

Durada: 88h



Continguts:

Tema 1: Internet, característiques i evolució

Tema 2: El llenguatge de marques HTML

Tema 3: Gestors de continguts Web

Tema 4: Sistemes de gestió de l'aprenentatge

Tema 5: Serveis de gestió d'arxius web

Tema 6: Aplicacions d'ofimàtica web

1. Què és Internet?

Internet és una xarxa global d'ordinadors interconnectats que permet la comunicació i l'intercanvi d'informació entre usuaris de tot el món. Aquesta xarxa utilitza un conjunt de protocols i tecnologies per facilitar la transferència de dades, la navegació web, l'enviament de correus electrònics i moltes altres activitats en línia. Internet ha esdevingut una part integral de la vida quotidiana per a moltes persones, i ofereix accés a una gran varietat de recursos i serveis, com ara pàgines web, xarxes socials, correu electrònic, botigues en línia i moltes altres opcions. És una eina poderosa que ha canviat la forma en què les persones es comuniquen, aprenen i treballen.

Internet té els seus orígens en un conjunt de projectes i desenvolupaments tecnològics que van tenir lloc a partir de la dècada de 1960. Ací hi ha una breu ressenya dels orígens d'Internet:

1. ARPANET: Els orígens d'Internet es remunten a la creació d'ARPANET (Advanced Research Projects Agency Network) als Estats Units el 1969. ARPANET va ser un projecte finançat pel Departament de Defensa dels Estats Units i va ser dissenyat inicialment per connectar universitats i centres de recerca. Va ser el primer sistema de commutació de paquets que va permetre la comunicació de manera descentralitzada entre diversos ordinadors.
2. Protocol IP: Els protocols de comunicació TCP/IP (Transmission Control Protocol/Internet Protocol) van ser desenvolupats per Bob Kahn i Vinton Cerf a la dècada de 1970. Aquests protocols van ser fonamentals per permetre la comunicació i l'interconnexió d'ordinadors a tot el món.
3. Expansió mundial: A mesura que es desenvolupava la tecnologia i s'estandarditzaven els protocols, la xarxa ARPANET va créixer i es va connectar amb altres xarxes similars en altres països, creant una xarxa global que es coneixia com a Internet.
4. World Wide Web: El 1989, el científic britànic Tim Berners-Lee va inventar el World Wide Web (WWW) a la Organització Europea per a la Recerca Nuclear (CERN). Aquesta tecnologia va permetre la creació de pàgines web i l'accés fàcil a la informació a través de navegadors web.
5. Creixement exponencial: A partir de la dècada de 1990, Internet va experimentar un creixement exponencial amb la incorporació de més usuaris, empreses i serveis en línia. Va esdevenir una eina essencial per a la comunicació, el comerç, l'educació i moltes altres àrees.

Huí en dia, Internet és una part integral de la nostra societat i ha canviat la forma en què vivim, treballem i ens comuniquem. La seua història està marcada per una sèrie de contribucions i avanços tecnològics que han donat lloc a la xarxa global que coneixem ara.

2. La Web o Web 1.0

La Web 1.0, també coneguda com la primera generació d'Internet, va ser una etapa inicial d'Internet caracteritzada per un enfocament predominantment estàtic i unidireccional. Aquí tens algunes de les característiques de la Web 1.0:

1. **Contingut Estàtic:** La majoria de les pàgines web de la Web 1.0 contenien contingut estàtic, el qual estava escrit en HTML i no canviava de manera dinàmica en funció de l'usuari.
2. **Llegir, no interactuar:** Les pàgines web es centraven principalment en la presentació d'informació i no permetien una interacció activa de l'usuari. Els usuaris podien llegir el contingut, però no participaven activament ni creaven contingut.
3. **Navegació senzilla:** Les pàgines web sovint tenien una estructura de navegació senzilla, amb menús de text bàsics i pocs enllaços hipertext.
4. **Falta de mitjans socials:** Les xarxes socials no eren una característica comuna en la Web 1.0, i no hi havia formes fàcils de compartir contingut o interaccionar amb altres usuaris.
5. **Escassa personalització:** Les pàgines web no oferien una personalització significativa en funció de les preferències de l'usuari. El contingut es mostrava igual per a tots els visitants.
6. **Tecnologia dial-up:** Molts usuaris es connectaven a Internet mitjançant connexions dial-up lentes, la qual cosa limitava la velocitat de càrrega de les pàgines web.
7. **Comerç electrònic inicial:** Si bé hi havia algunes empreses que operaven en línia, el comerç electrònic encara era incipient i no tan àmpliament utilitzat com ara.
8. **Directoris i cercadors:** Per trobar informació a la Web 1.0, els usuaris sovint depenien de directoris web o cercadors bàsics com Yahoo! o AltaVista.
9. **Anuncis estàtics:** La publicitat en línia es limitava principalment a anuncis estàtics que es mostraven als costats de les pàgines web o en forma de bàners.

La Web 1.0 va ser una fase important en el desenvolupament d'Internet, però posteriorment va evolucionar cap a formes més interactives i dinàmiques de comunicació i interacció en línia, conegudes com la Web 2.0 .

3. Web 2.0

La Web 2.0, també coneguda com la segona generació d'Internet, va arrancar en 2004 i va representar una transició important de la Web 1.0 cap a una web més interactiva i participativa. Aquí tens algunes de les característiques de la Web 2.0:

1. **Interactivitat:** La Web 2.0 va posar un èmfasi més gran en la interacció de l'usuari amb el contingut en línia. Això va permetre als usuaris comentar, compartir i col·laborar en l'edició de contingut.
2. **Xarxes Socials:** Les xarxes socials com Facebook, Twitter i LinkedIn van sorgir com a part integral de la Web 2.0. Aquestes plataformes van permetre als usuaris connectar-se amb altres persones i compartir contingut i experiències.
3. **Contingut generat per l'usuari:** En la Web 2.0, els usuaris podien contribuir amb el seu propi contingut, com ara pujar vídeos a YouTube, escriure blocs, o afegir comentaris i opinions als llocs web.

4. Aplicacions web: Les aplicacions web (conegudes com a "webapps") es van fer més populars en aquesta etapa. Això va permetre als usuaris utilitzar aplicacions directament a través del navegador sense necessitat de descarregar ni instal·lar software.
5. Col·laboració en línia: Les eines de col·laboració en línia com Google Docs van permetre a les persones treballar en documents de manera simultània i compartir-los fàcilment amb altres usuaris.
6. RSS i sindicació de contingut: La Web 2.0 va fer servir tecnologies com les feeds RSS per facilitar la subscripció a contingut actualitzat i la sindicació d'informació de llocs web.
7. Personalització: Les pàgines web de la Web 2.0 sovint oferien opcions de personalització, com ara pàgines d'inici personalitzades i recomanacions basades en les preferències de l'usuari.
8. Contingut en temps real: La Web 2.0 va ser pionera en la publicació de contingut en temps real, com ara actualitzacions en temps real a les xarxes socials i blogs.

La Web 2.0 va canviar radicalment la manera com les persones interactuaven en línia i va donar lloc a un ecosistema més ric de serveis i aplicacions web. Va ser una etapa crucial en el desenvolupament d'Internet i va obrir la porta a les futures innovacions en línia.

4. Línia del temps de l'evolució de la Web

Línia del temps de l'evolució de la Web



Web 1.0 (1990 – inicis 2000)

- 1991: Tim Berners-Lee crea la primera pàgina web.
- 1993: Mosaic, el primer navegador gràfic.

- Pàgines estàtiques.
- L'usuari només pot llegir la informació.
- Poc o nul contingut generat per usuaris.
- Exemples: Yahoo! (1994), GeoCities, pàgines personals i portals.

Web 2.0 (2004 – actualitat, encara vigent)

- 2004: O'Reilly Media popularitza el concepte Web 2.0.
- Pàgines dinàmiques i interactives.
- Els usuaris poden crear i compartir contingut.
- Aparició de xarxes socials i plataformes collaboratives.
- Exemples: Blogs (Blogger, WordPress), Wikipedia (2001, però guanya força amb la Web 2.0), Facebook (2004), YouTube (2005), Twitter (2006)...
- Aplicacions web com Google Docs.

Web 3.0 (aprox. 2010 – futur)

- Encara en desenvolupament i sense data exacta d'inici.
- Web semàntica: la informació és entesa per màquines.
- Ús d'IA i aprenentatge automàtic.
- Dades més personalitzades i connectades.
- Integració amb blockchain, criptomonedes i descentralització.
- Exemples: assistents com Siri, Alexa, Google Assistant, Webs amb recomanacions intel·ligents (Netflix, Spotify).
- Projectes basats en blockchain: **Ethereum (2015)** → Plataforma per crear aplicacions descentralitzades (dApps) i smart contracts, **Uniswap** → Intercanvi descentralitzat (DEX) que permet fer trading sense intermediaris, **IPFS (InterPlanetary File System)** → Sistema d'emmagatzematge distribuït, alternativa descentralitzada al web tradicional, **NFTs (Non-Fungible Tokens)** → Tokens únics per a art digital, colleccionables i identitats digitals, **DAOs (Organitzacions Autònomes Descentralitzades)** → Comunitats que prenen decisions a través de blockchain...
- Projectes basats en metavers: **Decentraland** → Món virtual descentralitzat on els usuaris compren parcel·les digitals i creen experiències, **The Sandbox** → Plataforma de jocs i creació 3D basada en blockchain i NFTs, **Meta (antic Facebook)** → Impulsa la idea del metavers amb realitat virtual (Oculus/Meta Quest) i entorns socials 3D, **Roblox i Fortnite (Epic Games)** → No són descentralitzats, però funcionen com a metaversos socials amb concerts virtuals, economia pròpia i entorns creatius

5. Tecnologies 2.0. Pila LAMP

La pila LAMP és un conjunt de programari que es fa servir com a base per a desenvolupar aplicacions web. Les lletres de "LAMP" representen les següents tecnologies:

1. **Linux:** És el sistema operatiu de codi obert que forma la base de la pila. Linux és conegut per la seva fiabilitat i seguretat, i és àmpliament utilitzat en servidors web.
2. **Apache:** És el servidor web més popular del món. L'Apache és utilitzat per gestionar les sol·licituds web i lliurar pàgines web als navegadors dels usuaris. També és de codi obert i molt configurable.
3. **MySQL:** És un sistema de gestió de bases de dades relacional de codi obert. MySQL s'utilitza per emmagatzemar i gestionar les dades que es fan servir en una aplicació web. És conegut per la seva rapidesa i fiabilitat.
4. **PHP:** És un llenguatge de programació de codi obert que s'utilitza per crear pàgines web dinàmiques. PHP pot generar contingut web personalitzat en funció de les dades emmagatzemades a la base de dades. És molt popular en el desenvolupament web.

La pila LAMP és molt utilitzada en el desenvolupament d'aplicacions web, especialment en el món del software de codi obert. Ofereix una plataforma sòlida per crear i executar llocs web i aplicacions web dinàmiques de manera eficient i fiable. Aquesta combinació de programari permet als desenvolupadors crear aplicacions web escalables i de baix cost.

Variants principals de la pila LAMP

- WAMP → Windows, Apache, MySQL, PHP/Python/Perl
- MAMP → macOS, Apache, MySQL, PHP/Python/Perl
- LAPP → Linux, Apache, PostgreSQL, PHP/Python/Perl
- LEMP → Linux, Nginx (es pronuncia "Engine-X"), MySQL/MariaDB, PHP/Python/Perl
- LLMP → Linux, Lighttpd, MySQL/MariaDB, PHP. Fa servir Lighttpd com a servidor web (molt lleuger).

Altres variants modernes

MEAN → MongoDB, Express.js, Angular, Node.js (Stack 100% JavaScript).

MERN → MongoDB, Express.js, React, Node.js (Variant de MEAN amb React).

LEPP → Linux, Nginx, PostgreSQL, PHP (Similar a LEMP però amb PostgreSQL).

6. Navegador Web

Un navegador web és un programa que permet visualitzar la informació que conté una pàgina web, ja que interpreta el codi de la pàgina i el visualitza en la pantalla. El primer

navegador web va aparèixer en els anys 90, i va ser desenvolupat en els laboratoris del CERN (the European Organization for Nuclear Research).

Inicialment només funcionava en entorns UNIX, però no tardaren en aparèixer versions per a Windows. Després entrà en el mercat Netscape Navigator, que ràpidament superà en capacitats i velocitat a navegadors anteriors com ara Mosaic. A finals del 2004 apareix i el mercat de Firefox, una branca de desenvolupament de Mozilla. Actualment ha aconseguit una gran popularitat gràcies a les seues pestanyes, la seua lleugeresa i la seua versatilitat, basada en un sistema modular d'extensions.

Un navegador mòbil o micro navegador és un navegador web dissenyat per a l'ús de dispositius mòbils i de reduïdes dimensions, com a telèfons mòbils, telèfon intel·ligents o tàblets. Els micro-navegadors estan optimitzats per a mostrar contingut d'Internet en pantalles reduïdes, i utilitzen grandàries d'arxiu reduïts per a ser instal·lats en dispositius amb memòries de baixa capacitat.

7. Servidors Web

Un servidor web és un programa dissenyat per a acceptar peticions HTTP del navegador i servir les pàgines web que té allotjades. Cada lloc web té una adreça única coneguda com a URL (Uniform Resource Locator - Localizador Uniforme de Recursos). El protocol que s'utilitza per a servir pàgines web és HTTP, el nom de domini indica a l'ordinador d'Internet que ens va a donar la informació sol·licitada quina és la ruta de l'arxiu sol·licitat del servidor.

Adreces IP o adreces d'Internet: són les adreces que utilitza el protocol IP per a identificar de forma única un dispositiu en Internet. Aquest protocol permet que es comuniquen els dispositius connectats a la xarxa, ja siguin ordinadors, routers, punts d'accés, dispositius mòbils, servidors, etc.

Els dominis: Són els noms dels llocs web. En lloc d'utilitzar les adreces IP per a accedir als llocs web, s'utilitza el domini.

- a) Internacionals o genèrics: Fan referència al tipus d'organització que és propietària del domini.
- b) Dominis territorials o dominis de país: Representen a un país i estan a càrrec de l'Estat de cada Nació.

Els proveïdors de serveis de Internet o ISP (Internet Service Provider): Estableixen una connexió a Internet es necessita un proveïdor de serveis d'Internet o ISP.

Existeixen 3 elements indispensables en Internet perquè siga possible la connexió:

- 1.- Servidor Web. És un servidor que està preparat per a servir pàgines web les 24 hores del dia.
- 2.- Domini. És el lloc web al que volem accedir. S'introdueix en la barra d'adreces del navegador.

3.- Servidor DNS. És l'encarregat de transformar el nom del domini d'un servidor web. El navegador estableix una connexió amb l'adreça IP proporcionada pel servidor DNS, El servidor web envia l'arxiu corresponent i el navegador mostra la pagina sol·licitada.

8. Funcionament bàsic d'un Servidor Web

Un servidor web és un programa informàtic que funciona com a intermediari entre el teu navegador web i els fitxers que es troben al servidor. El seu funcionament bàsic es pot resumir en els següents passos:

El navegador web del client (usuari) envia una sol·licitud HTTP (Hypertext Transfer Protocol) al servidor web. Aquesta sol·licitud pot ser per accedir a una pàgina web, una imatge, un fitxer CSS o qualsevol altre recurs disponible al servidor.

El servidor web rep la sol·licitud i la processa. Comprova si el recurs sol·licitat existeix i té els permisos necessaris per ser accedit.

Si el recurs sol·licitat existeix i es pot accedir, el servidor web crea una resposta HTTP. Aquesta resposta conté les dades del recurs sol·licitat, com ara el contingut HTML d'una pàgina web o la imatge sol·licitada. La resposta també inclou capçaleres HTTP que contenen informació sobre la resposta, com ara la data de la resposta i el tipus de contingut.

El servidor web envia la resposta HTTP al navegador del client a través de la xarxa.

El navegador del client rep la resposta del servidor web i la processa. Mostra el contingut de la pàgina web o el recurs sol·licitat a l'usuari en la finestra del navegador.

L'usuari pot interactuar amb el contingut de la pàgina mitjançant clics, formularis i altres accions. Quan es realitzen aquestes interaccions, el navegador envia noves sol·licituds HTTP al servidor web per obtenir més recursos o actualitzacions.

Aquest és un funcionament bàsic d'un servidor web. Hi ha molts programes de servidor web disponibles, com Apache, Nginx i Microsoft IIS, que executen aquest procés per permetre als usuaris accedir a pàgines web i altres continguts a través d'Internet.

9. Front-end vs back-end

"Front-end" i "back-end" són dos termes importants en el desenvolupament web que fan referència a les diferents parts d'una aplicació web.

Front-end: El "front-end" és la part visible i interactiva d'una aplicació web amb la qual els usuaris interactuen directament. Aquesta part es centra en la interfície d'usuari i en com es presenta el contingut. Alguns dels seus elements principals inclouen:

1. **Pàgines web:** Les pàgines web que veuen els usuaris, amb tot el seu contingut visual i elements interactius, com botons, formularis i menús.
2. **HTML:** És el llenguatge de marcatge que s'utilitza per estructurar i crear l'estructura de les pàgines web.

3. **CSS:** És l'estil i la presentació de les pàgines web. S'utilitza per definir els colors, la tipografia, la disposició i altres aspectes visuals.
4. **JavaScript:** És un llenguatge de programació que permet afegir funcionalitat interactiva i dinàmica a les pàgines web, com animacions o validació de formularis.

Back-end: El "back-end" és la part oculta d'una aplicació web que treballa en el background per fer que tot funcione correctament. Aquesta part se centra en la gestió de dades, la lògica de negoci i l'emmagatzematge de la informació. Alguns dels seus components principals inclouen:

1. **Servidor:** És el programari que s'executa en els servidors web i que processa les sol·licituds dels usuaris. També gestiona la comunicació amb la base de dades i altres serveis.
2. **Base de dades:** És on s'emmagatzemen i recuperen les dades de l'aplicació web. Aquesta pot ser una base de dades relacional com MySQL o una base de dades NoSQL com MongoDB.
3. **Lògica de negoci:** Són les regles i processos que determinen com funciona l'aplicació. Això pot incloure la gestió d'usuaris, l'autenticació, la gestió de contingut i altres funcionalitats.
4. **APIs:** Les APIs (Interfaces de Programació d'Aplicacions) permeten que les diferents parts de l'aplicació s'interconnecten i es comuniquin entre elles. Això pot incloure l'enviament i la recepció de dades entre el front-end i el back-end.

En resum, el "front-end" és la part visible i interactiva de l'aplicació web, mentre que el "back-end" és la part que gestiona la lògica de negoci, les dades i la comunicació amb els servidors. Els desenvolupadors "front-end" i "back-end" treballen junts per crear aplicacions web completes i funcionals.



