imatges, etc. En observar un *mockup* s'ha de tenir una idea de com es veurà el producte final i una idea aproximada de com funciona.

Podeu veure un exemple de *mockup* en la figura 3.7.

**FIGURA 3.7.** 'Mockup' per a la pàgina d'entrada d'una companyia d'aviació

The screenshot shows a wireframe tool interface for creating a user interface. On the left, there is a 'Stencils' panel containing various UI components like 'Common' (Heading 1, Heading 2, Paragraph, Click me, Radio, Checkbox, Text area), 'Inputs' (Text, Number, Date, Select, File, Color), and 'Buttons' (Submit, Button). The main workspace displays a wireframe of a registration form for a company named 'DESPEGA'. The form includes fields for 'First name', 'Last name', 'Origin' (with a dropdown menu 'Selecciona un origen'), 'Desti' (with a dropdown menu 'Selecciona un origen'), 'Data Sortida' (with a date picker set to '01/07/2016'), 'Data Tornada' (with a date picker set to '10/07/2016'), and a 'Buscar' (Search) button. To the right of the workspace, there are several panels: 'UPGRADE NOW', 'Styles' (with a preview window showing a light blue color), 'DEFAULT STYLES' (with settings for 'TEXT', 'FILLS & STROKES', and 'ROUNDED CORNERS'), and a 'TEXT' panel with font and size options. At the bottom of the workspace, there is a footer with links to 'Despega SA | Condicions d'ús del Ric | Accesibilitat | Mapa web'.

Per a la realització del 'mockup' s'ha utilitzat l'eina en línia [moqups.com](https://app.moqups.com/)

Existeixen moltes eines que permeten crear *mockups*. De fet, podem usar qualsevol programa de dibuix, ja sigui vectorial, com Illustrator o Inkscape, o programes més orientats al retoc d'imatges, com Fireworks o Photoshop.

Però també hi ha diverses eines específiques per a aquesta finalitat, algunes d'elles fins i tot *online*, que permeten fer el disseny de maquetació, com són:

- Balsamiq: ofereix una versió per descarregar i una versió en línia. Disponible a [balsamiq.com](http://balsamiq.com)
- Gomockingbird: eina en línia disponible a [gomockingbird.com](http://gomockingbird.com)
- Hotgloo: eina en línia disponible a [www.hotgloo.com](http://www.hotgloo.com)
- Invisionapp: eina en línia disponible a [www.invisionapp.com](http://www.invisionapp.com)
- Mockflow: eina en línia disponible a [mockflow.com](http://mockflow.com)
- Moqups.com: eina en línia disponible a [moqups.com](http://moqups.com)

### Interpretació de guies d'estil. Elements

Una de les formes d'assegurar-se que tot l'equip de projecte estigui en sintonia en el moment de dissenyar les diferents parts del lloc web és crear una documentació de disseny o una guia d'estils.

Una **guia d'estils** és una col·lecció d'elements predissenys i regles que els dissenyadors i desenvolupadors web han de seguir per assegurar-se que totes les parts del lloc web siguin consistentes i cohesionades.

Definir una guia d'estils és molt important quan diversos dissenyadors treballen conjuntament per tal d'assegurar-se que no s'interpretin malament els estils que s'han de seguir o que no es canviïn.

Per crear una guia d'estils caldrà definir la tipografia, la paleta de colors, les imatges i els botons.

**1. La tipografia:** consisteix a crear un llistat dels tipus de font que s'utilitzaran en el disseny web. A més, cal explicar les normes que regiran el seu ús. És a dir, establir els estils (mida, estil i color) de la jerarquia de capçaleres i del cos del text.

En la figura 3.8 hi veiem un exemple de definició de la tipografia que s'utilitzarà per desenvolupar el lloc web.

**FIGURA 3.8.** Definició de la tipografia del lloc web

**Títols nivell 1** Verdana 26px negreta

**Títols nivell 2** Verdana 24px normal

Cos del text Verdana 14px normal

Aquesta informació formaria part de la guia d'estils.

**2. La paleta de colors:** consisteix a mostrar els colors principals del lloc web. És important incloure el codi de color exacte i informació rellevant per a la seva aplicació.

En la figura 3.9 hi veiem un exemple de definició dels colors que s'utilitzaran per desenvolupar el lloc web.

**FIGURA 3.9.** Definició dels colors del lloc web

Base      fosc 10%      fosc 15%      clar 10%      clar 15%

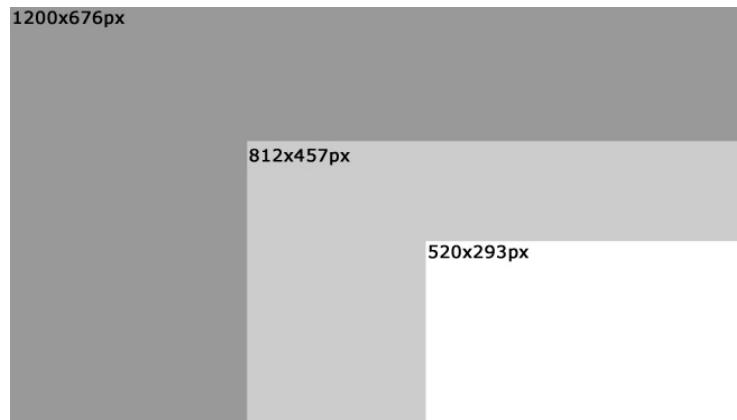


#EDEDED

Aquesta informació formaria part de la guia d'estils.

**3. Les imatges:** consisteix a indicar totes les mides i proporcions de les imatges que s'han d'usar al lloc web. També cal incloure els efectes o variacions que han de tenir si es tracta d'una imatge destacada o si apareix en el peu de pàgina o en altres seccions.

En la figura 3.10 hi veiem un exemple de definició dels colors que s'utilitzaran per desenvolupar el lloc web.

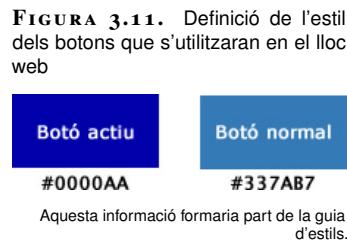
**FIGURA 3.10.** Definició de la mida de les imatges que s'utilitzaran en el lloc web

Aquesta informació formaria part de la guia d'estils.

**4. Botons i elements de navegació:** consisteix a definir l'ús combinat de la tipografia i la paleta de colors. Ha de mostrar com es veuran els botons i enllaços en estat de *hover*, selecció i inactiu. També s'ha de descriure el disseny dels

elements de navegació, com són fletxes de desplaçament, menús i submenús, així com el missatge per informar que un enllaç no funciona.

En la figura 3.11 hi veiem un exemple de definició de l'estil dels botons que s'utilitzaran per desenvolupar el lloc web.



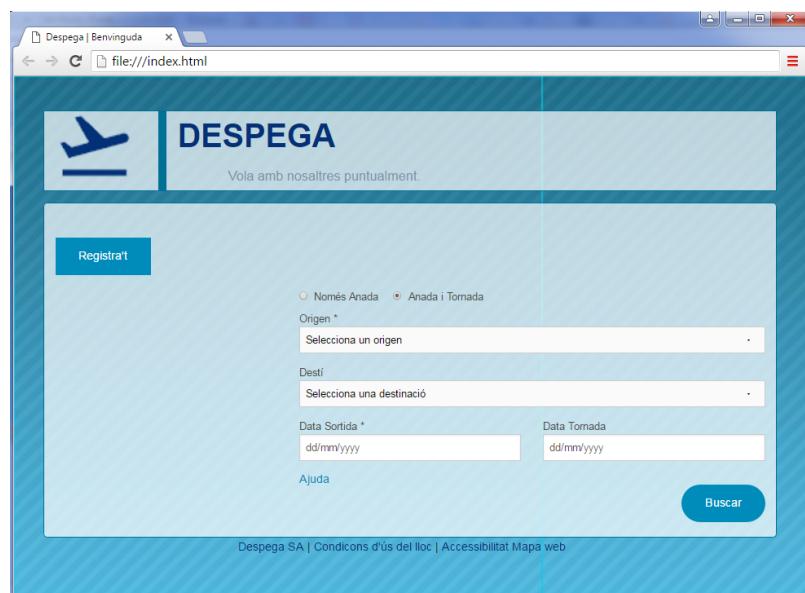
### 3.2.4 Pas 4: elaboració del prototip

El prototip es duu a terme per avaluar la navegació i la interacció amb l'usuari, i serveix per definir aspectes que no han quedat reflectits en el *wireframe* o *mockup*.

Un **prototip** és una representació amb tots els detalls de la interfície i ens serveix com a model de comportament del sistema.

D'aquesta manera, els prototips són navegables i serveixen per testejar elements d'interacció, com la validació dels formularis, les icones, els botons o qualsevol element amb què l'usuari final interactuï. Així doncs, a través de la seva creació es poden identificar i solucionar problemes sobre l'experiència d'usuari (UX) (vegeu la figura 3.12).

**FIGURA 3.12.** Visualització del prototip al navegador



Pàgina d'inici del prototip

En definitiva, un prototip és una primera implementació amb HTML i CSS basada en el *mockup* realitzat, en el qual es poden visualitzar tots els elements encara que alguns no funcionin. Per exemple, en un lloc web que ofereixi un formulari per donar-se d'alta al sistema o per validar-s'hi, el prototip ha de mostrar el formulari encara que aquest no s'envii a cap lloc o no estigui implementada la funcionalitat que ha de recollir les dades.

### 3.3 Plantilles web

L'elaboració d'un prototip i la següent fase de desenvolupament es pot dur a terme mitjançant el que s'anomena una plantilla web. Amb la seva utilització s'agilitza el procés de creació de la interfície oferint, si més no, un punt de partida o una idea aproximada del disseny que es vol aconseguir.

Una **plantilla web** és un conjunt de documents web predissenys que es poden utilitzar per inserir contingut i imatges per a així desenvolupar la interfície del lloc web.

El principal avantatge de la utilització d'una plantilla és que permet reduir considerablement el temps de desenvolupament i, en conseqüència, el seu cost. Tot i això, cal avaluar si la seva utilització és apropiada, ja que en ocasions no permeten un disseny i maquetació del lloc personalitzat. Per a això, una vegada més, l'opció a triar per desenvolupar un lloc web estarà força condicionat no tan sols per les eines a fer servir, sinó per l'objectiu de la web i l'usuari final a qui va dirigida.

#### 3.3.1 Criteris per triar una plantilla

A l'hora d'escollar una plantilla per al nostre lloc web és important tenir en compte algunes consideracions:

- El disseny ha de **reflectir el que s'està intentant comunicar**, és a dir, l'ànima del tipus de projecte o negoci que representi, ja que no és el mateix si el lloc web s'ha de destinar a una pàgina corporativa, una botiga virtual, un blog o una galeria d'imatges. Per tant, cal comprovar que el disseny de la plantilla s'hi adiqui.
- El disseny de la plantilla ha de ser **adaptable** a diferents tipus de dispositius.
- Cal comprovar que estigui **optimitzada per al posicionament SEO**. Si més no, que no tingui algun error que impedeixi aparèixer als primers resultats dels cercadors, com poden ser menús duplicats, etiquetes HTML inventades, etc.

- Depenen del públic al qual es vulgui destinar el lloc web, pot ser interessant que la plantilla ofereixi **suport per a diferents idiomes**.
- També cal assegurar-nos que la plantilla sigui **compatible amb la majoria de navegadors**, perquè tothom pugui visitar el nostre lloc web.
- Cal no descuidar la usabilitat de la plantilla, el seu temps de càrrega i la facilitat a l'hora de modificar-la.

Cal remarcar que algunes plantilles són gratuïtes i d'altres de pagament.

Per crear una plantilla es disposa de diverses eines informàtiques. Es poden obtenir i descarregar plantilles de múltiples llocs web a:

- [html5up.net](http://html5up.net)
- [templated.co](http://templated.co)
- [www.zerotheme.com](http://www.zerotheme.com)

També es poden obtenir plantilles per treballar amb els diferents *frameworks*, com per exemple Bootstrap, a:

- [startbootstrap.com](http://startbootstrap.com)
- [wrapbootstrap.com/tag/seo](http://wrapbootstrap.com/tag/seo)
- [themesseo.com/cat/bootstrap](http://themesseo.com/cat/bootstrap)

### 3.3.2 Com utilitzar una plantilla

Una vegada descarregada la plantilla, es disposarà d'una carpeta amb diferents recursos necessaris per utilitzar-la. Generalment hi trobem:

- Una pàgina anomenada index.html, en la qual es veu l'esquema de disseny de la pàgina d'entrada al lloc web. Depenen de la plantilla, també s'hi poden trobar altres pàgines HTML que proporcionaran altres dissenys per a les pàgines considerades com a secundàries.
- Carpetes amb els fulls d'estils CSS, arxius JavaScript i imatges, així com altres recursos per al *front-end*.

Amb la utilització d'una plantilla no es necessita pensar en el disseny, ja que ja està fet, però sí que cal personalitzar-la, en la mesura que es pugui, perquè encaixi amb les necessitats del lloc web.

Per tant, el que cal fer és:

1. Localitzar els arxius que inclou la plantilla. És a dir, els fitxers HTML, CSS i els gràfics.

L'inspector del navegador web és un eina que inclouen els principals navegadors i que permet examinar i modificar l'estructura i el disseny d'una pàgina web.

2. Examinar l'estructura de la plantilla amb una eina d'edició web com pot ser Brackets i buscar les parts que es desitgin modificar. Per facilitar la localització dels elements es pot obrir la plantilla amb el navegador web i utilitzar l'inspector.
3. Editar les àrees de text amb l'editor Brackets i afegir el nou contingut.
4. Personalitzar les imatges de la plantilla. Cal assegurar que tinguin la mateixa mida que les originals; si no, la disposició del disseny es pot veure deformada.
5. Treure qualsevol part de la plantilla no desitjada. No obstant això, cal fixar-se si en treure aquests ítems es distorsiona el disseny. Si fos el cas, caldria reemplaçar l'ítem eliminat per una imatge o text de la mateixa mida per mantenir l'ample de la columna.

Tot i que no és recomanable modificar els fulls d'estil CSS, potser ens interessa modificar-ne algun aspecte, com pot ser el color de fons, del text, etc.

### 3.3.3 Cas pràctic: creació d'un prototip a partir d'una plantilla web

En aquest cas pràctic es crearà un prototip basat en una plantilla de la pàgina principal d'una empresa de desenvolupament de *software*.