1. [Introducció HTML](#)
 2. [HTML bàsic](#)
 3. [Elements HTML](#)
 4. [Atributs HTML](#)
 5. [Encapçalaments HTML](#)
 6. [Paràgrafs HTML](#)
 7. [Estils HTML](#)
 8. [Format HTML](#)
 9. [Cites HTML](#)
 10. [Comentaris HTML](#)
 11. [Colors HTML](#)
 12. [HTML CSS](#)
 13. [Enllaços HTML](#)
 14. [Imatges HTML](#)
 15. [Favicon HTML](#)
 16. [Títol de la pàgina HTML](#)
 17. [Taules HTML](#)
 18. [Llistes HTML](#)
 19. [Bloc HTML i en línia](#)
 20. [Classes HTML](#)
 21. [Id HTML](#)
 22. [Iframes HTML](#)
 23. [HTML JavaScript](#)
 24. [Rutes dels fitxers HTML](#)
 25. [Capçalera HTML](#)
 26. [Disseny HTML](#)
 27. [HTML Responsive](#)
 28. [Codi informàtic HTML](#)
 29. [Semàntica HTML](#)
 30. [Guia d'estil HTML](#)
 31. [Entitats HTML](#)
 32. [Símbols HTML](#)
 33. [Conjunt de caràcters HTML](#)
 34. [HTML vs. XHTML](#)
 35. [Formularis HTML](#)
 36. [Atributs de formulari HTML](#)
 37. [Elements del formulari HTML](#)
 38. [Tipus de dades d'entrada en formularis HTML](#)
 39. [Atributs d'entrada en formularis HTML](#)
-

1. Introducció HTML

HTML és el llenguatge de marques estàndard per a pàgines web.

Amb HTML podeu crear el vostre propi lloc web.

L'HTML és fàcil d'aprendre: el gaudiràs!

HTML és el llenguatge de marques estàndard per crear pàgines web.

Què és HTML?

- HTML significa Hyper Text Markup Language
- HTML és el llenguatge de marques estàndard per crear pàgines web
- HTML descriu l'estructura d'una pàgina web
- HTML consta d'una sèrie d'elements
- Els elements HTML indiquen al navegador com mostrar el contingut
- Els elements HTML etiqueten fragments de contingut com ara "això és un encapçalament", "això és un paràgraf", "això és un enllaç", etc.

Exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

Exemple explicat

- La <!DOCTYPE html>declaració defineix que aquest document és un document HTML5
- L'element <html> és l'element arrel d'una pàgina HTML
- L'element <head> conté metainformació sobre la pàgina HTML
- L'element <title> especifica un títol per a la pàgina HTML (que es mostra a la barra de títol del navegador o a la pestanya de la pàgina)
- L'element <body> defineix el cos del document, i és un contenidor per a tots els continguts visibles, com ara encapçalaments, paràgrafs, imatges, hiperenllaços, taules, llistes, etc.
- L'element <h1> defineix un encapçalament gran
- L'element <p> defineix un paràgraf

Què és un element HTML?

Un element HTML es defineix per una etiqueta d'inici, una mica de contingut i una etiqueta final:

< tagname > El contingut va aquí... < /tagname >

L' element HTML és tot, des de l'etiqueta inicial fins a l'etiqueta final:

< h1 > El meu primer encapçalament < /h1 >

< p > El meu primer paràgraf. < /p >

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
 	<i>none</i>	<i>none</i>

Nota: alguns elements HTML no tenen contingut (com l'element
). Aquests elements s'anomenen elements buits. Els elements buits no tenen una etiqueta final!

Navegadors web

La finalitat d'un navegador web (Chrome, Edge, Firefox, Safari) és llegir documents HTML i mostrar-los correctament.

Un navegador no mostra les etiquetes HTML, però les utilitza per determinar com mostrar el document:

Història HTML

Des dels primers dies de la World Wide Web, hi ha hagut moltes versions d'HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

2. HTML bàsic

En aquest capítol mostrarem alguns exemples bàsics d'HTML. No us preocupeu si fem servir etiquetes que encara no coneixeu.

Documents HTML

Tots els documents HTML han de començar amb una declaració de tipus de document:

<!DOCTYPE html>.

El document HTML en si comença amb <html> i acaba amb </html>.

La part visible del document HTML es troba entre <body> i </body>. Exemple:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

La declaració <!DOCTYPE>

La declaració <!DOCTYPE> representa el tipus de document i ajuda els navegadors a mostrar les pàgines web correctament.

Només ha d'aparèixer una vegada, a la part superior de la pàgina (davant de les etiquetes HTML).

La declaració <!DOCTYPE> no distingeix entre majúscules i minúscules.

La declaració <!DOCTYPE> per a HTML5 és:

```
<!DOCTYPE html>
```

Encapçalaments HTML

Els encapçalaments HTML es defineixen amb les etiquetes de <h1> a <h6>.

<h1> defineix l'encapçalament més important. <h6> defineix l'encapçalament menys important. Exemple:

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Paràgrafs HTML

Els paràgrafs HTML es defineixen amb l'etiqueta <p>. Exemple:

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Enllaços HTML

Els enllaços HTML es defineixen amb l'etiqueta <a>. Exemple:

```
<a href="https://www.w3schools.com">This is a link</a>
```

La destinació de l'enllaç s'especifica a l'atribut href.

Els atributs s'utilitzen per proporcionar informació addicional sobre els elements HTML.

Imatges HTML

Les imatges HTML es defineixen amb l'etiqueta .

El fitxer font (src), el text alternatiu (alt), width i height es proporcionen com a atributs:

```

```

Veure codi font HTML:

Feu clic amb el botó dret en una pàgina HTML i seleccioneu "Mostra la font de la pàgina" (a Chrome) o "Mostra la font" (a l'Edge) o similar en altres navegadors. Això obrirà una finestra que conté el codi font HTML de la pàgina.

Inspeccioneu un element HTML:

Feu clic amb el botó dret sobre un element (o una àrea en blanc) i trieu "Inspeccionar" o "Inspeccionar l'element" per veure de quins elements estan formats (veureu tant l'HTML com el CSS). També podeu editar l'HTML o CSS sobre la marxa al tauler Elements o Estils que s'obre.

3. Elements HTML

Un element HTML es defineix per una etiqueta d'inici, una mica de contingut i una etiqueta final.

Nota: alguns elements HTML no tenen contingut (com l'element `
`). Aquests elements s'anomenen elements buits. Els elements buits no tenen una etiqueta final!

Elements dins d'elements en HTML

Els elements HTML es poden imbricar (això vol dir que els elements poden contenir altres elements).

Tots els documents HTML consisteixen en elements HTML imbricats.

L'exemple següent conté quatre elements HTML (`<html>`, `<body>`, `<h1>` i `<p>`). Exemple:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

Exemple explicat

L'element `<html>` és l'element arrel i defineix tot el document HTML.

Té una etiqueta d'inici `<html>` i una de final `</html>`.

Aleshores, dins de l'element `<html>` hi ha un element `<body>`:

L'element `<body>` defineix el cos del document.

Té una etiqueta d'inici `<body>` i una de final `</body>`.

Aleshores, dins de l'element `<body>` hi ha altres dos elements: `<h1>` i `<p>`

L'element `<h1>` defineix un encapçalament.

Té una etiqueta d'inici `<h1>` i una de final `</h1>`:

L'element `<p>` defineix un paràgraf.

Té una etiqueta d'inici `<p>` i una de final `</p>`:

No salteu mai l'etiqueta final

Alguns elements HTML es mostraran correctament, fins i tot si oblideu l'etiqueta final:

```
<html>
  <body>
    <p>This is a paragraph <!-- falta tancar p -->
    <p>This is a paragraph <!-- falta tancar p -->
  </body>
</html>
```


Mai confieu en això! Es poden produir resultats i errors inesperats si oblideu l'etiqueta final!

Elements HTML buits

Els elements HTML sense contingut s'anomenen elements buits.

L'etiqueta `
` defineix un salt de línia i és un element buit sense una etiqueta de tancament. Exemple:

```
<p>This is a <br> paragraph with a line break.</p>
```

HTML no distingeix entre majúscules i minúscules

Les etiquetes HTML no distingeixen entre majúscules i minúscules: `<P>` significa el mateix que `<p>`.

L'estàndard HTML no requereix etiquetes en minúscules, però W3C **recomana** minúscules en HTML i **exigeix** minúscules per a tipus de document més estrictes com XHTML.

4. Atributs HTML

Els atributs HTML proporcionen informació addicional sobre els elements HTML.

Atributs HTML

- Tots els elements HTML poden tenir **atributs**
- Els atributs proporcionen **informació addicional** sobre els elements
- Els atributs sempre s'especifiquen a **l'etiqueta d'inici**
- Els atributs solen venir en parells nom/valor com: **nom="valor"**

L'atribut href

L'etiqueta `<a>` defineix un hiperenllaç. L' href atribut especifica l'URL de la pàgina a la qual va l'enllaç. Exemple:

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

L'atribut src

L'etiqueta `` s'utilitza per incrustar una imatge en una pàgina HTML. L'atribut src especifica el camí a la imatge que es mostrarà. Exemple:

```

```

Hi ha dues maneres d'especificar l'URL a l'atribut src :

1. URL absolut - Enllaços a una imatge externa allotjada en un altre lloc web. Exemple: `src="https://www.w3schools.com/images/img_girl.jpg"` .

Notes: les imatges externes poden estar sota copyright. Si no obteniu permís per utilitzar-lo, és possible que infringiu les lleis de drets d'autor. A més, no podeu controlar imatges externes; es pot eliminar o canviar de sobte.

2. URL relatiu - Enllaços a una imatge allotjada dins del lloc web. Ací, l'URL no inclou el nom de domini. Si l'URL comença sense barra inclinada, serà relatiu a la pàgina actual. Exemple: `src="img_girl.jpg"`. Si l'URL comença amb una barra inclinada, serà relatiu al domini. Exemple: `src="/images/img_girl.jpg"`.

Consell: sempre és millor utilitzar URL relatius. No es trencaran si canvies de domini.

Els atributs d'amplada i alçada

L'etiqueta `` també ha de contenir els atributs `width` i `height`, que especifiquen l'amplada i l'alçada de la imatge (en píxels). Exemple:

```

```

L'atribut alt

L'atribut `Alt` obligatori per a l'etiqueta `` especifica un text alternatiu per a una imatge, si la imatge per algun motiu no es pot mostrar. Això pot ser degut a una connexió lenta, a un error en l'atribut `src` o si l'usuari utilitza un lector de pantalla. Exemple:

```

```

L'atribut d'estil

L'atribut `style` s'utilitza per afegir estils a un element, com ara el color, el tipus de lletra, la mida i molt més. Exemple:

```
<p style="color:red;">This is a red paragraph.</p>
```

L'atribut lang

Sempre hauríeu d'incloure l'atribut `lang` dins de l'etiqueta `<html>`, per declarar l'idioma de la pàgina web. Això està pensat per ajudar els motors de cerca i els navegadors.

L'exemple següent especifica l'anglès com a idioma:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

Els codis de país també es poden afegir al codi d'idioma de l'atribut `lang`. Així, els dos primers caràcters defineixen l'idioma de la pàgina HTML i els dos últims caràcters defineixen el país.

L'exemple següent especifica l'anglès com a idioma i els Estats Units com a país:

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```

L'atribut del títol

L'atribut `title` defineix informació addicional sobre un element.

El valor de l'atribut `title` es mostrarà com a informació en un recuadre quan passeu el ratolí per sobre de l'element. Exemple:

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

Suggerim: Utilitzeu sempre els atributs en minúscules

L'estàndard HTML no requereix noms d'atributs en minúscules.

L'atribut `title` (i tots els altres atributs) es pot escriure amb majúscules o minúscules com **title** o **TITLE**.

Tanmateix, W3C **recomana** atributs en minúscules en HTML i **exigeix** atributs en minúscules per a tipus de documents més estrictes com XHTML.

Us suggerim: citar sempre els valors dels atributs

L'estàndard HTML no requereix cometes al voltant dels valors dels atributs.

No obstant això, W3C **recomana** cites en HTML i **exigeix** cites per a tipus de documents més estrictes com XHTML.

Cometes simples o dobles?

Les cometes dobles al voltant dels valors dels atributs són les més habituals en HTML, però també es poden utilitzar cometes simples.

En algunes situacions, quan el valor de l'atribut conté cometes dobles, és necessari utilitzar cometes simples:

```
<p title='John "ShotGun" Nelson'>
```

O viceversa:

```
<p title="John 'ShotGun' Nelson">
```

5. Encapçalaments HTML

Els encapçalaments HTML són títols o subtítols que voleu mostrar en una pàgina web.

Encapçalaments HTML

Els encapçalaments HTML es defineixen amb les etiquetes de <h1> a <h6>.

<h1> defineix l'encapçalament més important. <h6> defineix l'encapçalament menys important. Exemple:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Nota: els navegadors afegeixen automàticament un espai en blanc (un marge) abans i després d'un encapçalament.

Els encapçalaments són importants

Els motors de cerca utilitzen els encapçalaments per indexar l'estructura i el contingut de les vostres pàgines web.

Els usuaris sovint passen per una pàgina pels seus encapçalaments. És important utilitzar els encapçalaments per mostrar l'estructura del document.

<h1> els encapçalaments s'han d'utilitzar per als encapçalaments principals, seguits dels <h2> encapçalaments, després els menys importants <h3>, i així successivament.

Nota: Utilitzeu encapçalaments HTML només per als encapçalaments. No utilitzeu encapçalaments per fer que el text siga **GRAN** o **en negreta**.

Títols més grans

Cada encapçalament HTML té una mida predeterminada. Tanmateix, podeu especificar la mida de qualsevol encapçalament amb l' `style` atribut, utilitzant la `font-size` propietat CSS. Exemple:

```
<h1 style="font-size:60px;">Heading 1</h1>
```

6. Paràgrafs HTML

Un paràgraf sempre comença en una línia nova i sol ser un bloc de text.

Paràgrafs HTML

L'element HTML <p> defineix un paràgraf.

Un paràgraf sempre comença en una línia nova i els navegadors afegeixen automàticament algun espai en blanc (un marge) abans i després d'un paràgraf. Exemple:

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Visualització HTML

No podeu estar segur de com es mostrarà l'HTML.

Les pantalles grans o menudes i les finestres redimensionades crearan resultats diferents.

Amb HTML, no podeu canviar la visualització afegint espais addicionals o línies addicionals al vostre codi HTML.

El navegador eliminarà automàticament els espais i línies addicionals quan es mostre la pàgina. Exemple:

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>
```

Regles horitzontals HTML

L'etiqueta <hr> defineix un salt temàtic en una pàgina HTML i es mostra com a una regla horitzontal. L'element <hr> s'utilitza per separar el contingut (o definir un canvi) en una pàgina HTML. Exemple:

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
```

```
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

L'etiqueta `<hr>` és una etiqueta buida, el que significa que no té cap etiqueta final.

Salts de línia HTML

L'element HTML `
` defineix un salt de línia.

Utilitzeu `
` si voleu un salt de línia (una línia nova) sense començar un paràgraf nou:

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

L'etiqueta `
` és una etiqueta buida, el que significa que no té cap etiqueta final.

El problema del poema

Aquest poema es mostrarà en una sola línia. Exemple:

```
<p>
  My Bonnie lies over the ocean.
  My Bonnie lies over the sea.
  My Bonnie lies over the ocean.
  Oh, bring back my Bonnie to me.
</p>
```

Solució: l'element HTML `<pre>`

L'element HTML `<pre>` defineix el text preformatat.

El text dins d'un element `<pre>` es mostra amb un tipus de lletra d'amplada fixa (normalment Courier) i conserva tant els espais com els salts de línia. Exemple:

```
<pre>
  My Bonnie lies over the ocean.
  My Bonnie lies over the sea.
  My Bonnie lies over the ocean.
  Oh, bring back my Bonnie to me.
</pre>
```

7. Estils HTML

L'atribut d'estil HTML s'utilitza per afegir estils a un element, com ara el color, el tipus de lletra, la mida i molt més.

```
<!DOCTYPE html>
<html>
<body>
  <p>I am normal</p>
  <p style="color:red;">I am red</p>
  <p style="color:blue;">I am blue</p>
  <p style="font-size:50px;">I am big</p>
</body>
</html>
```

L'atribut d'estil HTML

La configuració de l'estil d'un element HTML es pot fer amb l'atribut `style`. L'atribut d'estil HTML té la sintaxi següent:

`<tagname style="property:value;">`

La propietat és una propietat CSS. El valor és un valor CSS. Aprenderàs més sobre CSS més adavant.

Color de fons

La propietat CSS background-color defineix el color de fons per a un element HTML:

```
<body style="background-color:powderblue;">
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
```

Altres exemples:

```
<body>
  <h1 style="background-color:powderblue;">This is a heading</h1>
  <p style="background-color:tomato;">This is a paragraph.</p>
</body>
```

Color del text

La propietat de color CSS defineix el color del text per a un element HTML. Exemple:

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

Fonts

La propietat CSS font-family defineix el tipus de lletra que s'utilitzarà per a un element HTML. Exemple:

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

Mida del text

La propietat CSS font-size defineix la mida del text per a un element HTML. Exemple:

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

Alineació de text

La propietat CSS text-align defineix l'alineació horitzontal del text per a un element HTML. Exemple:

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

8. Format de text HTML

HTML conté diversos elements per definir text amb un significat especial: negreta, cursiva...

Elements de format HTML

Els elements de format es van dissenyar per mostrar tipus especials de text:

- `` - Text en negreta
- `<i>` - Text en cursiva
- `<mark>` - Text marcat

- **<small>** - Text més xicotet
- **** - Text eliminat
- **<ins>** - Text inserit
- **<sub>** - Text subíndex
- **<sup>** - Text en superíndex

Element **<small>** HTML

L'element HTML **<small>** defineix el text més xicotet:

Element **<mark>** HTML

L'element HTML **<mark>** defineix el text que s'ha de marcar o ressaltar. Exemple:

```
<p>Do not forget to buy <mark> milk </mark> today.</p>
```

HTML **** Element

L'element HTML **** defineix el text que s'ha suprimit d'un document. Els navegadors solen tocar una línia a través del text suprimit. Exemple:

```
<p>My favorite color is<del>blue</del>red.</p>
```

Element **<ins>** HTML

L'element **<ins>** HTML defineix un text que s'ha inserit en un document. Els navegadors solen subratllar el text inserit. Exemple:

```
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
```

Element **<sub>** HTML

L'element HTML **<sub>** defineix el text de subíndex. El text de subíndex apareix mig caràcter per baix de la línia normal i, de vegades, es representa amb un tipus de lletra més xicotet. El text d'índex es pot utilitzar per a fórmules químiques, com H₂O:

Element **<sup>** HTML

L'element HTML **<sup>** defineix el text en superíndex. El text en superíndex apareix mig caràcter per sobre de la línia normal i, de vegades, es representa amb un tipus de lletra més xicotet.

9. Elements de citació i citació HTML

En aquest punt repassarem els elements HTML **<blockquote>**, **<q>**, **<abbr>**, **<address>**, **<cite>** i **<bdo>**.

HTML **<blockquote>** per a cites

L'element **<blockquote>** HTML defineix una secció que es cita d'una altra font.

Els navegadors solen sagnar els ements **<blockquote>**

HTML **<q>** per a cites curtes

L'etiqueta HTML **<q>** defineix una cita curta.

Els navegadors normalment insereixen cometes al voltant de la citació.

Abreviatures HTML

L'etiqueta HTML `<abbr>` defineix una abreviatura o un acrònim, com ara "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

Marcar abreviatures pot proporcionar informació útil als navegadors, sistemes de traducció i motors de cerca. Exemple:

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

HTML `<address>` per a la informació de contacte

L'etiqueta HTML `<address>` defineix la informació de contacte de l'autor/propietari d'un document o d'un article.

La informació de contacte pot ser una adreça de correu electrònic, URL, adreça física, número de telèfon, identificador de xarxes socials, etc.

El text de l'element `<address>` normalment es mostra en *cursiva* i els navegadors sempre afegiran un salt de línia abans i després de l'element `<address>`. Exemple:

```
<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

HTML `<cite>` per al títol del treball

L'etiqueta `<cite>` HTML defineix el títol d'una obra creativa (per exemple, un llibre, un poema, una cançó, una pel·lícula, una pintura, una escultura, etc.).