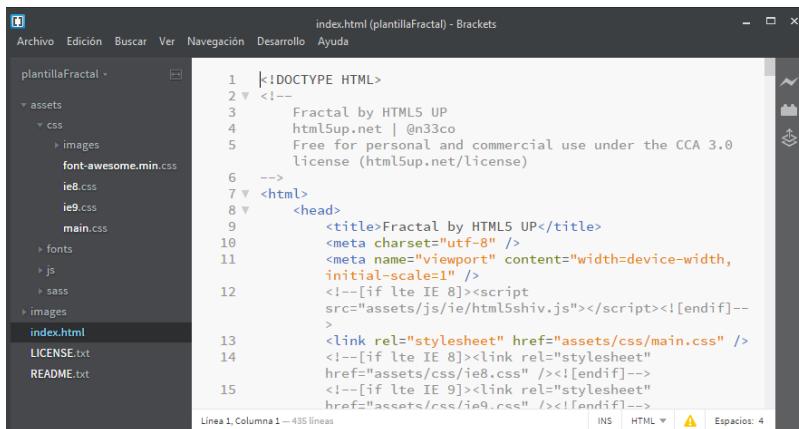
Primer de tot heu de descarregar la plantilla que heu triat per al vostre lloc web. En aquest cas es farà servir la que obtindreu a [html5up.net/fractal](http://html5up.net/fractal).

Una vegada descarregada cal que la descomprimiu i que examineu els fitxers que inclou. És important localitzar-ne el fitxer index.html i comprovar on es localitzen les imatges i altres recursos de la plantilla, com són els fitxers CSS, main.css.

Per a la realització d'aquest cas pràctic s'ha utilitzat l'editor Brackets, però podeu fer servir qualsevol altre editor que permeti la creació de documents HTML.

Vegeu a la figura 3.13 els documents de la plantilla oberts en l'editor Brackets.

**FIGURA 3.13.** Visualització dels documents de la plantilla



```

index.html (plantillaFractal) - Brackets
Archivo Edición Buscar Ver Navegación Desarrollo Ayuda
plantillaFractal
  assets
    css
      font-awesome.min.css
    ie8.css
    ie9.css
    main.css
  fonts
  js
  sass
  images
  index.html
  LICENSE.txt
  README.txt

1  <!DOCTYPE HTML>
2  <!--
3    Fractal by HTML5 UP
4    html5up.net | @n33co
5    Free for personal and commercial use under the CCA 3.0
       license (html5up.net/license)
6  -->
7  <html>
8    <head>
9      <title>Fractal by HTML5 UP</title>
10     <meta charset="utf-8" />
11     <meta name="viewport" content="width=device-width,
           initial-scale=1" />
12     <!--[if lte IE 8]><script
           src="assets/js/ie/html5shiv.js"></script><![endif]-->
13     <link rel="stylesheet" href="assets/css/main.css" />
14     <!--[if lte IE 8]><link rel="stylesheet"
           href="assets/css/ie8.css" /><![endif]-->
15     <!--[if lte IE 9]><link rel="stylesheet"
           href="assets/css/ie9.css" /><![endif]-->

```

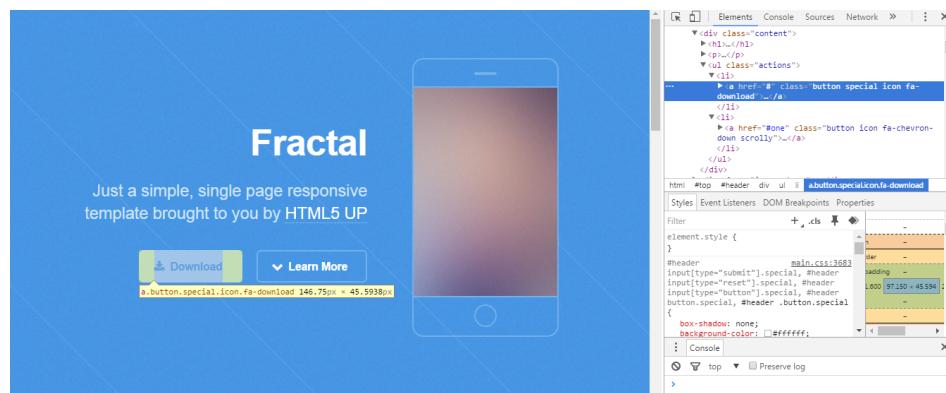
Línia 1, Columna 1 — 435 línies

INS HTML Espacios: 4

S'ha utilitzat l'editor Brackets.

Seguidament, fareu una còpia de la plantilla per no perdre els fitxers originals i obrireu el fitxer index.html amb el navegador. Amb l'ajuda de l'inspector (botó dret del ratolí damunt de l'element i seleccionar inspector) localitzareu el codi HTML i CSS referent a cada element que vulgueu modificar. Vegeu aquesta informació en la figura 3.14.

**FIGURA 3.14.** Visualització de la plantilla amb el navegador



L'inspector es veu a la dreta, a la part de dalt hi veiem el codi HTML, i a sota el CSS referent a l'element HTML seleccionat.

En primer lloc modificareu la capçalera per tal que quedí com la de la figura 3.15.

**FIGURA 3.15.** Aparència de la capçalera de la pàgina



Per fer aquesta pàgina s'ha modificat una plantilla HTML5 i CSS3.

Així doncs, amb l'editor Brackets cal que modifiqueu el text de la capçalera i que tragieu els elements que no siguin necessaris. El codi ha de quedar com el següent:

#### Codi HTML per a la capçalera

```

1 <header id="header">
2   <div class="content">
3     <h1>IOC Disseny</h1>
4     <p>Ajudem a que la vostra idea es converteixi<br />
5       en una realitat</p>
6     <ul class="actions">
7       <li><a href="#one" class="button icon fa-chevron-down scrollly">M

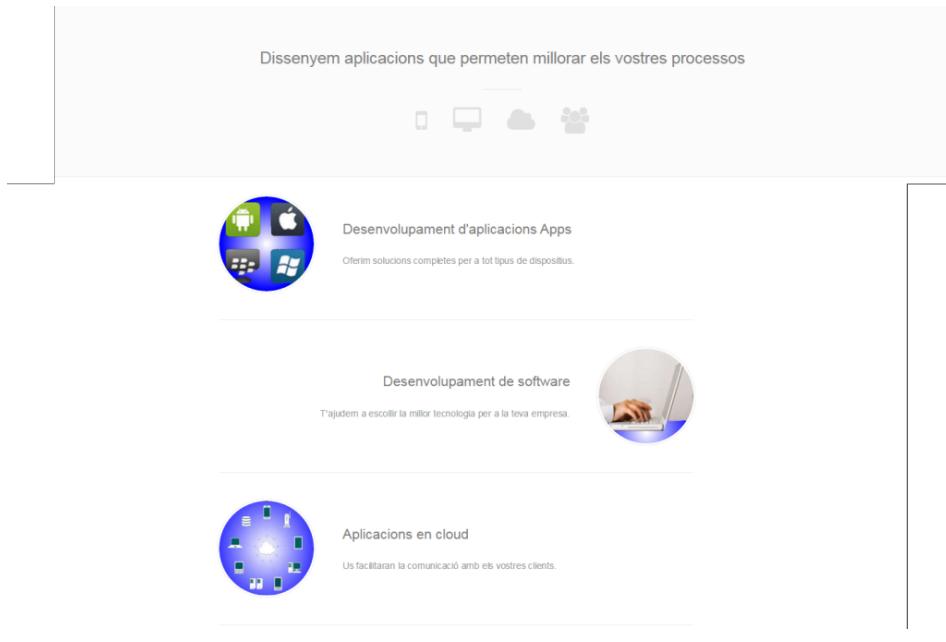
```

```

8     és informació</a></li>
9   </ul>
10  </div>
11  <div class="image phone"><div class="inner"></div>
13  </div>
14 </header>
```

En segon lloc canviareu el cos de la plantilla. Per tant, modificant la secció *One* per tal que quedi com a la figura 3.16. És important que en modificar la imatge aquesta tingui la mateixa mida que la imatge original de la plantilla.

**FIGURA 3.16.** Aparència del cos de la pàgina



Les icones de la secció *One* són enllaços a seccions posteriors, així que modificant també la secció *Two* i posareu un identificador a cada *section* per tal que els enllaços anteriors funcionin, i també modificant el text. També es poden modificar els fitxers de les imatges que presenta la plantilla, però recordeu que la mida de la imatge ha de ser la mateixa que l'original. El codi ha de quedar com el següent:

#### Codi HTML per a les seccions

```

1  <!-- One -->
2  <section id="one" class="wrapper style2 special">
3    <header class="major">
4      <h2>Dissenyem aplicacions que permeten millorar els vostres
         processos</h2>
5    </header>
6    <ul class="icons major">
7      <li><a href="#two" class="icon fa-mobile scrollly"><span class=
       "label">Mobile</span></a></li>
8      <li><a href="#two2" class="icon fa-desktop scrollly"><span class=
       "label">Desktop</span></a></li>
9      <li><a href="#two3" class="icon fa-cloud scrollly"><span class=
       "label">Cloud</span></a></li>
10     <li><a href="#two4" class="icon fa-users scrollly"><span class=
        "label">Users</span></a></li>
11   </ul>
```

```

12   </section>
13
14  <!-- Two -->
15  <section id="two" class="wrapper">
16    <div class="inner alt">
17      <section id="two1" class="spotlight">
18        <div class="image"></div>
19        <div class="content">
20          <h3>Desenvolupament d'aplicacions Apps</h3>
21          <p>Oferim solucions completes per a tot tipus de dispositius.</p>
22        </div>
23      </section>
24      <section id="two2" class="spotlight">
25        <div class="image"></div>
26        <div class="content">
27          <h3>Desenvolupament de software</h3>
28          <p>T'ajudem a escollir la millor tecnologia per a la teva
29            empresa.</p>
30        </div>
31      </section>
32      <section id="two3" class="spotlight">
33        <div class="image"></div>
34        <div class="content">
35          <h3>Aplicacions en cloud</h3>
36          <p>Us facilitarem la comunicació amb els vostres clients.</p>
37        </div>
38      </section>
39      <section id="two4" class="special">
40        <ul class="icons labeled">
41          <h3>La tranquilitat de confiar en un equip de
42            professionals</h3>
43          <li><span class="icon fa-user"><span class="label">Josep Rabentó
44            s</span></span></li>
45          <li><span class="icon fa-user"><span class="label">Sandra Maca</
46            span></span></li>
47          <li><span class="icon fa-user"><span class="label">Gerard Pinyol
48            </span></span></li>
49          <li><span class="icon fa-user"><span class="label">Adrià Marco</
          span></span></li>
          <li><span class="icon fa-user"><span class="label">Antoni
            Guillemat</span></span></li>
        </ul>
      </section>
    </div>
  </section>

```

En tercer lloc, inserireu un formulari de contacte a la secció 3 de la plantilla.

A la secció 4 de la plantilla, que està comentada i per tant no es visualitza, en mostrar la plantilla hi podeu trobar exemples d'utilització d'alguns elements que podeu utilitzar, així com també com inserir un formulari.

Fareu les modificacions necessàries per tal que es vegi com la figura 3.17, i en acabar podeu esborrar tota la secció 4.

**FIGURA 3.17.** Aparència del formulari de contacte

Per fer aquesta pàgina s'ha modificat una plantilla HTML5 i CSS3.

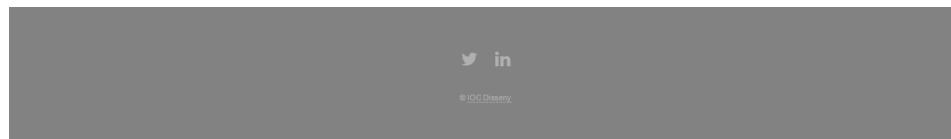
El codi resultant del formulari de contacte ha de ser com el següent:

#### Codi HTML per al formulari de contacte

```

1 <section id="three" class="wrapper style2 special">
2   <header class="major">
3     <h2>Contacta amb nosaltres</h2>
4   </header>
5   <ul class="actions">
6     <form method="post" action="#">
7       <div class="row uniform">
8         <div class="3u"></div>
9         <div class="6u$ 12u$(xsmall)">
10           <label class="align-left">Nom:</label>
11           <input type="text" name="demo_name" id="demo_name" value="">
12           <label class="align-left">placeholder="Nom complet" /></label>
13         </div>
14         <div class="3u"></div>
15         <div class="6u$ 12u$(xsmall)">
16           <label class="align-left">Correu electrònic:</label>
17           <input type="email" name="demo_email" id="demo_email" value="">
18           <label class="align-left">placeholder="El teu correu electrònic" /></label>
19         </div>
20         <div class="3u"></div>
21         <div class="6u$ 12u$(xsmall)">
22           <label class="align-left">Comentari:</label>
23           <textarea name="demo_message" id="demo_message" rows="6"></textarea></label>
24         </div>
25       <div class="3u"></div>
26       <div class="6u$ 12u$(xsmall)">
27         <ul class="actions">
28           <li><input type="reset" value="Esborra" class="small" /></li>
29           <li><input type="submit" value="Envia" class="special small" /></li>
30         </ul>
31       </div>
32     </form>
33   </ul>
</section>
```

En quart lloc modificareu el peu de pàgina, i el resultat hauria de ser com la següent figura 3.18.

**FIGURA 3.18.** Aparença del peu de pàgina

Per fer aquesta pàgina s'ha modificat una plantilla HTML5 i CSS3.

Encara que no és recomanable modificar gaire la plantilla, a vegades és necessari. Així doncs, per tal de canviar el color de fons del peu de pàgina podeu modificar l'atribut `background-color: #828282` de l'estil `#footer` que trobeu al fitxer `main.css`.