Nota: el nom d'una persona no és el títol d'una obra.

El text de l'element `<cite>` normalment es mostra en *cursiva*. Exemple:

```
<p><cite>The Scream</cite>by Edvard Munch. Painted in 1893.</p>
```

HTML `<bdo>` per a la substitució bidireccional

BDO són les sigles de Bi-Directional Override.

L'etiqueta HTML `<bdo>` s'utilitza per anul·lar la direcció del text actual. Exemple:

```
<bdo dir="rtl">This text will be written from right to left</bdo>
```

10. Comentaris HTML

Els comentaris HTML no es mostren al navegador, però poden ajudar a documentar el vostre codi font HTML.

Etiqueta de comentari HTML

Podeu afegir comentaris a la vostra font HTML utilitzant la sintaxi següent:

```
<!-- Write your comments here -->
```

Tingueu en compte que hi ha un signe d'exclamació (!) a l'etiqueta inicial, però no a l'etiqueta final.

Nota: el navegador no mostra els comentaris, però poden ajudar a documentar el vostre codi font HTML.

Afegeix comentaris

Amb els comentaris podeu col·locar notificacions i recordatoris al vostre codi HTML:

```
<!-- This is a comment -->

<p>This is a paragraph.</p>

<!-- Remember to add more information here -->
```

Amaga contingut

Els comentaris es poden utilitzar per ocultar contingut. Això pot ser útil si amagueu el contingut temporalment.

També podeu amagar més d'una línia. Tot el que hi ha entre el `<!--` i el `-->` s'amagarà de la pantalla.

Els comentaris també són excel·lents per depurar HTML, perquè podeu comentar les línies de codi HTML, una a la vegada, per cercar errors.

Amaga el contingut en línia

Els comentaris es poden utilitzar per ocultar parts al mig del codi HTML.

11. Colors HTML

Els colors HTML s'especifiquen amb noms de color predefinits o amb valors RGB, HEX, HSL, RGBA o HSLA.

Noms de colors

En HTML, es pot especificar un color mitjançant un nom de color:

https://www.w3schools.com/colors/colors_names.asp

HTML admet [140 noms de colors estàndard](#).

Color de fons

Podeu establir el color de fons per als elements HTML:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

Color del text

Podeu definir el color del text:

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Color de la vora

Podeu definir el color de les vores:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Valors de color

En HTML, els colors també es poden especificar mitjançant valors RGB, valors HEX, valors HSL, valors RGBA i valors HSLA. Exemple:

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

Valors de color RGB

En HTML, un color es pot especificar com a valor RGB, utilitzant aquesta fórmula:

rgb (vermell, verd, blau)

Cada paràmetre (vermell, verd i blau) defineix la intensitat del color amb un valor entre 0 i 255.

Això vol dir que hi ha $256 \times 256 \times 256 = 16777216$ colors possibles!

Per exemple, `rgb(255, 0, 0)` es mostra en roig, perquè el roig s'estableix en el seu valor més alt (255) i els altres dos (verd i blau) es defineixen en 0.

Un altre exemple, `rgb(0, 255, 0)` es mostra com a verd, perquè el verd s'estableix en el seu valor més alt (255) i els altres dos (vermell i blau) es defineixen en 0.

Per mostrar el negre, establiu tots els paràmetres de color a 0, com aquest: `rgb(0, 0, 0)`.

Per mostrar el blanc, establiu tots els paràmetres de color a 255, com aquest: `rgb(255, 255, 255)`.

Valors de color HEX

En HTML, es pot especificar un color mitjançant un valor hexadecimal de la forma:

#rrggbb

On rr (roig), gg (verd) i bb (blau) són valors hexadecimals entre 00 i ff (igual que el decimal 0-255).

Per exemple, `#ff0000` es mostra en roig, perquè el roig s'estableix en el seu valor més alt (ff) i els altres dos (verd i blau) es defineixen en 00.

Un altre exemple, `#00ff00` es mostra com a verd, perquè el verd s'estableix al seu valor més alt (ff) i els altres dos (vermell i blau) es defineixen en 00.

Per mostrar el negre, configureu tots els paràmetres de color a 00, com aquest: `#000000`.

Per mostrar el blanc, configureu tots els paràmetres de color a ff, com aquest: `#ffffff`

12. Estils HTML - CSS

CSS són les sigles de Cascading Style Sheets. CSS estalvia molta feina. Pot controlar el disseny de diverses pàgines web alhora.

Què és CSS?

Els fulls d'estil en cascada (CSS) s'utilitzen per donar format al disseny d'una pàgina web.

Amb CSS, podeu controlar el color, la font, la mida del text, l'espaiat entre els elements, la posició i la disposició dels elements, quines imatges de fons o colors de fons s'han d'utilitzar, diferents pantalles per a diferents dispositius i mides de pantalla, i molt més!

Consell: la paraula **cascada** significa que un estil aplicat a un element pare també s'aplicarà a tots els elements secundaris dins del pare. Per tant, si configureu el color del text del cos com a "blau", tots els encapçalaments, paràgrafs i altres elements de text del cos també tindran el mateix color (tret que especifiqueu alguna cosa més)!

Utilitzant CSS

El CSS es pot afegir als documents HTML de 3 maneres:

- **En línia:** utilitzant l'atribut `style` dins dels elements HTML
- **Interna:** utilitzant un element `<style>` a la `<head>` secció
- **Extern:** utilitzant un element `<link>` per enllaçar a un fitxer CSS extern

La forma més habitual d'afegir CSS és mantenir els estils en fitxers CSS externs. No obstant això, en aquest tutorial farem servir estils en línia i interns, perquè això és més fàcil de mostrar i més fàcil de provar-ho tu mateix.

CSS en línia

Un CSS en línia s'utilitza per aplicar un estil únic a un únic element HTML.

Un CSS en línia utilitza l'atribut `style` d'un element HTML.

L'exemple següent estableix el color del text de l'element `<h1>` en blau i el color del text de l'element `<p>` en roig. Exemple:

```
<h1 style="color:blue;">A Blue Heading</h1>
<p style="color:red;">A red paragraph.</p>
```

CSS intern

S'utilitza un CSS intern per definir un estil per a una única pàgina HTML.

Un CSS intern es defineix a la secció `<head>` d'una pàgina HTML, dins d'un element `<style>`.

L'exemple següent estableix el color del text de TOTS els elements `<h1>` (d'aquesta pàgina) en blau i el color del text de TOTS els elements `<p>` en roig. A més, la pàgina es mostrarà amb un color de fons "powderblue":

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body{ background-color: powderblue;}
    h1{color: blue;}
    p{color: red;}
  </style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

CSS extern

S'utilitza un full d'estil extern per definir l'estil de moltes pàgines HTML.

Per utilitzar un full d'estil extern, afegiu un element `<link>` a la secció `<head>` de cada pàgina HTML. Exemple:

```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>

```

El full d'estil extern es pot escriure en qualsevol editor de text. El fitxer no ha de contenir cap codi HTML i s'ha de desar amb una extensió .css. A continuació, es mostra el fitxer "styles.css":

```

body{background-color: powderblue;}
h1{color: blue;}
p{color: red;}

```

Consell: amb un full d'estil extern, podeu canviar l'aspecte d'un lloc web sencer canviant un fitxer!

Colors, tipus de lletra i mides CSS

Ací, demostrarem algunes propietats CSS d'ús habitual. Més endavant aprendràs més sobre ells.

La propietat CSS **color** defineix el color del text que s'utilitzarà.

La propietat CSS **font-family** defineix el tipus de lletra que s'utilitzarà.

La propietat CSS **font-size** defineix la mida del text que s'utilitzarà.

```

<!DOCTYPE html>
<html>
<head>
<style>
h1{
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p{
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>

```

Border CSS

La propietat CSS **border** defineix una vora al voltant d'un element HTML. Exemple:

```
p{border: 2px solid powderblue;}
```

Padding CSS

La propietat **padding** CSS defineix un farciment (espai) entre el text i la vora. Exemple:

```
p{border: 2px solid powderblue; padding: 30px;}
```

Margin CSS

La propietat CSS **margin** defineix un marge (espai) fora de la vora. Exemple:

```
p{border: 2px solid powderblue; margin: 50px;}
```

13. Enllaços HTML

Els enllaços es troben en totes les pàgines web. Els enllaços permeten als usuaris fer clic d'una pàgina a una altra.

Enllaços HTML - Hiperenllaços

Els enllaços HTML són hiperenllaços. Podeu fer clic a un enllaç i anar a un altre document. Quan moveu el ratolí per sobre d'un enllaç, la fletxa del ratolí es convertirà en una xicoteta mà.

Nota: un enllaç no només ha de ser de text. Un enllaç pot ser una imatge o qualsevol altre element HTML!

Enllaços HTML - Sintaxi

L'etiqueta HTML **<a>** defineix un hiperenllaç. Té la sintaxi següent:

```
<a href="url">Link text</a>
```

L'atribut més important de l'element **<a>** és l'atribut **href**, que indica la destinació de l'enllaç. El text de l'enllaç és la part que serà visible per al lector.

Si feu clic al text de l'enllaç, s'enviarà el lector a l'adreça URL especificada. Exemple:

```
<a href="https://www.w3schools.com/">Visit W3Schools!</a>
```

De manera predeterminada, els enllaços apareixeran de la manera següent a tots els navegadors:

- Un enllaç no visitat està subratllat i blau
- Un enllaç visitat està subratllat i lila
- Un enllaç actiu està subratllat i roig

Consell: per descomptat, els enllaços es poden dissenyar amb CSS, per a obtenir un altre aspecte!

Enllaços HTML: l'atribut target

Per defecte, la pàgina enllaçada es mostrarà a la finestra del navegador actual. Per canviar-ho, heu d'especificar un altre objectiu per a l'enllaç.

L'atribut **target** especifica on s'ha d'obrir el document enllaçat. Pot tenir un dels valors següents:

- **_self**- Per defecte. Obre el document a la mateixa finestra/pestanya en què es va fer clic
- **_blank**- Obre el document en una finestra o pestanya nova
- **_parent**- Obre el document al marc principal
- **_top**- Obre el document a tot el cos de la finestra

Utilitzeu `target="_blank"` per obrir el document enllaçat en una nova finestra o pestanya del navegador:

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

URL absoluts versus URL relatius

Els dos exemples anteriors utilitzen un **URL absolut** (una adreça web completa) a l'atribut `href`.

S'especifica un enllaç local (un enllaç a una pàgina dins del mateix lloc web) amb un **URL relatiu** (sense la part "<https://www>"). Exemple:

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>
<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

Enllaços HTML: utilitzeu una imatge com a enllaç

Per utilitzar una imatge com a enllaç, només cal posar l'etiqueta `` dins de l'etiqueta `<a>`. Exemple:

```
<a href="default.asp"></a>
```

Enllaç a una adreça de correu electrònic

Utilitzeu `mailto:` dins de l'atribut `href` per crear un enllaç que obliga el programa de correu electrònic de l'usuari (per permetre que s'envie un correu electrònic nou):

```
<a href="mailto:someone@example.com">Send email</a>
```

Botó com a enllaç

Per utilitzar un botó HTML com a enllaç, heu d'afegir algun codi JavaScript.

JavaScript us permet especificar què passa en determinats esdeveniments, com ara un clic en un botó. Exemple:

```
<button onclick="document.location='default.asp'">HTML Tutorial</button>
```

Títols d'enllaç

L'atribut `title` especifica informació addicional sobre un element. La informació es mostra més sovint com a text d'informació eines quan el ratolí es mou sobre un element:

```
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML
section">Visit our HTML Tutorial</a>
```

Enllaços HTML : diferents colors

Un enllaç HTML es mostra amb un color diferent en funció de si s'ha visitat, no visitat o està actiu. Però també podeu canviar els colors de l'enllaç mitjançant CSS:

Ací un exemple: un enllaç no visitat serà de color verd sense subratllat. Un enllaç visitat serà de color rosa sense subratllat. Un enllaç actiu serà groc i subratllat. A més, quan passeu el ratolí per sobre d'un enllaç (a:hover) es tornarà roig i subratllat:

```
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}

a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}

a:hover {
  color: red;
  background-color: transparent;
  text-decoration: underline;
}

a:active {
  color: yellow;
  background-color: transparent;
  text-decoration: underline;
}
</style>
```

Crea adreces d'interès

Els enllaços HTML es poden utilitzar per crear adreces d'interès, de manera que els lectors puguin anar a parts específiques d'una pàgina web.

Creeu un marcador en HTML. Els marcadors poden ser útils si una pàgina web és molt llarga.

Per crear un marcador: primer creeu el marcador i, a continuació, afegiu-hi un enllaç.

Quan es fa clic a l'enllaç, la pàgina es desplaçarà cap avall o cap amunt fins a la ubicació amb el marcador. Exemple:

Primer, utilitzeu l'atribut `id` per crear un marcador:

```
<h2 id="C4">Chapter 4</h2>
```

A continuació, afegiu un enllaç al marcador ("Va al capítol 4") des de la mateixa pàgina:

```
<a href="#C4">Jump to Chapter 4</a>
```

També podeu afegir un enllaç a un marcador d'una altra pàgina:

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```

14. Imatges HTML

Les imatges poden millorar el disseny i l'aparença d'una pàgina web. Exemple:

```

```

Sintaxi d'imatges HTML

L'etiqueta HTML `` s'utilitza per incrustar una imatge en una pàgina web.

Les imatges no s'insereixen tècnicament en una pàgina web. Les imatges estan enllaçades a pàgines web. L'etiqueta `` crea un espai de retenció per a la imatge de referència.

L'etiqueta `` està buida, només conté atributs i no té cap etiqueta de tancament.

L'etiqueta `` té dos atributs obligatoris:

- `src` - Especifica el camí a la imatge
- `alt` - Especifica un text alternatiu per a la imatge

Sintaxi:

```

```

L'atribut `src`

L'atribut `src` obligatori especifica el camí (URL) a la imatge.

Nota: Quan es carrega una pàgina web, és el navegador, en aquest moment, qui rep la imatge d'un servidor web i l'insereix a la pàgina. Per tant, assegureu-vos que la imatge es mantinga al mateix lloc en relació a la pàgina web, en cas contrari, els vostres visitants rebran una icona d'enllaç trencat. La icona d'enllaç trencat i el `alt` text es mostren si el navegador no troba la imatge.

L'atribut `alt`

L'atribut `alt` obligatori proporciona un text alternatiu per a una imatge, si l'usuari per algun motiu no la pot visualitzar (a causa d'una connexió lenta, un error a l'atribut `src` o si l'usuari utilitza un lector de pantalla).

El valor de l'atribut `alt` ha de descriure la imatge: Si un navegador no pot trobar una imatge, mostrarà el valor de l'atribut `alt`.

Mida de la imatge: amplada i alçada

Podeu utilitzar l'atribut `style` per especificar l'amplada i l'alçada d'una imatge. Exemple:

```

```

Alternativament, podeu utilitzar els atributs `width` i `height`:

```

```

Imatges en una altra carpeta

Si teniu les vostres imatges en una subcarpeta, heu d'incloure el nom de la carpeta a l'atribut `src`. Exemple:

```

```

Imatges en un altre servidor/lloc web

Alguns llocs web apunten a una imatge en un altre servidor.

Per apuntar a una imatge en un altre servidor, heu d'especificar un URL absolut (complet) a l'atribut `src`. Exemple:


```

```

Imatges animades

HTML permet GIF animats:

Imatge com a enllaç

Per utilitzar una imatge com a enllaç, poseu l'etiqueta `` dins de l'etiqueta `<a>`:

```
<a href="exemple.html">
  
</a>
```

Imatge flotant

Utilitzeu la propietat CSS `float` per deixar que la imatge flote a la dreta o a l'esquerra d'un text. Exemple:

```
<p>
The image will float to the right of the text.</p>
<p>
The image will float to the left of the text.</p>
```

Imatges de fons HTML

Es pot especificar una imatge de fons per a quasi qualsevol element HTML.

Imatge de fons en un element HTML

Per afegir una imatge de fons a un element HTML, utilitzeu l'atribut HTML `style` i la propietat CSS `background-image`. Afegiu una imatge de fons a un element HTML:

```
<p style="background-image: url('img_girl.jpg');">
```

Imatge de fons en una pàgina

Si voleu que tota la pàgina tinga una imatge de fons, heu d'especificar la imatge de fons a l'element `<body>`. Afegiu una imatge de fons per a tota la pàgina:

```
<style>
body {
  background-image: url('img_girl.jpg');
}
</style>
```

Repetició de fons

Si la imatge de fons és més menuda que l'element, la imatge es repetirà, horitzontalment i verticalment, fins que arribi al final de l'element. Per a evitar que la imatge de fons es repeteixca, establiu la propietat `background-repeat` a `no-repeat`:

```
<style>
body {
  background-image: url('example_img_girl.jpg');
  background-repeat: no-repeat;
}
</style>
```

Portada de fons

Si voleu que la imatge de fons cobreisca tot l'element, podeu establir la propietat `background-size` a `cover`.

A més, per assegurar-vos que tot l'element estiga sempre cobert, configureu la propietat `background-attachment` a `fixed`:

D'aquesta manera, la imatge de fons cobrirà tot l'element, sense estirar-se (la imatge mantindrà les seues proporcions originals). Exemple:

```
<style>
body {
  background-image: url('img_girl.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
</style>
```

Estirament de fons

Si voleu que la imatge de fons s'estire per adaptar-se a tot l'element, podeu establir la `background-size` propietat a `100% 100%`.