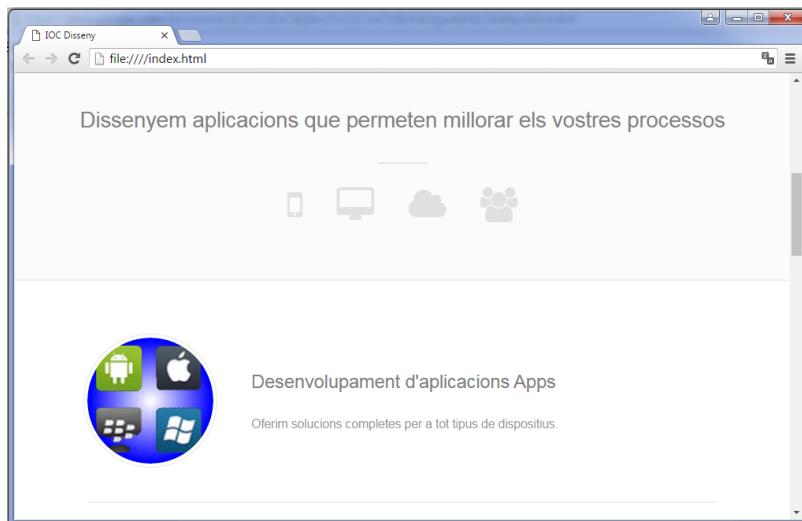
El codi resultant ha de ser com el següent:

#### Codi HTML per al peu de pàgina

```

1 <footer id="footer">
2   <ul class="icons">
3     <li><a href="#" class="icon fa-twitter"><span class="label">
4       Twitter</span></a></li>
5     <li><a href="#" class="icon fa-linkedin"><span class="label">
6       Instagram</span></a></li>
7   </ul>
8   <p class="copyright">&copy; <a href="#">IOC Disseny</a></p>
9 </footer>
```

Finalment, heu de comprovar que la pàgina es visualitza correctament en el navegador, tal com es veu en la figura 3.19.

**FIGURA 3.19.** Visualització de part del lloc web en el navegador

Aquesta pàgina té un 'scroll' molt llarg, i s'ha de comprovar que tots els elements es veuen i tenen el comportament desitjat.

## 3.4 Sistema de gestió de continguts

Un lloc web pot estar creat amb un gestor de continguts en la seva totalitat, o bé aquest pot ser un component més del lloc. És a dir, en un lloc web, amb la seva

estructura i contingut, es pot incloure un component que gestioni un fòrum, un blog, una botiga virtual, etc., i que aquest estigui gestionat a través d'un CMS concret.

**Un sistema de gestió de continguts**, també conegut amb les sigles CMS, és una aplicació web que proporciona una estructura de suport (*framework*) per a la creació i administració de contingut web, tant per part dels administradors com dels editors, com d'altres participants i usuaris.

Els gestors de continguts es poden classificar segons el seu ús i funcionalitat:

- Blogs: per a pàgines personals amb estructura cronològica que s'actualitza regularment.
- Fòrums: per compartir opinions i fer preguntes i respostes.
- *Wikis*: per a desenvolupament col·laboratiu.
- Ensenyament virtual: plataforma per a continguts d'ensenyament *online*.
- Comerç electrònic: plataforma de gestió d'usuaris, catàleg, compres i pagaments.
- Altres, difusió de contingut multimèdia, plataformes de gestió d'aplicacions mòbils, etc.

Alguns dels gestors de continguts més utilitzats avui dia són Wordpress i Joomla.

### 3.4.1 WordPress

És un sistema de gestió de continguts enfocat a la creació de qualsevol tipus de lloc web, encara que és molt usat per a la creació de blogs. Està desenvolupat en PHP i MySQL. Algunes de les seves funcionalitats més importants són:

- Capacitat de crear pàgines estàtiques.
- Permet ordenar articles en categories, subcategories i etiquetes.
- Permet distribuir articles mitjançant RDF, RSS i Atom.
- Admet complements, plantilles i *widgets*.

El trobem disponible a [wordpress.org](http://wordpress.org).



Logotip de WordPress

### 3.4.2 Joomla!

#### MVC

Patró de desenvolupament de programari que divideix l'aplicació en tres parts interconnectades: el model de dades, la interfície usuari i la lògica de control.

És un sistema de gestió de continguts lliure i de codi obert per publicar continguts al World Wide Web i en intranets, i alhora és un *framework* de desenvolupament d'aplicacions web de tipus Model-Vista-Controlador (MVC). Aquest sistema proporciona una sèrie de funcionalitats:

#### RSS, Really Simple Syndication

Família de formats de canals web XML utilitzats per publicar continguts actualitzats sovint, a través de les quals es pot compartir la informació i usar-la en altres llocs web o programes.

- Memòria cau de pàgines per millorar el rendiment.
- Sindicació RSS.
- Versions imprimibles o en PDF de les pàgines.
- Flaixos de notícies.
- Blogs.
- Enquestes d'opinió.
- Cerca en el web.
- Internacionalització de l'idioma.
- És escrit en llenguatge de programació PHP.
- Utilitza el sistema de bases de dades MySQL o PostgreSQL per emmagatzemar la informació.



Logotip de Joomla!

El trobem disponible a [www.joomla.com](http://www.joomla.com).

### 3.4.3 CMS per a comerç 'online'

#### Comerç electrònic, 'e-commerce'

Fa referència a una transacció comercial en què les parts involucrades interaccionen de manera electrònica en lloc de fer-ho de la manera tradicional.

Existeixen diferents gestors de continguts específics per a la creació de botigues virtuals, tots ells permeten crear un catàleg amb articles i seleccionar-ne diversos per comprar-los o veure els costos d'enviament, taxes i qualsevol altre cost per avançat.

Alguns exemples de CMS específics per a comerç *online* són:

- **Magento:** una molt flexible i escalable que permet personalitzar completamente el lloc web i compta amb més de 5.000 extensions a MagentoConnect. El seu ús és professional i el trobem disponible a [magento.com](http://magento.com).
- **Virtuemart:** és un component de Joomla!, i per tant no es pot utilitzar sense aquest gestor de continguts. Està pensat per ser utilitzat en negocis petit o mitjans. El trobem disponible a [virtuemart.net](http://virtuemart.net).

- **PrestaShop:** desenvolupat exclusivament amb PHP5, és molt lleuger i ràpid. El trobem disponible a [www.prestashop.com/es](http://www.prestashop.com/es).
- **OsCommerce:** desenvolupat sota llicència GPL, el trobem disponible a [www.oscommerce.com](http://www.oscommerce.com).



Aquests gestors de contingut ofereixen un *front-end* configurable i també *back-end* que ens permet administrar el catàleg, l'estoc, els preus, les comandes, els usuaris, etc.

Existeixen diverses maneres per fer el pagament dels articles seleccionats. Els CMS solen incorporar diversos mètodes de pagament, i per tant només caldrà activar els que ens interessin. Però si el nostre lloc web està fet a mida caldrà que implementem les diferents maneres de pagament que vulguem oferir als nostres clients.

Els mètodes de pagament més usuals són:

- **Passarel·la de pagament o TPV Virtual:** per implantar un TPV Virtual en el nostre lloc web cal que ho sol·licitem al banc amb el qual treballem. Llavors aquest ens envia la informació necessària sobre com enviar-li correctament totes les dades corresponents a la transacció bancària.
- **Transferència bancària:** és un altre mètode de pagament, encara que més lent. Cal que el client doni l'ordre de pagament a un número de compte de la botiga *online*, i també cal que indiqui el DNI i el número de comanda perquè el responsable de la botiga *online* pugui verificar el pagament.
- **Transferència online:** existeixen diverses empreses que proporcionen aquest servei de pagament en línia. Una de les empreses més conegudes que ofereixen aquest servei és Paypal. Per acceptar pagaments per aquesta passarel·la cal obrir un compte per a empreses a [www.paypal.com/es/home](http://www.paypal.com/es/home).