Proveu de canviar la mida de la finestra del navegador i veureu que la imatge s'estirarà, però sempre cobreix tot l'element. Exemple:

```
<style>
body {
  background-image: url('img_girl.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: 100% 100%;
}
</style>
```

Element HTML <picture>

L'element HTML `<picture>` us permet mostrar imatges diferents per a diferents dispositius o mides de pantalla. L'element HTML `<picture>` ofereix als desenvolupadors web més flexibilitat a l'hora d'especificar recursos d'imatge.

L'element `<picture>` conté un o més `<source>` elements, cadascun referint-se a diferents imatges a través de l'atribut `srcset`. D'aquesta manera, el navegador pot triar la imatge que millor s'adapte a la vista i/o dispositiu actuals.

Cada element `<source>` té un `media` atribut que defineix quan la imatge és la més adequada. Exemple:

```
<picture>
  <source media="(min-width: 650px)" srcset="img_food.jpg">
  <source media="(min-width: 465px)" srcset="img_car.jpg">
  
</picture>
```

Nota: especifiqueu sempre un element `` com a darrer element fill de l'element `<picture>`. L'element `` l'utilitzen els navegadors que no admeten l'element `<picture>` o si cap de les etiquetes `<source>` coincideix.

Quan utilitzar l'element d'picture

Hi ha dos propòsits principals per a l'element `<picture>`:

1. Ample de banda

Si teniu una pantalla o un dispositiu xicotet, no cal carregar un fitxer d'imatge gran. El navegador utilitzarà el primer element `<source>` amb valors d'atribut coincidents i ignorarà qualsevol dels elements següents.

2. Suport de format

És possible que alguns navegadors o dispositius no admeten tots els formats d'imatge. En utilitzar l'element `<picture>`, podeu afegir imatges de tots els formats i el navegador utilitzarà el primer format que reconega i ignorarà qualsevol dels elements següents. El navegador utilitzarà el primer format d'imatge que reconeix:

```
<picture>
  <source srcset="img_avatar.png">
  <source srcset="img_girl.jpg">
  
</picture>
```

Nota: el navegador utilitzarà el primer element `<source>` amb valors d'atribut coincidents i ignorarà els elements `<source>` següents.

15. Favicon HTML

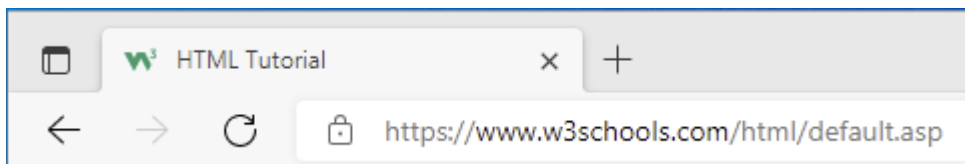
Un favicon és una xicoteta imatge que es mostra al costat del títol de la pàgina a la pestanya del navegador.

Com afegir un favicon en HTML

Podeu utilitzar qualsevol imatge que us agrade com a favicon. També podeu crear el vostre propi favicon a llocs com <https://www.favicon.cc>.

Consell: un favicon és una imatge menuda, de manera que hauria de ser una imatge senzilla amb un alt contrast.

Es mostra una imatge de favicon a l'esquerra del títol de la pàgina a la pestanya del navegador, com aquesta:



Per afegir un favicon al vostre lloc web, deseu la vostra imatge de favicon al directori arrel del vostre servidor web o creeu una carpeta al directori arrel anomenada imatges i deseu la vostra imatge de favicon en aquesta carpeta. Un nom comú per a una imatge de favicon és "favicon.ico".

A continuació, afegiu un element `<link>` al fitxer "index.html", després de l'element `<title>`, com aquest:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
```

```
</head>
<body>
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

16. Títol de la pàgina HTML

Cada pàgina web hauria de tenir un títol per descriure el significat de la pàgina. L'element `<title>` afegeix un títol a la vostra pàgina:

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Tutorial</title>
</head>
<body>
  The content of the document.....
</body>
</html>
```

El títol es mostra a la barra de títol del navegador. El títol ha de descriure el contingut i el significat de la pàgina.

El títol de la pàgina és molt important per a l'optimització de motors de cerca (SEO). Els algorismes dels motors de cerca utilitzen el text per decidir l'ordre quan s'enumeren pàgines als resultats de la cerca.

L'element `<title>`:

- defineix un títol a la barra d'eines del navegador
- proporciona un títol per a la pàgina quan s'afegeix als preferits
- mostra un títol per a la pàgina als resultats del motor de cerca

Per tant, intenta que el títol siga com més precís i significatiu millor!

17. Taules HTML

Les taules HTML permeten als desenvolupadors web organitzar les dades en files i columnes. Exemple:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada

Definiu una taula HTML

Una taula en HTML consta de cel·les de taula dins de files i columnes. Una taula HTML senzilla:

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

Cel·les de la taula

Cada cel·la de la taula està definida per una `<td>` i una etiqueta `</td>`. `td` representa les dades de la taula.

Nota: una cel·la de taula pot contenir tot tipus d'elements HTML: text, imatges, llistes, enllaços, altres taules, etc.

Files de taula

Cada fila de la taula comença amb una `<tr>` i acaba amb una etiqueta `</tr>`. `tr` significa fila de taula. Exemple:

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

Podeu tenir tantes files com vulgueu en una taula. Només assegureu-vos que el nombre de cel·les siga el mateix a cada fila.

Capçaleres de la taula

De vegades voleu que les vostres cel·les siguin cel·les de capçalera de taula. En aquests casos, utilitzeu l'etiqueta `<th>` en lloc de l'etiqueta `<td>`. `th` representa la capçalera de la taula. Exemple:

```

<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>

```

De manera predeterminada, el text dels elements `<th>` està en negreta i centrat, però ho podeu canviar amb CSS.

Etiquetes de taula HTML

Tag	Description
<code><table></code>	Defines a table
<code><th></code>	Defines a header cell in a table
<code><tr></code>	Defines a row in a table
<code><td></code>	Defines a cell in a table
<code><caption></code>	Defines a table caption
<code><colgroup></code>	Specifies a group of one or more columns in a table for formatting
<code><col></code>	Specifies column properties for each column within a <code><colgroup></code> element
<code><thead></code>	Groups the header content in a table
<code><tbody></code>	Groups the body content in a table
<code><tfoot></code>	Groups the footer content in a table

Vores de la taula HTML

Les taules HTML poden tenir vores de diferents estils i formes.

Com afegir una vora

Per afegir una vora, utilitzeu la propietat `border` CSS als elements `table`, `th`, i `td`:


```

table, th, td {
  border: 1px solid black;
}

```

Vores de taula col·lapsades

Per evitar tenir vores dobles com a l'exemple anterior, establiu la `border-collapse` propietat CSS a `collapse`.

Això farà que les vores col·lapsen en una única vora:


```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
```

Vores de taula amb estil

Si configureu un color de fons per a cada cella i doneu a la vora un color blanc (el mateix que el fons del document), obtindreu la impressió d'una vora invisible:


```
table, th, td {
  border: 1px solid white;
  border-collapse: collapse;
}
th, td {
  background-color: #96D4D4;
}
```

Vores de la taula rodones

Amb la propietat `border-radius`, les vores tenen cantonades arrodonides:


```
table, th, td {
  border: 1px solid black;
  border-radius: 10px;
}
```

Ometeu la vora de la taula deixant de banda el selector CSS `table`:


```
th, td {
  border: 1px solid black;
  border-radius: 10px;
}
```

Vores de taula amb punts

Amb la propietat `border-style`, podeu definir l'aspecte de la vora.

Es permeten els valors següents:

- dotted
- dashed
- solid
- double
- groove
- ridge
- inset
- outset
- none
- hidden

```
th, td {  
  border-style: dotted;  
}
```

Color de la vora

Amb la propietat **border-color**, podeu establir el color de la vora.



```
th, td {  
  border-color: #96D4D4;  
}
```

Mides de la taula HTML

Les taules HTML poden tenir diferents mides per a cada columna, fila o tota la taula.

Utilitzeu l'atribut **style** amb les propietats **width** o **height** per especificar la mida d'una taula, fila o columna.

Amplada de la taula HTML

Per establir l'amplada d'una taula, afegiu l'atribut **style** a l'element **<table>**. Estableix l'amplada de la taula al 100%:

Nota: L'ús d'un percentatge com a unitat de mida d'una amplada significa quina amplada es comparará aquest element amb el seu element pare, que en aquest cas és l'element **<body>**.

Amplada de la columna de la taula HTML

Per establir la mida d'una columna específica, afegiu l'atribut **style** a un element **<th>** o **<td>**. Exemple: Estableix l'amplada de la primera columna al 70% i també exemple d'amplada 100% de la taula:

```
<!DOCTYPE html>  
<html>  
  <style>  
    table, th, td {  
      border: 1px solid black;  
      border-collapse: collapse;  
    }  
  </style>  
<body>  
  
  <h2>Set the first column to 70% of the table width</h2>  
  
  <table style="width:100%">  
    <tr>  
      <th style="width:70%">Firstname</th>
```



```

        <th>Lastname</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>50</td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>94</td>
    </tr>
</table>
</body>
</html>

```

Alçada de fila de la taula HTML

Per establir l'alçada d'una fila específica, afegiu l'atribut **style** a un element de fila de taula:

Exemple: Estableix l'alçada de la segona fila a 200 píxels.

```

<table style="width:100%">
  <tr style="height:200px">
    <th style="width:70%">Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
</table>

```

Capçaleres HTML

Les taules HTML poden tenir capçaleres per a cada columna o fila, o per a moltes columnes/files.

EMIL	TOBIES	LINUS

8:00		
9:00		
10:00		
11:00		
12:00		
13:00		

	EL MEU	DIMAR	DIM	COL·LECCIONAR	DIV
8:00					
9:00					
10:00					
11:00					
12:00					

DESEMBRE		

Capçaleres de taula HTML

Les capçaleres de les taules es defineixen amb elements **th**. Cada element **th** representa una cel·la de taula.

Capçaleres de taules verticals

Per utilitzar la primera columna com a capçaleres de taula, definiu la primera cel·la de cada fila com a element **<th>**:

Alinea les capçaleres de la taula

De manera predeterminada, les capçaleres de la taula estan en negreta i centrades:

Nom	Cognom	Edat
Jill	Smith	50

Per alinear a l'esquerra les capçaleres de la taula, utilitzeu la propietat CSS `text-align`:

```
th {  
  text-align: left;  
}
```

Capçalera per a diverses columnes

Podeu tenir una capçalera que abaste dues o més columnes.

Nom		Edat
Jill	Smith	50
Vigília	Jackson	94

Per fer-ho, utilitzeu l'atribut `colspan` de l'element `<th>`. Exemple:

```
<table>  
  <tr>  
    <th colspan="2">Name</th>  
    <th>Age</th>  
  </tr>  
  <tr>  
    <td>Jill</td>  
    <td>Smith</td>  
    <td>50</td>  
  </tr>  
</table>
```

Títol de la taula

Podeu afegir un títol que serveisca d'encapçalament per a tota la taula.

Estalvi mensual

Mes	Estalvis
gener	\$100

Per afegir un títol a una taula, utilitzeu l'etiqueta `<caption>`. Exemple:

```
<table style="width:100%">  
  <caption>Estalvi mensual</caption>  
  <tr>  
    <th>Month</th><th>Savings</th>  
  </tr>  
  <tr>  
    <td>January</td><td>$100</td>  
  </tr>  
</table>
```

Emplenat i espaiat de la taula HTML

Les taules HTML poden ajustar el farciment, i també l'espai entre les cel·les.

Amb farciment

Hola	Hola	Hola
Hola	Hola	Hola
Hola	Hola	Hola

Amb espaiat

Hola	Hola	Hola
Hola	Hola	Hola
Hola	Hola	Hola

Taula HTML: farciment de cel·les

El farciment de la cella és l'espai entre les vores de la cella i el contingut de la cella. Per defecte, el farciment s'estableix a 0. Per afegir farciment a les cel·les de la taula, utilitzeu la propietat **padding** CSS. Exemple:

```
th, td {  
  padding: 15px;  
}
```

Per afegir farciment només per dalt del contingut, utilitzeu la propietat **padding-top**.

I els altres costats amb les propietats **padding-bottom**, **padding-left**, i **padding-right**:

```
th, td {  
  padding-top: 10px;  
  padding-bottom: 20px;  
  padding-left: 30px;  
  padding-right: 40px;  
}
```

Taula HTML - Espaiat de cel·les

L'espai entre cel·les és l'espai entre cada cella. Per defecte, l'espai està configurat en 2 píxels. Per canviar l'espai entre cel·les de la taula, utilitzeu la propietat CSS **border-spacing** de l'element: **table**. Exemple:

```
table {  
  border-spacing: 30px;  
}
```

Taula HTML Colspan i Rowspan

Les taules HTML poden tenir cel·les que s'estenen per diverses files i/o columnes.

NOM		

ABRIL		

2022		
FIESTA		

Taula HTML - Colspan

Per fer que una cella s'estenga en diverses columnes, utilitzeu l'atribut **colspan** (ja vist).

Taula HTML - Cowspan

Per fer que una cella s'estenga en diverses files, utilitzeu l'atribut **rowspan**:

```
<!DOCTYPE html>  
<html>  
<head>
```

```

<style>
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
</style>
</head>
<body>

<h2>Cell that spans two rows</h2>
<p>To make a cell span more than one row, use the rowspan attribute.</p>

<table style="width:100%">
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
  </tr>
</table>
</body>
</html>

```

Estil de taula HTML

Utilitzeu CSS per fer que les vostres taules es vegen millor.

Taula HTML - Zebra Stripes

Si afegiu un color de fons a ina fila sí i altra no de la taula, obtindreu un bon efecte de ratlles de zebra.

Per estilitzar tots els altres elements de fila de la taula, utilitzeu el **:nth-child(even)** selector com aquest:

```

<!DOCTYPE html>
<html>
<head>
  <style>
    table {
      border-collapse: collapse;
      width: 100%;
    }
    th, td {
      text-align: left;
      padding: 8px;
    }

    tr:nth-child(even) {
      background-color: #D6EEEE;
    }
  </style>
</head>
<body>
<h2>Zebra Striped Table</h2>
<p>For zebra-striped tables, use the nth-child() selector and add a
background-color to all even (or odd) table rows:</p>
<table>
  <tr>

```

```

        <th>First Name</th>
        <th>Last Name</th>
        <th>Points</th>
    </tr>
    <tr>
        <td>Peter</td>
        <td>Griffin</td>
        <td>$100</td>
    </tr>
    <tr>
        <td>Lois</td>
        <td>Griffin</td>
        <td>$150</td>
    </tr>
</table>
</body>
</html>

```

Zebra Striped Table

For zebra-striped tables, use the `nth-child()` selector and add a `background-color` to all even (or odd) table rows:

First Name	Last Name	Points
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

Nota: si feu servir **(odd)** en comptes de **(even)**, l'estil es produirà a la fila 1, 3, 5, etc. en lloc de 2, 4, 6, etc.

Taula HTML: ratlles verticals de zebra

Per fer ratlles de zebra verticals, estileu totes les altres columnes , en lloc de totes les altres files .

Estableix els elements **:nth-child(even)** de dades de la taula com aquest:

```

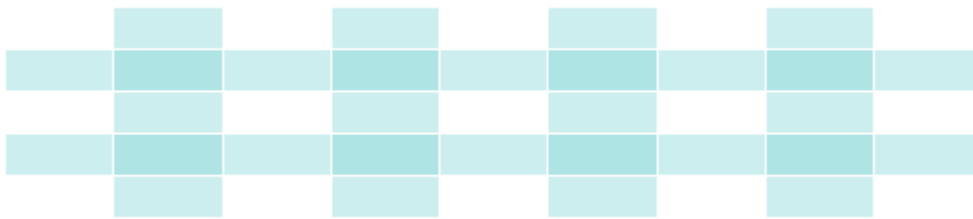
td:nth-child(even), th:nth-child(even) {
    background-color: #D6EEEE;
}

```

Nota: col·loqueu el **:nth-child()** selector a tots dos **th** i **td** als elements si voleu tindre l'estil tant a les capçaleres com a les cel·les normals de la taula.

Combina ratlles de zebra verticals i horitzontals

Podeu combinar l'estil dels dos exemples anteriors i tindreu ratlles a cada altra fila i a cada altra columna.



Si utilitzeu un color transparent obtindreu un efecte de superposició.

Utilitzeu un `rgba()` color per especificar la transparència del color. Exemple:

```
tr:nth-child(even) {  
  background-color: rgba(150, 212, 212, 0.4);  
}  
  
th:nth-child(even),td:nth-child(even) {  
  background-color: rgba(150, 212, 212, 0.4);  
}
```

Divisors horitzontals

Nom	Cognom	Estalvis
Pere	Grifó	\$100
Lois	Grifó	\$150
Joe	Swanson	\$300

Si especifiqueu vores només a la part inferior de cada fila de la taula, tindreu una taula amb divisors horitzontals.

Afegiu la propietat `border-bottom` a tots els elements `<tr>` per obtenir divisors horitzontals. Exemple:

```
tr {  
  border-bottom: 1px solid #ddd;  
}
```

Taula Hoverable

Utilitzeu el selector `:hover` per essaltar les files de la taula sobre el ratolí. Exemple:

```
tr:hover {background-color: #D6EEEE;}
```

18. Llistes HTML

Les llistes HTML permeten als desenvolupadors web agrupar un conjunt d'elements relacionats en llistes.

Exemple

Una llista HTML no ordenada:

- Article
- Article
- Article
- Article

Una llista HTML ordenada:

1. Primer element
2. Segon article
3. Tercer element
4. Quart element

Llista HTML no ordenada

Una llista no ordenada comença amb l'etiqueta ``. Cada element de la llista comença amb l'etiqueta ``.

Els elements de la llista es marcaran amb vinyetes (xicotets cercles negres) per defecte:

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Llista HTML ordenada

Una llista ordenada comença amb l'etiqueta ``. Cada element de la llista comença amb l'etiqueta ``.

Els elements de la llista es marcaran amb números per defecte:

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Llistes de descripció HTML

HTML també admet llistes de descripcions.

Una llista de descripció és una llista de termes, amb una descripció de cada terme.

L'etiqueta `<dl>` defineix la llista de descripció, l'etiqueta `<dt>` defineix el terme (nom) i l'etiqueta `<dd>` descriu cada terme:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Etiquetes de llista HTML

Tag Description

<code></code>	Defines an unordered list
<code></code>	Defines an ordered list
<code></code>	Defines a list item
<code><dl></code>	Defines a description list
<code><dt></code>	Defines a term in a description list
<code><dd></code>	Describes the term in a description list

19. Bloc HTML i elements en línia

Cada element HTML té un valor de visualització per defecte, depenent del tipus d'element que siga. Hi ha dos valors de visualització: block i inline.

Elements a nivell de bloc

Un element a nivell de bloc sempre comença en una línia nova i els navegadors afegeixen automàticament algun espai (un marge) abans i després de l'element.

Un element a nivell de bloc sempre ocupa tota l'amplada disponible (s'estén cap a l'esquerra i la dreta tant com pot).

Dos elements de bloc utilitzats habitualment són: `<p>` i `<div>`.

L'element `<p>` defineix un paràgraf en un document HTML.

L'element `<div>` defineix una divisió o una secció en un document HTML. Exemple:

```
<p>Hello World</p>
<div>Hello World</div>
```

Aquests són els elements a nivell de bloc en HTML:

<code><address></code>	<code><article></code>	<code><aside></code>	<code><blockquote></code>	<code><canvas></code>
<code><dd></code>	<code><div></code>	<code><dl></code>	<code><dt></code>	<code><fieldset></code>
<code><figcaption></code>	<code><figure></code>	<code><footer></code>	<code><form></code>	<code><h1>-<h6></code>
<code><header></code>	<code><hr></code>	<code></code>	<code><main></code>	<code><nav></code>
<code><noscript></code>	<code></code>	<code><p></code>	<code><pre></code>	<code><section></code>
<code><table></code>	<code><tfoot></code>	<code></code>	<code><video></code>	

Elements en línia

Un element en línia no comença en una línia nova.

Un element en línia només ocupa l'amplada necessària.

Aquest és un element `` dins d' un paràgraf. Exemple:

```
<!DOCTYPE html>
<html>
<body>
<p>This is an inline span <span style="border: 1px solid black">Hello
World</span> element inside a paragraph.</p>
<p>The SPAN element is an inline element, and will not start on a new line
and only takes up as much width as necessary.</p>
</body>
</html>
```

Aquests són els elements en línia en HTML:

<code><a></code>	<code><abbr></code>	<code><acronym></code>	<code></code>	<code><bdo></code>
<code><big></code>	<code>
</code>	<code><button></code>	<code><cite></code>	<code><code></code>
<code><dfn></code>	<code></code>	<code><i></code>	<code></code>	<code><input></code>
<code><kbd></code>	<code><label></code>	<code><map></code>	<code><object></code>	<code><output></code>
<code><q></code>	<code><samp></code>	<code><script></code>	<code><select></code>	<code><small></code>
<code></code>	<code></code>	<code><sub></code>	<code><sup></code>	<code><textarea></code>
<code><time></code>	<code><tt></code>	<code><var></code>		

20. Atribut de classe

L'atribut HTML **class** s'utilitza per especificar una classe per a un element HTML.

Diversos elements HTML poden compartir la mateixa classe.

Ús de l'atribut de classe

L'atribut **class** s'utilitza normalment per assenyalar un nom de classe en un full d'estil. També pot ser utilitzat per un JavaScript per accedir i manipular elements amb el nom de classe específic.

En l'exemple següent tenim tres **<div>** elements amb un atribut **class** amb el valor de "ciutat". Tots els tres **<div>** elements tindran un estil igual d'acord amb la definició d'estil a la secció **.city** de capçalera. Exemple:

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
<h2>London</h2>
<p>London is the capital of England.</p>
</div>

<div class="city">
<h2>Paris</h2>
<p>Paris is the capital of France.</p>
</div>

<div class="city">
<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

En l'exemple següent tenim dos elements **** amb un atribut **class** amb el valor de "nota". Els dos elements **** tindran un estil igual d'acord amb la definició d'estil **.note** a la secció de capçalera. Exemple:

```
<!DOCTYPE html>
<html>
<head>
<style>
.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>
```

```
<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```

L'atribut **class** es pot utilitzar en **qualsevol** element HTML. El nom de la classe distingeix entre majúscules i minúscules!

La sintaxi per a la classe

Per crear una classe escriu un punt (.), seguit d'un nom de classe. A continuació, defineix les propietats CSS dins de claus {}. Ja hem vist .city abans:

```
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
```

Classes múltiples

Els elements HTML poden pertànyer a més d'una classe.

Per definir diverses classes, separeu els noms de classe amb un espai, p. ex. <div class="city main">. L'element s'estilitzarà segons totes les classes especificades.

A l'exemple següent, el primer <h2> element pertany tant a la **city** classe com a la **main** classe, i obtindrà els estils CSS de les dues classes. Exemple:

```
<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>
```

Diferents elements poden compartir la mateixa classe

Diferents elements HTML poden apuntar al mateix nom de classe.

A l'exemple següent, tots dos <h2> i <p> assenyalen la classe "ciutat" i compartiran el mateix estil. Exemple:

```
<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>
```

Ús de l'atribut de classe en JavaScript

JavaScript també pot utilitzar el nom de la classe per realitzar determinades tasques per a elements específics.

JavaScript pot accedir a elements amb un nom de classe específic amb el **getElementsByClassName()** mètode:

Exemple. Feu clic a un botó per amagar tots els elements amb el nom de classe "city":

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
```

```
}  
</script>
```

21. Atribut id HTML

L'atribut HTML **id** s'utilitza per especificar un identificador únic per a un element HTML.

No podeu tenir més d'un element amb el mateix identificador en un document HTML.

Ús de l'atribut id

L'atribut **id** especifica un identificador únic per a un element HTML. El valor de l'atribut **id** ha de ser únic dins del document HTML.

L'atribut **id** s'utilitza per apuntar a una declaració d'estil específica en un full d'estil. JavaScript també l'utilitza per accedir i manipular l'element amb l'identificador específic.

La sintaxi per a id és: escriuiu un caràcter hash (#), seguit d'un nom d'identificador. A continuació, definiu les propietats CSS dins de claus {}.

A l'exemple següent tenim un `<h1>` element que apunta al nom d'identificador "myHeader". Aquest `<h1>` element s'estilitzarà segons la `#myHeader` definició d'estil a la secció de capçalera. Exemple:

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#myHeader {  
    background-color: lightblue;  
    color: black;  
    padding: 40px;  
    text-align: center;  
}  
</style>  
</head>  
<body>  
<h1 id="myHeader">My Header</h1>  
</body>  
</html>
```

Nota: el nom d'identificador distingeix entre majúscules i minúscules! El nom d'identificador ha de contenir almenys un caràcter, no pot començar amb un número i no ha de contenir espais en blanc (espais, tabulacions, etc.).

Diferència entre classe i ID

Un nom de classe pot ser utilitzat per diversos elements HTML, mentre que un nom d'identificació només l'ha d'utilitzar un element HTML dins de la pàgina. Exemple:

```
<style>  
/* Style the element with the id "myHeader" */  
#myHeader {  
    background-color: lightblue;  
    color: black;  
    padding: 40px;  
    text-align: center;  
}
```

```

/* Style all elements with the class name "city" */
.city {
    background-color: tomato;
    color: white;
    padding: 10px;
}
</style>

<!-- An element with a unique id -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple elements with same class -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

```

Ús de l'atribut id a JavaScript

JavaScript també pot utilitzar l' atribut **id** per realitzar algunes tasques per a aquest element específic.

JavaScript pot accedir a un element amb un identificador específic amb el **getElementById()** mètode:

Exemple. Utilitzeu l'atribut **id** per manipular text amb JavaScript:

```

<script>
function displayResult() {
    document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>

```

22. Iframes HTML

Un iframe HTML s'utilitza per mostrar una pàgina web dins d'una pàgina web.

Sintaxi HTML Iframe

L'etiqueta HTML **<iframe>** especifica un marc en línia.

Un marc en línia s'utilitza per incrustar un altre document dins del document HTML actual.

Sintaxi

```
<iframe src="url " title="description"></iframe>
```

Consell: és una bona pràctica incloure sempre un atribut **title** per a **<iframe>**. Els lectors de pantalla l'utilitzen per llegir quin és el contingut de l'iframe.

Iframe: estableix l'alçada i l'amplada

Utilitzeu els atributs **height** i **width** per especificar la mida de l'iframe.

L'alçada i l'amplada s'especifiquen en píxels de manera predeterminada. Exemple:

```
<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>
```

Iframe: eliminar la vora

Per defecte, un iframe té una vora al seu voltant.

Per eliminar la vora, afegiu l'atribut **style** i utilitzeu la propietat **border** CSS. Exemple:

```
<iframe src="demo_iframe.htm" style="border:none;" title="Iframe Example"></iframe>
```

Amb CSS, també podeu canviar la mida, l'estil i el color de la vora de l'iframe:

```
<iframe src="demo_iframe.htm" style="border:2px solid red;" title="Iframe Example"></iframe>
```

Iframe: objectiu per a un enllaç

Un iframe es pot utilitzar com a marc de destinació per a un enllaç.

L'atribut **target** de l'enllaç ha de fer referència a l'atribut **name** de l'iframe. Exemple:

```
<!DOCTYPE html>
<html>
<body>

  <h2>Iframe - Target for a Link</h2>

  <iframe src="demo_iframe.htm" name="iframe_a" height="300px" width="50%"
title="Iframe Example"></iframe>

  <p><a href="https://www.w3schools.com"
target="iframe_a">W3Schools.com</a></p>

  <p>When the target attribute of a link matches the name of an iframe, the
link will open in the iframe.</p>

</body>
</html>
```

23. JavaScript HTML

JavaScript fa que les pàgines HTML siguin més dinàmiques i interactives.

L'etiqueta HTML <script>

L'etiqueta HTML **<script>** s'utilitza per definir un script del costat del client (JavaScript).

L'element **<script>** conté declaracions d'script o apunta a un fitxer de script extern mitjançant l'atribut **src**.

Els usos habituals de JavaScript són la manipulació d'imatges, la validació de formularis i els canvis dinàmics de contingut.

Per seleccionar un element HTML, JavaScript fa servir el mètode **document.getElementById()**

Aquest exemple de JavaScript escriu "Hola JavaScript!" en un element HTML amb id="demo":

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

Una xicoteta mostra de JavaScript

Aquests són alguns exemples del que pot fer JavaScript:

Exemple. JavaScript pot canviar el contingut:

```
document.getElementById("demo").innerHTML = "Hello JavaScript!";
```

JavaScript pot canviar els estils:

```
document.getElementById("demo").style.fontSize = "25px";  
document.getElementById("demo").style.color = "red";  
document.getElementById("demo").style.backgroundColor = "yellow";
```

JavaScript pot canviar els atributs:

```
document.getElementById("image").src = "picture.gif";
```

L'etiqueta HTML <noscript>

L'etiqueta HTML <noscript> defineix un contingut alternatiu que es mostrarà als usuaris que tinguen scripts desactivats al seu navegador o que tinguen un navegador que no admet scripts. Exemple:

```
<script>  
document.getElementById("demo").innerHTML = "Hello JavaScript!";  
</script>  
<noscript>Sorry, your browser does not support JavaScript!</noscript>
```

24. Rutes dels fitxers HTML

Una ruta de fitxer descriu la ubicació d'un fitxer a l'estructura de carpetes d'un lloc web.

Exemples de ruta de fitxer

Camí	Descripció
	El fitxer "picture.jpg" es troba a la mateixa carpeta que la pàgina actual
	El fitxer "picture.jpg" es troba a la carpeta d'imatges de la carpeta actual
	El fitxer "picture.jpg" es troba a la carpeta d'imatges a l'arrel del web actual
	El fitxer "picture.jpg" es troba a la carpeta un nivell més amunt de la carpeta actual

Rutes dels fitxers HTML

Una ruta de fitxer descriu la ubicació d'un fitxer a l'estructura de carpetes d'un lloc web.

Els camins dels fitxers s'utilitzen quan s'enllaça amb fitxers externs, com ara:

- Pàgines web
- Imatges
- Fulls d'estil
- JavaScripts

Rutes de fitxer absolutes

Una ruta de fitxer absoluta és l'URL complet d'un fitxer:

```

```

Camins de fitxers relatius

Una ruta relativa del fitxer apunta a un fitxer relatiu a la pàgina actual.

A l'exemple següent, la ruta del fitxer apunta a un fitxer de la carpeta d'imatges situada a l'arrel del web actual:

```

```

Millors pràctiques

La millor pràctica és utilitzar camins de fitxer relatius (si és possible).

Quan utilitzeu camins de fitxer relatius, les vostres pàgines web no estaran vinculades a la vostra URL base actual. Tots els enllaços funcionaran al vostre propi ordinador (localhost), així com al vostre domini públic actual i als vostres futurs dominis públics.

25. Capçalera HTML

L'element HTML `<head>` és un contenidor per als elements següents: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>` i `<base>`.

L'element HTML `<head>`

L'element `<head>` és un contenidor per a metadades (dades sobre dades) i es col·loca entre l'etiqueta `<html>` i l'etiqueta `<body>`.

Les metadades HTML són dades sobre el document HTML. Les metadades no es mostren.

Les metadades normalment defineixen el títol del document, el conjunt de caràcters, els estils, els scripts i altres metainformació.

L'element HTML `<title>`

L'element `<title>` defineix el títol del document. El títol ha de ser només de text i es mostra a la barra de títol del navegador o a la pestanya de la pàgina.

El contingut d'un títol de pàgina és molt important per a l'optimització de motors de cerca. Els algorismes dels motors de cerca utilitzen el títol de la pàgina per decidir l'ordre quan s'enumeren pàgines als resultats de la cerca.

L'element `<title>`:

- defineix un títol a la barra d'eines del navegador
- proporciona un títol per a la pàgina quan s'afegeix als preferits
- mostra un títol per a la pàgina als resultats del motor de cerca

Per tant, intenta que el títol siga com més precís i significatiu millor!

L'element HTML `<style>`

L'element `<style>` s'utilitza per definir informació d'estil per a una única pàgina HTML:

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

L'element HTML <link>

L'element' <link> defineix la relació entre el document actual i un recurs extern.

L'etiqueta <link> s'utilitza sovint per enllaçar a fulls d'estil externs:

```
<link rel="stylesheet" href="mystyle.css">
```

L'element HTML <meta>

L'element <meta> s'utilitza normalment per especificar el conjunt de caràcters, la descripció de la pàgina, les paraules clau, l'autor del document i la configuració de la finestra gràfica.

Les metadades no es mostraran a la pàgina, però les fan servir els navegadors (com mostrar el contingut o tornar a carregar la pàgina), els motors de cerca (paraules clau) i altres serveis web. Exemple:

```
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
```

Configuració de la finestra gràfica

La finestra gràfica és l'àrea visible de l'usuari d'una pàgina web. Varia segons el dispositiu: serà més xicoteta en un telèfon mòbil que en una pantalla d'ordinador.

Heu d'incloure l'element <meta> següent a totes les vostres pàgines web:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Això dona al navegador instruccions sobre com controlar les dimensions i l'escala de la pàgina.

La **width=device-width** estableix l'amplada de la pàgina a l'amplada de la pantalla del dispositiu (que variarà segons el dispositiu).

La **initial-scale=1.0** estableix el nivell de zoom inicial quan el navegador carrega la pàgina per primera vegada.

L'element HTML <script>

Ja vistos.

L'element HTML <base>

L'element' <base> especifica l'URL base i/o l'objectiu per a tots els URL relatius d'una pàgina.

L'etiqueta <base> ha de tenir present un atribut href o target, o tots dos.

Només hi pot haver un únic <base> element en un document!

Exemple. Especifica un URL predeterminat i un objectiu predeterminat per a tots els enllaços d'una pàgina:

```
<head>
  <base href="https://www.w3schools.com/" target="_blank">
</head>
```


Elements de capçalera HTML

Tag	Description
<code><head></code>	Defines information about the document
<code><title></code>	Defines the title of a document
<code><base></code>	Defines a default address or a default target for all links on a page
<code><link></code>	Defines the relationship between a document and an external resource
<code><meta></code>	Defines metadata about an HTML document
<code><script></code>	Defines a client-side script
<code><style></code>	Defines style information for a document

26. Elements i tècniques de disseny HTML

Els llocs web normalment mostren contingut en diverses columnes (com una revista o un diari).

Elements de disseny HTML

HTML té diversos elements semàntics que defineixen les diferents parts d'una pàgina web:



- **`<header>`**- Defineix una capçalera per a un document o una secció
- **`<nav>`**- Defineix un conjunt d'enllaços de navegació
- **`<section>`**- Defineix una secció en un document
- **`<article>`**- Defineix un contingut independent i autònom
- **`<aside>`**- Defineix contingut a part del contingut (com una barra lateral)
- **`<footer>`**- Defineix un peu de pàgina per a un document o una secció
- **`<details>`**- Defineix detalls addicionals que l'usuari pot obrir i tancar sota demanda
- **`<summary>`**- Defineix un encapçalament per a l'element **`<details>`**

Tècniques de disseny HTML

Hi ha quatre tècniques diferents per crear dissenys multicolumnes. Cada tècnica té els seus pros i contres:

- marc CSS
- Propietat flotant CSS
- Flexbox CSS
- quadrícula CSS

Frameworks CSS

Si voleu crear el vostre disseny ràpidament, podeu utilitzar un marc CSS, com [W3.CSS](#) o [Bootstrap](#).

Disseny flotant CSS

És habitual fer dissenys web sencers mitjançant la propietat **float** CSS. Float és fàcil d'aprendre: només cal recordar com funcionen les propietats **float** i **clear**.

Desavantatges: els elements flotants estan lligats al flux de documents, cosa que pot perjudicar la flexibilitat.

Disseny CSS Flexbox

L'ús de flexbox garanteix que els elements es comporten de manera previsible quan el disseny de la pàgina ha d'adaptar-se a diferents mides de pantalla i diferents dispositius de visualització.

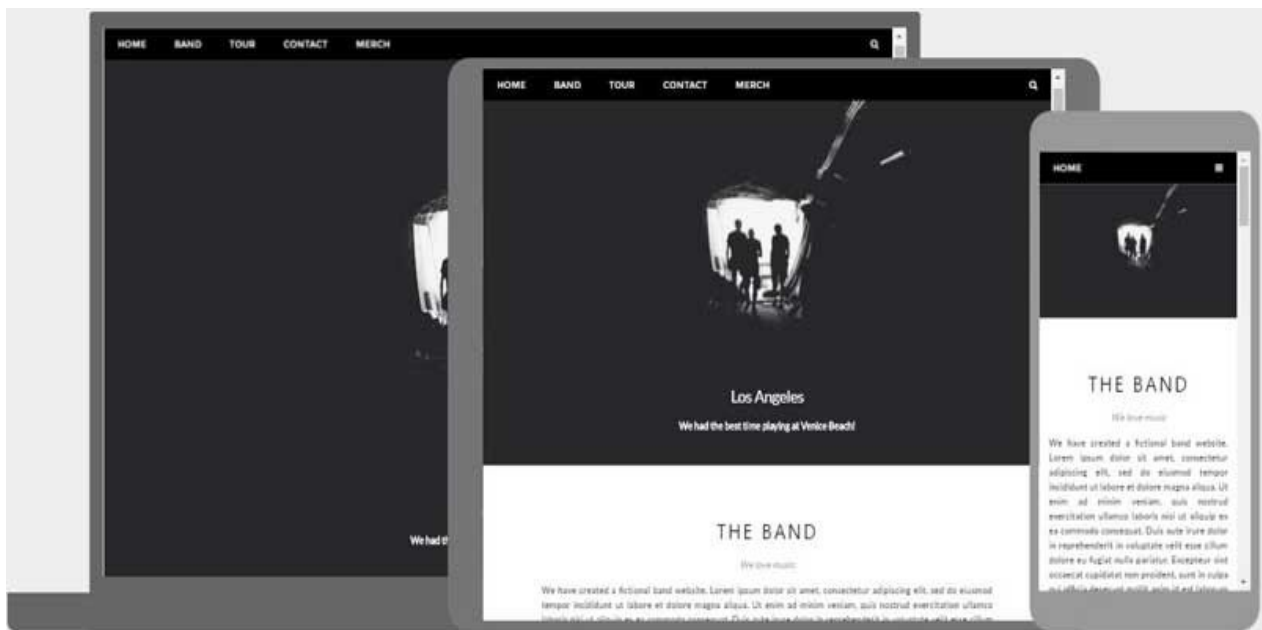
Disseny de quadrícula CSS

El mòdul de disseny de quadrícula CSS ofereix un sistema de disseny basat en quadrícula, amb files i columnes, que facilita el disseny de pàgines web sense haver d'utilitzar flotadors i posicionament.

27. Disseny web responsiu HTML

El disseny web responsiu consisteix a crear pàgines web que es vegen bé en tots els dispositius!

Un disseny web responsiu s'ajustarà automàticament a diferents mides de pantalla i finestres.



Què és el disseny web responsiu?

El disseny web responsiu consisteix a utilitzar HTML i CSS per redimensionar, amagar, reduir o ampliar automàticament un lloc web per fer-lo veure bé en tots els dispositius (ordinadors, tauletes i telèfons):

Configuració de la finestra gràfica

Per crear un lloc web responsiu, afegiu l'etiqueta `<meta>` següent a totes les vostres pàgines web. Exemple:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Això establirà la finestra gràfica de la vostra pàgina, que donarà instruccions al navegador sobre com controlar les dimensions i l'escala de la pàgina.

Imatges sensibles o responsives

Les imatges responsives són imatges que s'ajusten bé a qualsevol mida del navegador.

Utilitzant la propietat `width`. Si la propietat `width` CSS s'estableix al 100%, la imatge respondrà i augmentarà i baixarà l'escala. Exemple:

```

```

Tingueu en compte que a l'exemple anterior, la imatge es pot escalar per a ser més gran que la seva mida original. Una millor solució, en molts casos, serà utilitzar la propietat `max-width`.

Utilitzant la propietat `max-width`

Si la propietat `max-width` s'estableix al 100%, la imatge es reduirà si cal, però mai augmentarà per ser més gran que la seva mida original. Exemple:

```

```

Mostra imatges diferents segons l'amplada del navegador

L'element `<picture>`. Vist en el punt sobre imatges.

Mida del text sensible

La mida del text es pot configurar amb una unitat "vw", que significa "amplada de la finestra gràfica".

D'aquesta manera, la mida del text seguirà la mida de la finestra del navegador. Exemple

```
<h1 style="font-size:10vw">Hello World</h1>
```

Viewport és la mida de la finestra del navegador. 1vw = 1% de l'amplada de la finestra visual. Si el visor té 50 cm d'ample, 1vw és 0,5 cm.

Consultes de mitjans

A més de canviar la mida del text i les imatges, també és habitual utilitzar consultes multimèdia en pàgines web responsives.

Amb les consultes multimèdia podeu definir estils completament diferents per a diferents mides de navegador.

Exemple: canvieu la mida de la finestra del navegador per veure que els tres elements div següents es mostraran horitzontalment en pantalles grans i s'apilen verticalment en pantalles xicotetes:

```
<style>
.left, .right {
  float: left;
  width: 20%; /* The width is 20%, by default */
}

.main {
  float: left;
  width: 60%; /* The width is 60%, by default */
}
/* Use a media query to add a breakpoint at 800px: */
```

```

@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or
smaller */
  }
}
</style>

```

Pàgina web responsiva: exemple complet

Una pàgina web responsiva hauria de tenir un bon aspecte en pantalles grans d'escriptori i en telèfons mòbils xicotets. Prova aquest exemple:

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}

.menu {
  float: left;
  width: 20%;
  text-align: center;
}

.menu a {
  background-color: #e5e5e5;
  padding: 8px;
  margin-top: 7px;
  display: block;
  width: 100%;
  color: black;
}

.main {
  float: left;
  width: 60%;
  padding: 0 20px;
}

.right {
  background-color: #e5e5e5;
  float: left;
  width: 20%;
  padding: 15px;
  margin-top: 7px;
  text-align: center;
}

@media only screen and (max-width: 620px) {
  /* For mobile phones: */
  .menu, .main, .right {
    width: 100%;
  }
}
</style>
</head>
<body style="font-family:Verdana;color:#aaaaaa;">

<div style="background-color:#e5e5e5;padding:15px;text-align:center;">
  <h1>Hello World</h1>

```

```

</div>

<div style="overflow:auto">
  <div class="menu">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
    <a href="#">Link 4</a>
  </div>

  <div class="main">
    <h2>Lorum Ipsum</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
    nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
    volutpat.</p>
  </div>

  <div class="right">
    <h2>About</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</div>

<div style="background-color:#e5e5e5;text-align:center;padding:10px;margin-
top:7px;">© copyright w3schools.com</div>

</body>
</html>

```

Bootstrap

Un autre marque CSS populaire est Bootstrap:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap 5 Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.c
ss" rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.
min.js"></script>
</head>
<body>

<div class="container-fluid p-5 bg-primary text-white text-center">
  <h1>My First Bootstrap Page</h1>
  <p>Resize this responsive page to see the effect!</p>
</div>

<div class="container mt-5">
  <div class="row">
    <div class="col-sm-4">
      <h3>Column 1</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris...</p>
    </div>
    <div class="col-sm-4">
      <h3>Column 2</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>

```

```

        <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris...</p>
    </div>
    <div class="col-sm-4">
        <h3>Column 3</h3>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</p>
        <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris...</p>
    </div>
</div>
</div>
</div>

</body>
</html>

```

28. Elements del codi informàtic HTML

HTML `<code>` per al codi informàtic

L'element `<code>` HTML s'utilitza per definir un fragment de codi informàtic. El contingut interior es mostra amb el tipus de lletra monoespai predeterminat del navegador.

Penseu que l'element `<code>` no conserva espais en blanc addicionals i salts de línia.

Per solucionar-ho, podeu posar l'element `<code>` dins d'un element `<pre>`. Exemple:

```

<pre>
<code>
x = 5;
y = 6;
z = x + y;
</code>
</pre>

```

29. Elements semàntics HTML

Elements semàntics = elements amb significat.

Què són els elements semàntics?

Un element semàntic descriu clarament el seu significat tant per al navegador com per al desenvolupador.

Exemples d'elements **no semàntics** : `<div>` i `` - No diu res sobre el seu contingut.

Exemples d'elements **semàntics** : `<form>`, `<table>`, i `<article>` - Defineix clarament el seu contingut.

Elements semàntics en HTML

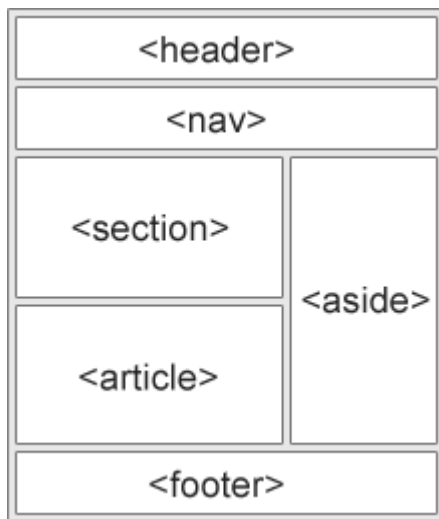
Molts llocs web contenen codi HTML com: `<div id="nav">` `<div class="header">` `<div id="footer">` per indicar la navegació, la capçalera i el peu de pàgina.

En HTML hi ha alguns elements semàntics que es poden utilitzar per definir diferents parts d'una pàgina web:

```

<article> <aside> <details> <figcaption> <figure> <footer> <header> <main>
<mark> <nav> <section> <summary> <time>

```



Element <secció> HTML

L'element **<section>** defineix una secció en un document.

Segons la documentació HTML del W3C: "Una secció és una agrupació temàtica de contingut, normalment amb un encapçalament".

Exemples d'on es pot utilitzar un element **<section>**:

- Capítols
- Introducció
- Notícies
- Informació de contacte

Una pàgina web normalment es pot dividir en seccions per a la introducció, el contingut i la informació de contacte.

Exemple. Dues seccions en un document:

```
<section>
<h1>WWF</h1>
<p>The World Wide Fund for Nature (WWF) is an international organization
working on issues regarding the conservation, research and restoration of
the environment, formerly named the World Wildlife Fund. WWF was founded in
1961.</p>
</section>

<section>
<h1>WWF's Panda symbol</h1>
<p>The Panda has become the symbol of WWF. The well-known panda logo of WWF
originated from a panda named Chi Chi that was transferred from the Beijing
Zoo to the London Zoo in the same year of the establishment of WWF.</p>
</section>
```

Element <article> HTML

L'element **<article>** especifica contingut independent i autònom.

Un article hauria de tenir sentit per si mateix i hauria de ser possible distribuir-lo independentment de la resta del lloc web.

Exemples d'on es pot utilitzar l'element **<article>**:

- Publicacions del fòrum
- Publicacions del blog

- Comentaris dels usuaris
- Targetes de productes
- Articles periodístics

Exemple .Dos articles amb contingut independent i autònom:

```
<article>
  <h2>Google Chrome</h2>
  <p>Google Chrome is a web browser developed by Google, released in 2008.
  Chrome is the world's most popular web browser today!</p>
</article>

<article>
  <h2>Mozilla Firefox</h2>
  <p>Mozilla Firefox is an open-source web browser developed by Mozilla.
  Firefox has been the second most popular web browser since January,
  2018.</p>
</article>
```

Exemple 2. Utilitzeu CSS per estilitzar l'element <article>:

```
<html>
<head>
<style>
.all-browsers {
  margin: 0;
  padding: 5px;
  background-color: lightgray;
}

.all-browsers > h1, .browser {
  margin: 10px;
  padding: 5px;
}

.browser {
  background: white;
}

.browser > h2, p {
  margin: 4px;
  font-size: 90%;
}
</style>
</head>
<body>
<article class="all-browsers">
  <h1>Most Popular Browsers</h1>
  <article class="browser">
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in
    2008. Chrome is the world's most popular web browser today!</p>
  </article>
  <article class="browser">
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla.
    Firefox has been the second most popular web browser since January,
    2018.</p>
  </article>
  <article class="browser">
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in
    2015. Microsoft Edge replaced Internet Explorer.</p>
```



```
</article>
</article>
</body>
</html>
```

Nidificar <article> a <section> o viceversa?

L'element <article> especifica contingut independent i autònom.

L'element <section> defineix la secció d'un document.

Podem utilitzar les definicions per decidir com aniar aquests elements? No podem!

Per tant, trobareu pàgines HTML amb <section> elements que contenen <article> elements i <article> elements que contenen elements <section>.

Element HTML <capçalera>

L'element <header> representa un contenidor per a contingut introductori o un conjunt d'enllaços de navegació.

Un element <header> normalment conté:

- un o més elements d'encapçalament (<h1> - <h6>)
- logotip o icona
- informació d'autoria

Nota: podeu tenir diversos <header> elements en un document HTML. Tanmateix, <header> no es pot col·locar dins d'un <footer>, <address> o d'un altre <header> element.

Exemple. Una capçalera per a un <article>:

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment,
and build a future in which humans live in harmony with nature.</p>
</article>
```

Element HTML <footer>

L'element <footer> defineix un peu de pàgina per a un document o secció.

Un element <footer> normalment conté:

- informació d'autoria
- informació de copyright
- informació de contacte
- mapa del lloc
- tornar als enllaços superiors
- documents relacionats

Podeu tenir diversos elements <footer> en un document.

Exemple. Una secció de peu de pàgina d'un document:

```
<footer>
  <p>Author: Hege Refsnes</p>
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>
```

Element <nav> HTML

L'element <nav> defineix un conjunt d'enllaços de navegació.

Tingueu en compte que NO tots els enllaços d'un document haurien d'estar dins d'un element <nav>. L'element <nav> només està pensat per als principals blocs d'enllaços de navegació.

Els navegadors, com ara els lectors de pantalla per a usuaris amb discapacitat, poden utilitzar aquest element per determinar si s'ometen la representació inicial d'aquest contingut.

Exemple. Un conjunt d'enllaços de navegació:

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

Element HTML <aside>

L'element <aside> defineix algun contingut a part del contingut en què es col·loca (com una barra lateral).

El contingut <aside> ha d'estar indirectament relacionat amb el contingut que l'envolta.

Exemple. Mostra alguns continguts a part del contingut on es col·loca:

```
<p>My family and I visited The Epcot center this summer. The weather was
nice, and Epcot was amazing! I had a great summer together with my family!
</p>

<aside>
<h4>Epcot Center</h4>
<p>Epcot is a theme park at Walt Disney World Resort featuring exciting
attractions, international pavilions, award-winning fireworks and seasonal
special events.</p>
</aside>
```

Exemple 2. Utilitzeu CSS per estilitzar l'element <aside>:

```
<html>
<head>
<style>
  aside {
    width: 30%;
    padding-left: 15px;
    margin-left: 15px;
    float: right;
    font-style: italic;
    background-color: lightgray;
  }
</style>
</head>
```

```

<body>

<p>My family and I visited The Epcot center this summer. The weather was
nice, and Epcot was amazing! I had a great summer together with my family!
</p>

<aside>
<p>The Epcot center is a theme park at Walt Disney World Resort featuring
exciting attractions, international pavilions, award-winning fireworks and
seasonal special events.</p>
</aside>

<p>My family and I visited The Epcot center this summer. The weather was
nice, and Epcot was amazing! I had a great summer together with my family!
</p>
<p>My family and I visited The Epcot center this summer. The weather was
nice, and Epcot was amazing! I had a great summer together with my family!
</p>

</body>
</html>

```

Elements HTML <figure> i <figcaption>

L'etiqueta **<figure>** especifica contingut autònom, com ara il·lustracions, diagrames, fotos, llistats de codi, etc.

L'etiqueta **<figcaption>** defineix un títol per a un element **<figure>**. L'element **<figcaption>** es pot col·locar com a primer o com a darrer fill d'un element **<figure>**.

L'element **** defineix la imatge/il·lustració real. Exemple:

```

<figure>
  
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>
</figure>

```

Per què els elements semàntics?

Segons el W3C: "Una web semàntica permet compartir i reutilitzar dades entre aplicacions, empreses i comunitats".

Elements semàntics en HTML

A continuació es mostra una llista d'alguns dels elements semàntics en HTML.

Tag	Description
<u><article></u>	Defines independent, self-contained content
<u><aside></u>	Defines content aside from the page content
<u><details></u>	Defines additional details that the user can view or hide
<u><figcaption></u>	Defines a caption for a <figure> element
<u><figure></u>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<u><footer></u>	Defines a footer for a document or section
<u><header></u>	Specifies a header for a document or section
<u><main></u>	Specifies the main content of a document
<u><mark></u>	Defines marked/highlighted text
<u><nav></u>	Defines navigation links
<u><section></u>	Defines a section in a document
<u><summary></u>	Defines a visible heading for a <details> element
<u><time></u>	Defines a date/time

30. Guia d'estil HTML

Un codi HTML coherent, net i ordenat fa que siga més fàcil per als altres llegir i entendre el vostre codi.

Ací teniu algunes directrius i consells per crear un bon codi HTML.

Declarar sempre el tipus de document

Declara sempre el tipus de document com a primera línia del document.

El tipus de document correcte per a HTML és:

```
<!DOCTYPE html>
```

Utilitzeu noms d'elements en minúscules

HTML permet barrejar lletres majúscules i minúscules en els noms dels elements.

Però, us recomanem que utilitzeu noms d'elements en minúscules, perquè:

- Barrejar noms en majúscules i minúscules sembla malament
- Els desenvolupadors normalment utilitzen noms en minúscula
- Les minúscules es veuen més netes
- Les minúscules són més fàcils d'escriure

Tanca tots els elements HTML

En HTML, no cal que tanqueu tots els elements (per exemple, l'element `<p>`).

Però, us recomanem fermament que tanqueu tots els elements HTML.

Utilitzeu noms d'atributs en minúscules

HTML permet barrejar lletres majúscules i minúscules en els noms dels atributs.

Bé:

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

Mal:

```
<a HREF="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

Citeu sempre els valors dels atributs

HTML permet valors d'atributs sense cometes. Però, recomanem citar els valors dels atributs, perquè:

- Els desenvolupadors solen citar valors d'atributs
- Els valors citats són més fàcils de llegir
- Heu d'utilitzar cometes si el valor conté espais

Especifiqueu sempre alt, amplada i alçada per a les imatges

Especifiqueu sempre l'atribut `alt` de les imatges. Aquest atribut és important si la imatge per algun motiu no es pot mostrar.

A més, defineix sempre el `width` i `height` de les imatges. Això redueix el parpelleig, perquè el navegador pot reservar espai per a la imatge abans de carregar-la.

Espais i signes iguals

HTML permet espais al voltant dels signes iguals. Però sense espai és més fàcil de llegir i agrupa millor les entitats.

Eviteu les línies de codi llargues

Quan utilitzeu un editor HTML, NO és convenient desplaçar-vos cap a la dreta i l'esquerra per llegir el codi HTML. Intenteu evitar línies de codi massa llargues.

Línies en blanc i sagnat

No afegiu línies, espais o sagnat en blanc sense un motiu.

Per facilitar la lectura, afegiu línies en blanc per separar blocs de codi grans o lògics.

Exemple de bona taula:

```
<table>
  <tr>
    <th>Name</th>
    <th>Description</th>
  </tr>
  <tr>
    <td>A</td>
    <td>Description of A</td>
  </tr>
  <tr>
    <td>B</td>
    <td>Description of B</td>
  </tr>
</table>
```

Exemple de bona llista:

```
<ul>
  <li>London</li>
  <li>Paris</li>
  <li>Tokyo</li>
</ul>
```

No ometeu mai l'element <title>

L'element <title> és obligatori en HTML.

El contingut d'un títol de pàgina és molt important per a l'optimització de motors de cerca (SEO)! Els algorismes dels motors de cerca utilitzen el títol de la pàgina per decidir l'ordre quan s'enumeren pàgines als resultats de la cerca.