Logotip de Paypal

### 3.5 Altres components d'un lloc web

En el nostre lloc web hi podem incloure altres components per poder oferir algunes funcionalitats concretes. Aquests es poden classificar entre els que només tenen *front-end* i els que també tenen *back-end*.

En una aplicació web, el **front-end** és la part que interactua amb l'usuari; així doncs, és el responsable de mostrar la interfície web i d'agafar les dades d'entrada de l'usuari per facilitar-les al **back-end**, és a dir, a la part de l'aplicació web que s'executa al servidor. Aquest les processa, i generalment retorna una resposta que rep de nou el *front-end* per així presentar el resultat a l'usuari.

#### Aplicació web

Són aquelles eines que els usuaris poden utilitzar accedint a un servidor web a través del seu navegador. És a dir, seria el programari que, almenys la seva interfície gràfica, l'executa el navegador.

El **front-end** fa referència a totes aquelles tecnologies que corren al costat del client, és a dir, al navegador web. És la part del desenvolupament web que s'encarrega de la part frontal d'un lloc web; en definitiva, de la presentació del disseny del lloc web.

Generalitzant, els llenguatges que s'utilitzen el *front-end* són HTML, CSS i JavaScript, entre d'altres.

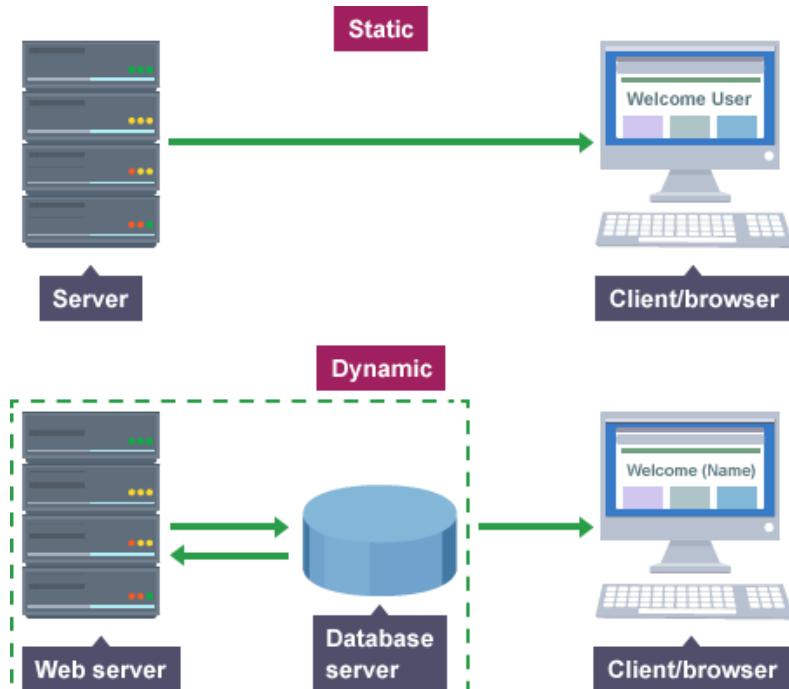
El **back-end** fa referència al costat del servidor que no és visible a l'usuari, és a dir a l'accés i la manipulació de les dades.

Els llenguatges que es poden utilitzar per desenvolupar el *back-end* són PHP, Python, .NET i Java.

D'aquesta manera, podem distingir entre llocs web estàtics i dinàmics. Els primers no tenen la capacitat d'interactuar amb l'usuari, ja que no tenen *back-end* i tot el que s'hi pot fer és modificar l'aparença. En canvi, els dinàmics són capaços de mostrar uns continguts o uns altres en funció de la resposta que proporcioni el *back-end* de l'aplicació web.

En la figura 3.20 podem veure el flux d'informació que es produeix en una pàgina web estàtica i dinàmica.

**FIGURA 3.20.** Flux d'informació en una pàgina web estàtica i dinàmica



En aquests materials ens centrarem en els aspectes del *front-end*, encara que hem de recordar que el *front-end* sovint es recolza en elements proporcionats pel *back-end*.

### 3.5.1 Lloc de cerca

Un motor de cerca web és un programari que està dissenyat per cercar informació en el World Wide Web.

#### Buscador google

```

1 <form method="get" action="http://www.google.com/search">
2 <table>
3   <tr><td>
4     <a href=" http://www.google.com/">
5       </A>
7       <input type="text" name="q" size="31" maxlength="255"
8           value="">
9       <input type="hidden" name="hl" value="es">
10      <input type="submit" name="btnG" value="Cerca a Google">
11    </td></tr>
12  </table>
13 </form>
```

Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-dawm09/pen/adaqaa](https://codepen.io/ioc-dawm09/pen/adaqaa).

També ens pot interessar incloure un camp de cerqui patrons de text en el nostre lloc web, enllac de fer-ho al World Wide Web. Per a això cal que modifiquem el formulari anterior de la següent manera:

---

Podeu consultar més informació sobre com incloure un formulari de cerca a: [goo.gl/PPoHqX](http://goo.gl/PPoHqX)

```

1 <form method="get" action="http://www.google.com/search">
2 <table>
3   <tr><td>
4     <a href=" http://www.google.com/">
5       </A>
7       <input type="text" name="q" size="31" maxlength="255"
8           value="">
9       <input type="hidden" name="hl" value="es">
10      <input type="submit" name="btnG" value="Cerca a Google">
11      <input type="hidden" name="domains" value="ioc.xtec.cat"><br>
12        <input type="radio" name="sitesearch" value=""> WWW <
13        <input type="radio" name="sitesearch" value="ioc.xtec.cat"
14            checked> ioc.xtec.cat
15    </td></tr>
16  </table>
17 </form>
```

Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-dawm09/pen/yexKox](https://codepen.io/ioc-dawm09/pen/yexKox).

### 3.5.2 Espai de Xat

En funció del caire del nostre lloc web, ens pot resultar interessant incloure un component per poder xatejar. Per exemple, es poden establir horaris de serveis d'atenció al client a través d'un xat.

Per incloure aquesta funcionalitat hem d'avaluar l'arquitectura que hem seguit a l'hora de desenvolupar el nostre web; és a dir, si el nostre lloc web es basa en un CMS concret, probablement existeixi un *plug-in* que permeti afegir aquesta funcionalitat.

Si no és el cas i el nostre lloc web no es basa en cap CMS, també podem optar per incloure un component xat ja desenvolupat. Com sol passar en aquestes situacions, hem de decidir si volem una solució de codi obert o una altra de pagament. Alguns exemples serien:

- Chatango ([www.chatango.com](http://www.chatango.com)): bastant complet, també hi ha una part d'administració on es poden donar permisos als usuaris.
- ZopimChat ([www.zopim.com](http://www.zopim.com)): permet conversar en temps real amb els visitants del teu lloc web.
- Smartsupp ([www.smartsupp.com/es](http://www.smartsupp.com/es)): permet conversar amb els clients d'un lloc. És compatible amb Wordpress, Prestashop, Joomla i Magento.

### 3.5.3 Formulari de contacte

Sovint, en el nostre lloc web es vol donar la possibilitat que els usuaris es puguin posar en contacte amb nosaltres sense exposar la nostra adreça de correu electrònic.