L'element <title>:

- defineix un títol a la barra d'eines del navegador
- proporciona un títol per a la pàgina quan s'afegeix als preferits
- mostra un títol per a la pàgina als resultats del motor de cerca

Per tant, intenta que el títol siga com més precís i significatiu millor:

```
<title>HTML Style Guide and Coding Conventions</title>
```

Vols ometre <html> i <body>?

Una pàgina HTML es validarà sense les etiquetes <html> i <body>:

Però, us recomanem que afegiu sempre les etiquetes <html> i <body>!

L'omissió <body> pot produir errors en navegadors antics.

Omet <html> i <body> també pot bloquejar el programari DOM i XML.

Vols ometre <head>?

L'etiqueta HTML <head> també es pot ometre.

Els navegadors afegiran tots els elements abans de <body>, a un <head> element predeterminat.

Però, recomanem utilitzar l'etiqueta <head>.

Tanqueu els elements HTML buits?

En HTML, és opcional tancar els elements buits.

Permès:

```
<meta charset="utf-8">
```

També es permet:

```
<meta charset="utf-8" />
```

Si espereu que el programari XML/XHTML accedeisca a la vostra pàgina, manteniu la barra inclinada de tancament (/), perquè és obligatori en XML i XHTML.

Afegiu l'atribut lang

Sempre hauríeu d'incloure l'atribut **lang** dins de l'etiqueta <html>, per declarar l'idioma de la pàgina web. Això està pensat per ajudar els motors de cerca i els navegadors.

```
<!DOCTYPE html>
<html lang="en-us">
<head>
...
```

Metadades

Per garantir una interpretació adequada i una indexació correcta del motor de cerca, tant l'idioma com la codificació de caràcters s'han de definir en un document HTML:

```
<head>
  <meta charset="UTF-8">
  <title>Page Title</title>
</head>
```

Configuració de la finestra gràfica

La finestra gràfica és l'àrea visible de l'usuari d'una pàgina web. Varia segons el dispositiu: serà més xicoteta en un telèfon mòbil que en una pantalla d'ordinador.

Heu d'incloure l'element <meta> següent a totes les vostres pàgines web:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Comentaris HTML

Els comentaris breus s'han d'escriure en una línia, com aquesta:

```
<!-- This is a comment -->
```

Els comentaris que abasten més d'una línia s'han d'escriure així:

```
<!--
  This is a long comment example. This is a long comment example.
-->
```

```
This is a long comment example. This is a long comment example.
-->
```

Els comentaris llargs són més fàcils d'observar si estan sagnats amb dos espais.

Ús de fulls d'estil

Utilitzeu una sintaxi senzilla per enllaçar amb fulls d'estil :

```
<link rel="stylesheet" href="styles.css">
```

Les regles CSS curtes es poden escriure comprimides, com aquesta:

```
p.intro {font-family:Verdana;font-size:16em;}
```

Les regles CSS llargues s'han d'escriure en diverses línies:

```
body {
  background-color: lightgrey;
  font-family: "Arial Black", Helvetica, sans-serif;
  font-size: 16em;
  color: black;
}
```

- Col·loqueu el suport d'obertura a la mateixa línia que el selector
- Feu servir un espai abans del suport d'obertura
- Utilitzeu dos espais de sagnat
- Utilitzeu punt i coma després de cada parell propietat-valor, inclòs l'últim
- Només utilitzeu cometes al voltant dels valors si el valor conté espais
- Col·loqueu el suport de tancament en una línia nova, sense espais davanters

S'està carregant JavaScript en HTML

Utilitzeu una sintaxi senzilla per carregar scripts externs:

```
<script src="myscript.js">
```

Accés a elements HTML amb JavaScript

L'ús de codi HTML "desordenat" pot provocar errors de JavaScript.

Aquestes dues declaracions de JavaScript produiran resultats diferents:

```
getElementById("Demo").innerHTML = "Hello";
getElementById("demo").innerHTML = "Hello";
```

Utilitzeu noms de fitxers en minúscules

Alguns servidors web (Apache, Unix) distingeixen entre majúscules i minúscules sobre els noms de fitxers: no es pot accedir a "london.jpg" com a "London.jpg".

Altres servidors web (Microsoft, IIS) no distingeixen entre majúscules i minúscules: es pot accedir a "london.jpg" com a "London.jpg".

Si feu servir una barreja de majúscules i minúscules, n'heu de ser conscient.

Si passeu d'un servidor que no distingeix entre majúscules i minúscules a un servidor, fins i tot xicotets errors trencaran la vostra web!

Per evitar aquests problemes, utilitzeu sempre noms de fitxers en minúscules!

Extensions de fitxer

Els fitxers HTML han de tenir una extensió **.html** (es permet **.htm**).

Els fitxers CSS haurien de tenir una extensió **.css** .

Els fitxers JavaScript haurien de tenir una extensió **.js** .

Diferències entre .htm i .html?

No hi ha cap diferència entre les extensions de fitxer .htm i .html!

Tots dos seran tractats com a HTML per qualsevol navegador web i servidor web.

Noms de fitxer per defecte

Quan un URL no especifica un nom de fitxer al final (com ara "https://www.w3schools.com/"), el servidor només afegeix un nom de fitxer predeterminat, com ara "index.html", "index.htm", " default.html", o "default.htm".

Si el vostre servidor només està configurat amb "index.html" com a nom de fitxer predeterminat, el vostre fitxer s'ha de denominar "index.html" i no "default.html".

Tanmateix, els servidors es poden configurar amb més d'un nom de fitxer predeterminat. Normalment podeu configurar tants noms de fitxer predeterminats com vulgueu.

31. Entitats HTML

Els caràcters reservats en HTML s'han de substituir per entitats de caràcters.

Entitats HTML

Alguns caràcters estan reservats en HTML.

Si feu servir els signes de menys de (<) o de major que (>) al vostre text, el navegador pot barrejar-los amb etiquetes.

Les entitats de caràcters s'utilitzen per mostrar caràcters reservats en HTML.

Una entitat de caràcter té aquest aspecte:

&entity_name;

OR

#entity_number;

Per mostrar un signe menor que (<) hem d'escriure: < or <

Avantatge d'utilitzar un nom d'entitat: un nom d'entitat és fàcil de recordar.

Desavantatge d'utilitzar un nom d'entitat: és possible que els navegadors no admeten tots els noms d'entitat, però el suport per als números d'entitat és bo.

Espai ininterromput

Una entitat que s'utilitza habitualment en HTML és l'espai ininterromput:

Un espai que no es trenca és un espai que no trencarà en una línia nova.

Dues paraules separades per un espai ininterromput s'enganxaran (no es trenquen en una línia nova). Això és útil quan trencar les paraules pot ser pertorbador. Exemple:

•10 km/h

Un altre ús comú de l'espai no trencat és evitar que els navegadors trunquen espais a les pàgines HTML.

Si escriviu 10 espais al vostre text, el navegador n'eliminarà 9. Per afegir espais reals al vostre text, podeu utilitzar el ` ` entitat de caràcter.

Algunes entitats de caràcters HTML útils

Result	Description	Entity Name	Entity Number
	non-breaking space	<code>&nbsp;</code>	<code>&#160;</code>
<code><</code>	less than	<code>&lt;</code>	<code>&#60;</code>
<code>></code>	greater than	<code>&gt;</code>	<code>&#62;</code>
<code>&</code>	ampersand	<code>&amp;</code>	<code>&#38;</code>
<code>"</code>	double quotation mark	<code>&quot;</code>	<code>&#34;</code>
<code>'</code>	single quotation mark (apostrophe)	<code>&apos;</code>	<code>&#39;</code>
¢	cent	<code>&cent;</code>	<code>&#162;</code>
£	pound	<code>&pound;</code>	<code>&#163;</code>
¥	yen	<code>&yen;</code>	<code>&#165;</code>
€	euro	<code>&euro;</code>	<code>&#8364;</code>
©	copyright	<code>&copy;</code>	<code>&#169;</code>
®	registered trademark	<code>&reg;</code>	<code>&#174;</code>

Nota: els noms de les entitats distingeixen entre majúscules i minúscules.

Combinació de signes diacrítics

Un signe diacrític és un "glif" afegit a una lletra.

Alguns signes diacrítics, com greu (¨) i agut (´) s'anomenen accents.

Els signes diacrítics poden aparèixer tant sobre com baix d'una lletra, dins d'una lletra i entre dues lletres.

Els signes diacrítics es poden utilitzar en combinació amb caràcters alfanumèrics per produir un caràcter que no està present en el conjunt de caràcters (codificació) utilitzat a la pàgina.

Aquests són alguns exemples:

Mark	Character	Construct	Result
`	a	<code>a&#768;</code>	à
'	a	<code>a&#769;</code>	á
^	a	<code>a&#770;</code>	â
~	a	<code>a&#771;</code>	ã
`	O	<code>O&#768;</code>	Ò
'	O	<code>O&#769;</code>	Ó
^	O	<code>O&#770;</code>	Ô
~	O	<code>O&#771;</code>	Õ

32. Símbols HTML

Els símbols que no estan presents al teclat també es poden afegir utilitzant entitats.

Entitats de símbol HTML

Les entitats HTML es van descriure al punt anterior.

Molts símbols matemàtics, tècnics i de moneda no estan presents en un teclat normal.

Per afegir aquests símbols a una pàgina HTML, podeu utilitzar el nom de l'entitat o el número d'entitat (una referència decimal o hexadecimal) per al símbol.

Exemple: mostra el signe de l'euro, €, amb un nom d'entitat, un decimal i un valor hexadecimal.

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>
```

Alguns símbols matemàtics compatibles amb HTML

Char Number	Entity	Description
∀	∀ ∀	FOR ALL
∂	∂ ∂	PARTIAL DIFFERENTIAL
∃	∃ ∃	THERE EXISTS
∅	∅ ∅	EMPTY SETS
∇	∇ ∇	NABLA
∈	∈ ∈	ELEMENT OF
∉	∉ ∉	NOT AN ELEMENT OF
⊃	∋ ∋	CONTAINS AS MEMBER
∏	∏ ∏	N-ARY PRODUCT
Σ	∑ ∑	N-ARY SUMMATION

Algunes lletres gregues compatibles amb HTML

Char Number	Entity	Description
Α	Α Α	GREEK CAPITAL LETTER ALPHA
Β	Β Β	GREEK CAPITAL LETTER BETA
Γ	Γ Γ	GREEK CAPITAL LETTER GAMMA
Δ	Δ Δ	GREEK CAPITAL LETTER DELTA
Ε	Ε Ε	GREEK CAPITAL LETTER EPSILON
Ζ	Ζ Ζ	GREEK CAPITAL LETTER ZETA

Algunes altres entitats compatibles amb HTML

Char Number	Entity	Description
©	© ©	COPYRIGHT SIGN
®	® ®	REGISTERED SIGN
€	€ €	EURO SIGN
™	™ ™	TRADEMARK
←	← ←	LEFTWARDS ARROW
↑	↑ ↑	UPWARDS ARROW
→	→ →	RIGHTWARDS ARROW
↓	↓ ↓	DOWNWARDS ARROW
♠	♠ ♠	BLACK SPADE SUIT
♣	♣ ♣	BLACK CLUB SUIT
♥	♥ ♥	BLACK HEART SUIT
♦	♦ ♦	BLACK DIAMOND SUIT

33. Codificació HTML (conjunts de caràcters)

Per mostrar correctament una pàgina HTML, un navegador web ha de saber quin conjunt de caràcters utilitzar.

D'ASCII a UTF-8

ASCII va ser el primer estàndard de codificació de caràcters. ASCII va definir 128 caràcters diferents que es podrien utilitzar a Internet: números (0-9), lletres angleses (AZ) i alguns caràcters especials com ! \$ + - () @ < > .

ISO-8859-1 era el conjunt de caràcters predeterminat per a HTML 4. Aquest conjunt de caràcters admetia 256 codis de caràcters diferents. HTML 4 també suportava UTF-8.

ANSI (Windows-1252) era el conjunt de caràcters original de Windows. ANSI és idèntic a ISO-8859-1, excepte que ANSI té 32 caràcters addicionals.

L'especificació HTML5 anima els desenvolupadors web a utilitzar el conjunt de caràcters UTF-8, que cobreix gairebé tots els caràcters i símbols del món!

L'atribut del conjunt de caràcters HTML

Per mostrar correctament una pàgina HTML, un navegador web ha de conèixer el conjunt de caràcters utilitzat a la pàgina.

Això s'especifica a l'etiqueta `<meta>`:

```
<meta charset="UTF-8">
```

34. HTML versus XHTML

XHTML és una versió d'HTML més estricta i basada en XML.

Què és XHTML?

- XHTML significa E **X**tensible **H**yper **T**ext **M**arkup **L**anguage
- XHTML és una versió d'HTML més estricta i basada en XML
- XHTML és HTML definit com una aplicació XML
- XHTML és compatible amb tots els navegadors principals

Per què XHTML?

XML és un llenguatge de marques on tots els documents han d'estar marcats correctament (estar "ben formats").

XHTML es va desenvolupar per fer HTML més extensible i flexible per treballar amb altres formats de dades (com ara XML). A més, els navegadors ignoren els errors a les pàgines HTML i intenten mostrar el lloc web encara que tinga alguns errors en el marcatge. Així, XHTML inclou un tractament d'errors molt més estricte.

Les diferències més importants amb l'HTML

- `<!DOCTYPE>` és **obligatori**
- L'atribut `xmlns` a `<html>` és **obligatori**
- `<html>`, `<head>`, `<title>` i `<body>` són **obligatoris**
- Els elements sempre s'han d' **anar correctament**
- Els elements han d'estar sempre **tancats**
- Els elements han d'estar sempre en **minúscula**
- Els noms dels atributs sempre han d'estar en **minúscules**
- Els valors dels atributs sempre s'han de **citar**

- La minimització d'atributs està **prohibida**

XHTML - <!DOCTYPE> És obligatori

Un document XHTML ha de tenir una declaració XHTML <!DOCTYPE>.

Els elements <html>, <head>, <title> i <body> també han d'estar presents, i l'atribut xmlns a <html> ha d'especificar l'espai de noms xml per al document. Exemple:

Ací teniu un document XHTML amb un mínim d'etiquetes requerides:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Title of document</title>
</head>
<body>

  some content here...

</body>
</html>
```

Els elements XHTML s'han d'anar correctament imbrincats

En XHTML, els elements sempre s'han d'imbricar correctament entre si.

Els elements XHTML han d'estar sempre tancats

En XHTML, els elements sempre han d'estar tancats.

Els elements buits XHTML s'han de tancar sempre

En XHTML, els elements buits sempre s'han de tancar.

Els elements XHTML han d'estar en minúscules

En XHTML, els noms dels elements sempre han d'estar en minúscules.

Els noms dels atributs XHTML han d'estar en minúscules

En XHTML, els noms dels atributs sempre han d'estar en minúscules.

Els valors dels atributs XHTML s'han de citar

En XHTML, els valors dels atributs sempre s'han de citar.

La minimització d'atributs XHTML està prohibida

En XHTML, la minimització d'atributs està prohibida.

35. Formularis HTML

S'utilitza un formulari HTML per recollir les entrades de l'usuari. L'entrada de l'usuari s'envia sovint a un servidor per processar-la. Exemple:

John

Doe

Submit

L'element <form>

L'element **<form>** HTML s'utilitza per crear un formulari HTML per a l'entrada de l'usuari:

```
<form>
.
form elements
.
</form>
```

L'element **<form>** és un contenidor per a diferents tipus d'elements d'entrada, com ara: camps de text, caselles de selecció, botons d'opció, botons d'enviament, etc.

L'element <input>

L'element HTML **<input>** és l'element de formulari més utilitzat.

Un element **<input>** es pot mostrar de moltes maneres, depenent de l'atribut **type**.

Aquests són alguns exemples:

Tipus	Descripció
<code><input type="text"></code>	Mostra un camp d'entrada de text d'una sola línia
<code><input type="radio"></code>	Mostra un botó d'opció (per seleccionar una de moltes opcions)
<code><input type="checkbox"></code>	Mostra una casella de selecció (per seleccionar zero o més de moltes opcions)
<code><input type="submit"></code>	Mostra un botó d'enviament (per enviar el formulari)
<code><input type="button"></code>	Mostra un botó en què es pot fer clic

Camps de text

El **<input type="text">** defineix un camp d'entrada d'una sola línia per a l'entrada de text. Exemple. Un formulari amb camps d'entrada de text:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

Així és com es mostrarà el codi HTML anterior en un navegador:

First name:

Last name:

Nota: el formulari en si no és visible. Tingueu en compte també que l'amplada predeterminada d'un camp d'entrada és de 20 caràcters.

L'element `<label>`

Observeu l'ús de l'element `<label>` a l'exemple anterior.

L'etiqueta `<label>` defineix una etiqueta per a molts elements del formulari.

L'element `<label>` és útil per als usuaris de lectors de pantalla, perquè el lector de pantalla llegirà en veu alta l'etiqueta quan l'usuari se centra en l'element d'entrada.

L'element `<label>` també ajuda els usuaris que tenen dificultats per fer clic a regions molt xicotetes (com ara botons d'opció o caselles de verificació), perquè quan l'usuari fa clic al text de l'element `<label>`, canvia el botó d'opció o la casella de selecció.

L'atribut `for` de l'etiqueta `<label>` ha de ser igual a l'atribut `id` de l' `<input>` element per unir-los.

Botons de ràdio

El `<input type="radio">` defineix un botó d'opció.

Els botons d'opció permeten a l'usuari seleccionar UNA d'un nombre limitat d'opcions.

Exemple. Un formulari amb botons d'opció. Prova el codi per veure el resultat:

```
<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="js" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```

Caselles de verificació

El `<input type="checkbox">` defineix una **casella de selecció**.

Les caselles de selecció permeten a l'usuari seleccionar ZERO o MÉS opcions d'un nombre limitat d'opcions.

Exemple. Un formulari amb caselles de selecció:

```
<form action="/action_page.php">
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
```

```
<label for="vehicle3"> I have a boat</label><br><br>
<input type="submit" value="Submit">
</form>
```

El botó d'enviament

El `<input type="submit">` defineix un botó per enviar les dades del formulari a un gestor de formularis.

El gestor de formularis és normalment un fitxer al servidor amb un script per processar les dades d'entrada.

El gestor de formularis s'especifica a l'atribut `action` del formulari.

Exemple. Un formulari amb un botó d'enviament:

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Send">
</form>
```

L'atribut de name per a <entrada>

Tingueu en compte que cada camp d'entrada ha de tenir un atribut `name` per ser enviat.

Si s'omet l'atribut `name`, el valor del camp d'entrada no s'enviarà en absolut.

36. Atributs de formulari HTML

L'atribut action

L'atribut `action` defineix l'acció que s'ha de realitzar quan s'envia el formulari.