Per a això es crearà un formulari de contacte que generalment tindrà els següents camps:

- Nom de l'usuari, per poder dirigir-nos-hi.
- Adreça de correu, per saber on enviar la resposta.
- Comentari on l'usuari exposarà el seu comentari o pregunta.

#### Formulari de contacte

```

1  <h3>Contacteu amb nosaltres: </h3>
2  <form>
3      <input type="text" name="nom" id="nom" placeholder="Escriu aquí
4          el teu nom"/>
5      <br />
6      <input type="email" name="email" id="id" placeholder="Escriu
7          aquí el teu correu electrònic" />
8      <br />
9      Comentari:
10     <textarea name="comentari" id="comentari" cols="15" rows="5"></
11         textarea>
12
13
14     <input type="reset" value="Esborrar" />
15     <input type="submit" value="Enviar" />
16
17 </form>
```

Podeu provar l'anterior exemple en el següent enllaç: [codepen.io/ioc-dawm09/pen/dGqqJV?editors=1000](http://codepen.io/ioc-dawm09/pen/dGqqJV?editors=1000).

### 3.5.4 Newsletter

Les *newsletters* fan referència a una publicació digital informativa que conté articles d'interès sobre la marca. Es distribueix a través del correu electrònic amb una periodicitat (diària, setmanal, mensual, trimestral...). Els subscriptors han mostrat interès per la marca i han sol·licitat rebre aquesta informació.

No s'ha de confondre les *newsletters* amb altres tècniques que també es distribueixen a través del correu electrònic, com són el *mailing* o l'*e-mail-marketing*.

Hi ha diversos motius pels quals pot resultar interessant incloure un component per a la subscripció a *newsletters* en el nostre lloc web, ja que fa que els subscriptors estiguin informats de les últimes notícies i promocions. I així, encara que no sigui una eina de venda directa, es pot aconseguir influir en els subscriptors de manera indirecta.

#### 'Mailing'

Es refereix a un format més de publicitat utilitzat per a la promoció de productes o serveis d'una marca. Se sol enviar a bases de dades massives, no pròpies de la marca.

#### 'e-mail-marketing'

Tècnica utilitzada per les marques per contactar amb el seu públic objectiu a través del correu electrònic. Aquesta tècnica de màrqueting inclou *newsletters* i *mailing*.

### 3.5.5 Bases de dades en línia

Una base de dades en línia, en anglès *online database*, és una base de dades accessible des d'Internet. Així es diferencien de les bases de dades convencionals que gestionen les dades a través d'un navegador web.

La majoria d'aquests serveis de bases de dades ofereixen interfícies web que permeten que l'usuari final pugi configurar les instàncies de les bases de dades.

Algunes de les característiques més importants de les bases de dades en línia són l'escalabilitat i l'alta disponibilitat que ofereixen. Cal tenir en compte que en molts casos acostumen a ser un servei de pagament.

Alguns exemples serien:

- Amazon Relational Database Service ([aws.amazon.com/rds](http://aws.amazon.com/rds))
- Microsoft SQL Azure ([azure.microsoft.com/en-us/services/sql-database](http://azure.microsoft.com/en-us/services/sql-database))
- Heroku ([www.heroku.com](http://www.heroku.com))
- Clustrix ([www.clustrix.com](http://www.clustrix.com))
- EnterpriseDB ([www.enterprisedb.com](http://www.enterprisedb.com))

### 3.5.6 Seccions protegides amb contrasenya

Pot resultar convenient protegir algunes seccions del nostre lloc web i restringir-ne l'accés a uns usuaris determinats. D'aquesta manera, voldrem que en accedir a

determinades seccions se’ns demanin unes credencials, és a dir, un nom d’usuari i una contrasenya.

La gestió dels usuaris i dels permisos i privilegis que tinguin es porta a terme a la banda del servidor. Així doncs, trobem diverses opcions per protegir el nostre lloc web:

- Si tenim accés al servidor web podem configurar-lo per limitar l'accés a determinades carpetes. Per exemple, si el servidor web és Apache podem configurar l'accés a les carpetes fent servir les directives d'Apache al fitxer *htaccess*.
- Si el nostre lloc web està basat en un CMS, aquest generalment ja porta incorporat un mòdul per a la gestió dels usuaris en el *back-end* en el qual podem definir quines accions pot realitzar cada tipus d'usuari.
- Si el nostre lloc web està fet a mida, llavors també caldrà que es desenvolupi en el *back-end* tota la lògica per a la gestió d'usuaris i les sessions d'aquests.

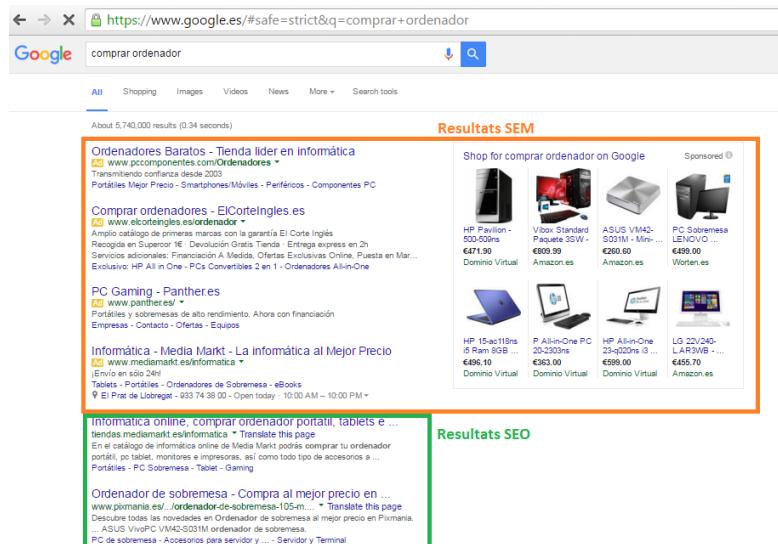
### 3.6 Posicionament web

Una vegada s’ha publicat el lloc web es vol que sigui visible des dels principals cercadors, i per a això existeixen diferents tècniques per millorar-ne el posicionament en la llista de resultats.

Alhora, es veu que els cercadors generen dos tipus de resultats:

- Els resultats **naturals** o **orgànics**, els quals es generen a través de l’algoritme imparcial del motor de cerca; les tècniques basades a millorar aquest tipus de posicionament s’anomenen **SEO**, Search Engine Optimization.
- Els resultats **patrocinats**, que basen la classificació mitjançant una contraprestació econòmica; les pràctiques basades a millorar aquest tipus de posicionament s’anomenen **SEM**, Search Engine Marketing.

A la figura 3.21 es veu on se situa la llista de resultats patrocinats i la llista de resultats orgànics o naturals.

**FIGURA 3.21.** Resultat d'una cerca

En taronja veieu la llista de resultats patrocinats i en verd els resultats naturals.

Generalment, una campanya de SEO té un resultat a llarg termini i una visibilitat excel·lent. En canvi, les campanyes de SEM tenen un resultat gairebé immediat, aconseguint posicionar-se ràpidament en els primers llocs. No obstant això, els preus s'aniran incrementant i igualment serà necessari aplicar alguna tècnica d'optimització per mantenir la visualització i competir amb els altres llocs.

Així doncs, a l'hora de millorar el posicionament d'un lloc web caldrà tenir en compte el cost econòmic i de temps per retornar la inversió realitzada utilitzant SEO o SEM.

### 3.6.1 Tècniques SEO

Per tal de millorar el posicionament del lloc web cal definir quines són les possibles paraules clau, **keywords**, que tenen relació amb el contingut del lloc. D'aquesta manera, cada paraula clau pot ser una sola paraula o un conjunt de paraules, i fan referència a una possible cerca que els usuaris poden fer en un cercador.

**Les tècniques SEO** consisteixen a preparar una web perquè sigui més comprensible per als motors de cerca, aconseguint així més tràfic provinent de les paraules clau que són rellevants per a l'activitat del lloc web.

No hi ha una manera directa de dir-li al cercador que per a una paraula clau mostri una determinada pàgina web, sinó que és el mateix cercador, segons el contingut que es trobi a la pàgina web, qui s'encarrega de posicionar-te o no en relació amb una cerca.

D'aquesta manera, cada URL d'un lloc web pot competir per un conjunt de paraules clau. Per exemple, si tenim un lloc web sobre una empresa de disseny

i desenvolupament d'aplicacions web es pot competir per paraules clau com ara “disseny web”, “desenvolupament web”, “disseny professional app”, etc.

### On col·locar les paraules clau

Per tal que el cercador reconegui els *keywords* s'ha de donar rellevància a certes paraules dins del contingut de la pàgina perquè el cercador indexi la pàgina sobre els conceptes pels quals volen ser trobats.

Així doncs, és important localitzar certes paraules:

- Al *title* de la pàgina i als encapçalamets, *h1*, *h2*, etc.
- Al contingut de la pàgina, sobretot als primers paràgrafs. Fins i tot hi pot aparèixer alguna paraula en negreta.
- Als enllaços. És interessant evitar enllaços de l'estil “llegir-ne més” o “fes clic aquí” i posar enllaços que continguin alguna de les paraules clau.
- A l'atribut *alt* de les imatges, encara que, perquè tinguin pes les paraules que es posin en aquest atribut, també hauran d'aparèixer en el text de la pàgina.

És important no usar una sola localització perquè el motor de cerca, quan rastregi el lloc web, pugui deduir que s'està parlant de certs conceptes associats a paraules clau.

### Etiquetes <meta>

Les etiquetes *meta* o *meta tags* s'usen per resumir el contingut de la pàgina web perquè pugui ser entesa pels robots i motors de cerca.

Hi ha diferents etiquetes <meta> que es poden descriure en el *head* d'una pàgina web:

#### 1. Etiqueta meta description

S'utilitza per descriure breument el contingut de la pàgina, i per molt que no es vegi des del navegador alguns cercadors la poden utilitzar com a resum a les pàgines de resultats. Google no utilitza aquesta descripció i usará com a descripció el text del document on s'han usat les paraules per les quals s'ha buscitat.

#### Exemple d'utilització de l'etiqueta “meta description”

```
1  <meta name="description" content="Aquesta és la descripció de la
pàgina />
```

En ocasions s'affirma que les paraules que apareixen a la *meta description* ajuden a guanyar visibilitat en els rànquings, però els robots d'alguns cercadors ja no es refien al 100% de les paraules definides en aquesta etiqueta. Malgrat tot, aquesta etiqueta és interessant perquè:

- **Pot ajudar a guanyar visites**, ja que quan un usuari veu una pàgina en els resultats del cercador pren la decisió de fer clic a l'enllaç en funció de la informació que es mostra. Una part important d'aquesta informació és la descripció, així que si aquesta és suficientment persuasiva augmenten les possibilitats que l'usuari hi faci clic.
- **Pot ajudar a la difusió**. Moltes xarxes socials també utilitzen la descripció per compartir les pàgines, així que una bona descripció facilita que l'usuari comparteixi la pàgina web.

Algunes consideracions per definir aquesta etiqueta són:

- Limitar la longitud del text: 156 caràcters és l'espai aproximat que els cercadors dediquen a la descripció.
- No és recomanable omplir-la de paraules clau, ja que el robot ho pot interpretar com un indicí de *spam*.
- No és recomanable repetir-la en totes les pàgines, ja que una manera que tenen els cercadors de detectar contingut duplicat és comprovar si la mateixa descripció apareix en més d'una pàgina.
- No és recomanable usar el mateix text del títol; de fet, en aquest cas seria millor deixar la descripció en blanc.
- Deixar la descripció en blanc no és ben bé un error, ja que quan no es proporciona el cercador s'encarrega de seleccionar un fragment del text de la pàgina.

## 2. Etiqueta meta keywords

S'utilitza per resumir el contingut d'una pàgina a partir d'unes quantes paraules clau.

### Exemple d'utilització de l'etiqueta "meta keywords"

```
1 <meta name="keywords" content="paraula clau 1, paraula 2, paraula  
3" />
```

Anys enrere era important definir les paraules clau en etiquetes `<meta>`, però avui en dia molts motors de cerca ja no se'n refien, atès que sovint eren falsejades pels desenvolupadors per tal de posicionar-se en *keywords* que potser no eren tan precisos pel seu contingut.

## 3. Etiqueta meta robots

Aquesta etiqueta permet donar instruccions directes als robots dels cercadors. Així doncs, depenent de la configuració que s'utilitzi, es pot autoritzar o no el rastreig i la indexació de la pàgina per part del cercador.

Les combinacions bàsiques d'aquesta etiqueta són:

- **Index, Follow**. És el valor per defecte i permet la indexació i el rastreig de la pàgina.

---

Entenem per rastreig de la pàgina el fet de recórrer o seguir recorrent el lloc web a través dels enllaços que es troben en el cos de la pàgina.

---

- NoIndex, Follow. Evita la indexació però permet el rastreig. És la configuració quan no es vol que la pàgina aparegui en els resultats del cercador.
- Index, NoFollow. Permet la indexació però evita el rastreig.
- NoIndex, NoFollow. Evita la indexació i el rastreig.

```
1 <meta name="robots" content="noIndex, noFollow">
```

### 3.6.2 Optimitzar el lloc web

Per optimitzar el contingut del lloc web és recomanable seguir, entre d'altres, les directrius de qualitat que marca Google, i que es poden consultar a [bit.ly/2qj5maI](http://bit.ly/2qj5maI).

A més a més, existeixen altres tècniques per millorar el posicionament d'un lloc web:

- **El domini:** es recomana usar un nom de domini que contingui alguna de les paraules clau. Per exemple, si es tracta d'una botiga *online* és recomanable incloure-hi el nom del producte o servei.
- **URL amigables:** es recomana utilitzar URL que facilitin saber més sobre la pàgina on porta. Per exemple, és millor usar un URL que sigui [www.elmeulloc.com/serveis.html](http://www.elmeulloc.com/serveis.html) que [www.elmeulloc.com/llistaS.html](http://www.elmeulloc.com/llistaS.html).
- **Els llenguatges:** cal tenir en compte que els llenguatges que més comprenen els cercadors són HTML i CSS. Per tant, tenir una bona estructura de pàgines HTML i CSS sense errors serà clau en el posicionament SEO.
- **Disseny responsive:** els cercadors premien si el lloc web està preparat per a mòbils i tauletes.
- **La popularitat:** els cercadors tenen en compte el nombre d'enllaços publicats en altres pàgines web que portin a aquest. D'aquesta manera, és important la col·laboració amb altres llocs web o comunitats, així com també dissenyar un pla de promoció a les principals xarxes socials.