Normalment, les dades del formulari s'envien a un fitxer del servidor quan l'usuari fa clic al botó d'enviament.

A l'exemple següent, les dades del formulari s'envien a un fitxer anomenat "action_page.php". Aquest fitxer conté un script del servidor que gestiona les dades del formulari:

Exemple. En enviar, envieu les dades del formulari a "action_page.php":

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Forms</h2>

<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>

<p>If you click the "Submit" button, the form-data will be sent to a page
called "/action_page.php".</p>
```

```
</body>
</html>
```

Consell: si s'omet l'atribut **action**, l'acció s'estableix a la pàgina actual.

L'atribut target

L'atribut **target** especifica on mostrar la resposta que es rep després d'enviar el formulari.

L'atribut **target** pot tenir un dels valors següents:

Value	Description
<code>_blank</code>	The response is displayed in a new window or tab
<code>_self</code>	The response is displayed in the current window
<code>_parent</code>	The response is displayed in the parent frame
<code>_top</code>	The response is displayed in the full body of the window
<code>framename</code>	The response is displayed in a named iframe

El valor predeterminat és **_self** la qual cosa significa que la resposta s'obrirà a la finestra actual.

Exemple. Aquí, el resultat enviat s'obrirà en una nova pestanya del navegador:

```
<form action="action_page.php" target="_blank">
```

L'atribut del method

L'atribut **method** especifica el mètode HTTP que s'utilitzarà en enviar les dades del formulari.

Les dades del formulari es poden enviar com a variables URL (amb **method="get"**) o com a transacció posterior HTTP (amb **method="post"**).

El mètode HTTP predeterminat quan s'envien dades del formulari és GET.

Exemple. Aquest exemple utilitza el mètode GET quan s'envien les dades del formulari:

```
<form action="action_page.php" method="get">
```

Exemple. Aquest exemple utilitza el mètode POST quan s'envien les dades del formulari:

```
<form action="/action_page.php" method="post">
```

Notes sobre GET:

- Afegeix les dades del formulari a l'URL, en parells nom/valor
- NO utilitzeu MAI GET per enviar dades sensibles! (les dades del formulari enviat són visibles a l'URL!)
- La longitud d'un URL és limitada (2048 caràcters)
- Útil per a l'enviament de formularis on un usuari vol marcar el resultat
- GET és bo per a dades no segures, com ara les cadenes de consulta a Google

Notes sobre POST:

- Afegeix les dades del formulari dins del cos de la sol·licitud HTTP (les dades del formulari enviat no es mostren a l'URL)
- POST no té limitacions de mida i es pot utilitzar per enviar grans quantitats de dades.
- Els enviaments de formularis amb POST no es poden afegir a favorits

Consell: feu servir sempre POST si les dades del formulari contenen informació sensible o personal.

L'atribut d'autocomplete

L'atribut **autocomplete** especifica si un formulari ha d'activar o desactivar l'emplenament automàtic.

Quan l'emplenament automàtic està activat, el navegador completa automàticament els valors en funció dels valors que l'usuari haja introduït abans.

Exemple. Un formulari amb autocomplete a:

```
<form action="action_page.php" autocomplete="on">
```

L'atribut novalidate

L'atribut **novalidate** és un atribut booleà. Quan està present, especifica que les dades del formulari (entrada) no s'han de validar quan s'envien.

Exemple. Un formulari amb un atribut novalidate:

```
<form action="action_page.php" novalidate>
```

Llista de tots els atributs <form>

Atribut	Descripció
accept-charset	Especifica les codificacions de caràcters utilitzades per a l'enviament del formulari
accion	Especifica on enviar les dades del formulari quan s'envii un formulari
autocomplete	Especifica si un formulari ha d'activar o desactivar l'emplenament automàtic
enctype	Especifica com s'han de codificar les dades del formulari en enviar-les al servidor (només per a method="post")
metod	Especifica el mètode HTTP que cal utilitzar quan s'envien dades de formulari
name	Especifica el nom del formulari
novalidate	Especifica que el formulari no s'ha de validar quan s'envii
rel	Especifica la relació entre un recurs enllaçat i el document actual
Target	Especifica on mostrar la resposta que es rep després d'enviar el formulari

37. Elements del formulari HTML

L'element HTML **<form>** pot contenir un o més dels elements de formulari següents:

- **<input>**
- **<label>**
- **<select>**
- **<textarea>**
- **<button>**
- **<fieldset>**
- **<legend>**
- **<datalist>**
- **<output>**
- **<option>**
- **<optgroup>**

L'element <input>

Un dels elements de forma més utilitzats és l'element **<input>**.

L'element **<input>** es pot mostrar de diverses maneres, depenent de l'atribut **type**.

```
<label for="fname">First name:</label>
<input type="text" id="fname" name="fname">
```

Tots els diferents valors de l' **type** atribut els vorem al punt següent:

L'element **<label>**

L'element **<label>** defineix una etiqueta per a diversos elements del formulari.

L'element **<label>** és útil per als usuaris de lectors de pantalla, perquè el lector de pantalla llegirà en veu alta l'etiqueta quan l'usuari se centre en l'element d'entrada.

L'element **<label>** també ajuda els usuaris que tenen dificultats per fer clic en regions molt xicotetes (com ara botons d'opció o caselles de selecció), perquè quan l'usuari fa clic al text de l'element **<label>**, activa el botó d'opció o la casella de selecció.

L'atribut **for** de l'etiqueta **<label>** ha de ser igual a l'atribut **id** de l'element **<input>** per unir-los.

L'element **<select>**

L'element **<select>** defineix una llista desplegable. Exemple:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

L'element **<option>** defineix una opció que es pot seleccionar.

Per defecte, el primer element de la llista desplegable està seleccionat.

Per definir una opció preseleccionada, afegiu l'atribut **selected** a l'opció. Exemple:

```
<option value="fiat" selected>Fiat</option>
```

Valors visibles:

Utilitzeu l'atribut **size** per especificar el nombre de valors visibles. Exemple:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Permet seleccions múltiples:

Utilitzeu l'atribut **multiple** per permetre a l'usuari seleccionar més d'un valor. Exemple:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

L'element <textarea>

L'element <textarea> defineix un camp d'entrada de diverses línies (una àrea de text):

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

L'atribut **rows** especifica el nombre visible de línies en una àrea de text.

L'atribut **cols** especifica l'amplada visible d'una àrea de text.

També podeu definir la mida de l'àrea de text mitjançant CSS. Exemple:

```
<textarea name="message" style="width:200px; height:600px;">
The cat was playing in the garden.
</textarea>
```

L'element <button>

L'element <button> defineix un botó clicable. Exemple:

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

Els elements <fieldset> i <legend>

L'element <fieldset> s'utilitza per agrupar dades relacionades en un formulari.

L'element <legend> defineix un títol per a l'element <fieldset>. Exemple:

```
<form action="action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

L'element <datalist>

L'element <datalist> especifica una llista d'opcions predefinides per a un element <input>.

Els usuaris veuran una llista desplegable de les opcions predefinides mentre introdueixen dades.

L'atribut **list** de l'element <input>, ha de fer referència a l'atribut **id** de l'element <datalist>. Exemple:

```
<form action="action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

L'element <output>

L'element <output> representa el resultat d'un càlcul (com el realitzat per un script).

```
<!DOCTYPE html>
<html>
<body>

<h2>The output Element</h2>
<p>The output element represents the result of a calculation.</p>

<form action="/action_page.php"
oninput="x.value=parseInt(a.value)+parseInt(b.value)"> 0
  <input type="range" id="a" name="a" value="50"> 100 +
  <input type="number" id="b" name="b" value="50"> =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>

</body>
</html>
```

Elements del formulari HTML

Tag	Description
<u><form></u>	Defines an HTML form for user input
<u><input></u>	Defines an input control
<u><textarea></u>	Defines a multiline input control (text area)
<u><label></u>	Defines a label for an <input> element
<u><fieldset></u>	Groups related elements in a form
<u><legend></u>	Defines a caption for a <fieldset> element
<u><select></u>	Defines a drop-down list
<u><optgroup></u>	Defines a group of related options in a drop-down list
<u><option></u>	Defines an option in a drop-down list
<u><button></u>	Defines a clickable button
<u><datalist></u>	Specifies a list of pre-defined options for input controls
<u><output></u>	Defines the result of a calculation

38. Tipus de dades d'entrada en formularis HTML

En aquest punt es descriu els diferents tipus de l'element <input> HTML.

Tipus d'entrada HTML

Aquests són els diferents tipus d'entrada que podeu utilitzar en HTML:

- <input type="button">
- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="image">
- <input type="month">
- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text">
- <input type="time">
- <input type="url">

- `<input type="week">`

Consell: el valor predeterminat de l'atribut `type` és "text".

Restriccions d'entrada

Ací hi ha una llista d'algunes restriccions d'entrada habituals:

Attribute	Description
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

39. Atributs d'entrada HTML

Aquest punt descriu els diferents atributs de l'element `<input>` HTML.

L'Atribut value

L'atribut `value` d'entrada especifica un valor inicial per a un camp d'entrada:

Exemple. Camps d'entrada amb valors inicials (per defecte):

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

L'atribut de només lectura

L'atribut `readonly` d'entrada especifica que un camp d'entrada és només de lectura.

No es pot modificar un camp d'entrada només de lectura (no obstant això, un usuari pot fer-hi tabulació, ressaltar-lo i copiar-ne el text).

El valor d'un camp d'entrada només de lectura s'enviarà en enviar el formulari!

Exemple. Un camp d'entrada de només lectura:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" readonly><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

L'atribut desactivat

L'atribut **disabled** d'entrada especifica que s'ha de desactivar un camp d'entrada.

Un camp d'entrada desactivat és inutilitzable i no es pot fer clic.

El valor d'un camp d'entrada desactivat no s'enviarà en enviar el formulari!

Exemple. Un camp d'entrada desactivat:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" disabled><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

L'atribut de mida

L'atribut **size** d'entrada especifica l'amplada visible, en caràcters, d'un camp d'entrada.

El valor per defecte de **size** és 20.

Nota: l'atribut **size** funciona amb els tipus d'entrada següents: text, cerca, tel, url, correu electrònic i contrasenya.

Exemple. Estableix una amplada per a un camp d'entrada:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" size="4">
</form>
```

L'atribut maxlength

L'atribut **maxlength** d'entrada especifica el nombre màxim de caràcters permès en un camp d'entrada.

Nota: quan **maxlength** s'estableix a, el camp d'entrada no acceptarà més del nombre de caràcters especificat. Tanmateix, aquest atribut no proporciona cap comentari. Per tant, si voleu avisar l'usuari, heu d'escriure codi JavaScript.

Exemple. Estableix una longitud màxima per a un camp d'entrada:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" maxlength="4" size="4">
</form>
```

Els atributs mínim i màxim

L'entrada **min** i **max** els atributs especifiquen els valors mínims i màxims per a un camp d'entrada.

Els atributs **min** i **max** funcionen amb els tipus d'entrada següents: nombre, interval, data, datahora local, mes, hora i setmana.

Consell: utilitzeu els atributs max i min junts per crear una sèrie de valors legals.

Exemple. Estableix una data màxima, una data mínima i un interval de valors legals:

```
<form>
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>

  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>

  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```

L'atribut múltiple

L'atribut **multiple** d'entrada especifica que l'usuari pot introduir més d'un valor en un camp d'entrada.

L'atribut **multiple** funciona amb els tipus d'entrada següents: correu electrònic i fitxer.

Exemple. Un camp de càrrega de fitxers que accepta diversos valors:

```
<form>
  <label for="files">Select files:</label>
  <input type="file" id="files" name="files" multiple>
</form>
```

L'atribut del pattern

L'atribut **pattern** d'entrada especifica una expressió regular amb la qual es contrasta el valor del camp d'entrada quan s'envia el formulari.

L'atribut **pattern** funciona amb els tipus d'entrada següents: text, data, cerca, url, tel, correu electrònic i contrasenya.

Exemple. Un camp d'entrada que només pot contenir tres lletres (sense números ni caràcters especials):

```
<form>
  <label for="country_code">Country code:</label>
  <input type="text" id="country_code" name="country_code"
    pattern="[A-Za-z]{3}" title="Three letter country code">
</form>
```

L'atribut de marcador de posició

L'atribut **placeholder** d'entrada especifica una pista breu que descriu el valor esperat d'un camp d'entrada (un valor de mostra o una breu descripció del format esperat).

La pista breu es mostra al camp d'entrada abans que l'usuari introdueixi un valor.

L'atribut **placeholder** funciona amb els tipus d'entrada següents: text, cerca, URL, tel, correu electrònic i contrasenya.

Exemple. Un camp d'entrada amb un text de marcador de posició:

```
<form>
  <label for="phone">Enter a phone number:</label>
  <input type="tel" id="phone" name="phone"
    placeholder="123-45-678"
    pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

L'atribut requerit

L'atribut **required** d'entrada especifica que s'ha d'omplir un camp d'entrada abans d'enviar el formulari.

L'atribut **required** funciona amb els tipus d'entrada següents: text, cerca, URL, tel, correu electrònic, contrasenya, selectores de dates, número, casella de selecció, ràdio i fitxer.

Exemple. Un camp d'entrada obligatori:

```
<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>
</form>
```

L'atribut de step

L'atribut **step** d'entrada especifica els intervals de nombre legal per a un camp d'entrada.

Exemple: si `step="3"`, els números legals podrien ser -3, 0, 3, 6, etc.

Consell: aquest atribut es pot utilitzar juntament amb els atributs `max` i `min` per crear una sèrie de valors legals.

L'atribut **step** funciona amb els tipus d'entrada següents: nombre, interval, data, datahora-local, mes, hora i setmana.

Exemple. Un camp d'entrada amb un interval de nombre legal especificat:

```
<form>
  <label for="points">Points:</label>
  <input type="number" id="points" name="points" step="3">
</form>
```

les restriccions d'entrada no són infal·libles i JavaScript ofereix moltes maneres d'afegir entrades il·legals. Per restringir l'entrada de manera segura, també l'ha de comprovar el receptor (el servidor)!

L'atribut d'enfocament automàtic

L'atribut **autofocus** d'entrada especifica que un camp d'entrada hauria de centrar-se automàticament quan es carrega la pàgina.

Exemple. Permet que el camp d'entrada "Nom" es concentre automàticament quan es carregue la pàgina:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" autofocus><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Els atributs d'alçada i amplada

L'entrada **height** i **width** els atributs especifiquen l'alçada i l'amplada d'un `<input type="image">` element.

Consell: especifiqueu sempre els atributs d'alçada i amplada de les imatges. Si s'estableixen alçada i amplada, l'espai necessari per a la imatge es reserva quan es carrega la pàgina. Sense aquests atributs, el navegador no coneix la mida de la imatge i no pot

reservar-li l'espai adequat. L'efecte serà que el disseny de la pàgina canviarà durant la càrrega (mentre es carreguen les imatges).

Exemple. Definiu una imatge com a botó d'enviament, amb atributs d'alçada i amplada:

```
<form>
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="image" src="img_submit.gif" alt="S" width="48" height="48">
</form>
```

L'atribut de la llista

L'atribut **list** d'entrada fa referència a un **<datalist>** element que conté opcions predefinides per a un element **<input>**.

Exemple. Un element **<input>** amb valors predefinits en una **<datalist>**:

```
<form>
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

L'atribut d'emplenament automàtic

L'atribut **autocomplete** d'entrada especifica si un formulari o un camp d'entrada ha d'activar o desactivar l'emplenament automàtic.

L'emplenament automàtic permet al navegador predir el valor. Quan un usuari comença a escriure un camp, el navegador hauria de mostrar opcions per omplir el camp, basant-se en els valors escrits anteriorment.

L'atribut **autocomplete** funciona amb **<form>** i els **<input>** tipus següents: text, cerca, URL, tel, correu electrònic, contrasenya, selectors de dates, rang i color.

Exemple. Un formulari HTML amb l'emplenament automàtic activat i desactivat per a un camp d'entrada:

```
<form action="/action_page.php" autocomplete="on">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" autocomplete="off"><br><br>
  <input type="submit" value="Submit">
</form>
```



Índex

- [1. Definició](#)
 - [2. Història](#)
 - [3. Funcionament dels CMS](#)
 - [4. Tipus de CMS](#)
 - [5. Instal·lar WordPress i Drupal](#)
 - [6. Crear un lloc web amb WordPress](#)
-

1. Definició

Un CMS (Sistema de Gestió de Continguts, en anglès Content Management System) és una plataforma de programari que facilita la creació, gestió i publicació de continguts en línia de manera eficient i sense requerir coneixements avançats de programació.

Aquestes eines són àmpliament utilitzades per a crear i gestionar llocs web, blogs, portals d'informació i altres plataformes digitals. Ací tens una definició més detallada i exemples de CMS:

Un CMS consta de dues parts principals:

Interfície d'usuari: Proporciona una plataforma fàcil d'utilitzar, similar a un editor de text, on els usuaris poden crear i editar continguts, com textos, imatges, vídeos i altres elements multimèdia. Aquesta interfície sol ser accessible a través d'un navegador web.

Base de dades: Emmagatzema tots els continguts creats i les dades del lloc web, com ara textos, imatges i configuracions. Aquesta base de dades facilita la gestió i l'organització de continguts i permet una publicació ràpida i àgil.

Exemples de CMS populars inclouen:

WordPress: És un dels CMS més populars i àmpliament utilitzats. Inicialment creat per a blogs, s'ha convertit en una plataforma versàtil per a qualsevol tipus de lloc web, des de blogs fins a llocs web d'empreses.

Joomla: Aquest CMS és conegut per la seva flexibilitat i capacitat de personalització. És una opció popular per a la creació de llocs web més complexos i amb funcionalitats avançades.

Drupal: Ofereix una gran flexibilitat i escalabilitat, essent adequat per a llocs web corporatius i comunitats en línia que necessiten característiques avançades i personalitzacions.

Magento: Especialitzat en comerç electrònic, Magento és una opció popular per crear botigues en línia amb característiques avançades per a la gestió de productes i pagaments.

Wix: Amb un enfocament en la simplicitat, Wix és adequat per a xicotets negocis i professionals que volen crear ràpidament llocs web amb un aspecte atractiu.

Aquests CMS simplifiquen la creació i la gestió de continguts en línia, permetent que propietaris de llocs web i usuaris sense coneixements tècnics puguin actualitzar i mantenir els seus llocs web amb facilitat.

2. Història

La història dels CMS (Sistemes de Gestió de Continguts) és una part integral de la història de la web i ha evolucionat significativament des dels seus inicis. Ací tens una visió general de la història dels CMS:

Dècada de 1990: Els primers llocs web es creaven manualment amb HTML i codificació personalitzada. No hi havia CMS en aquesta època, i la creació de llocs web era un procés complex i laboriós.

Dècada de 2000: Apareixen els primers CMS. L'any 2001 es llança Drupal com a projecte de codi obert. També apareix Joomla (anteriorment conegut com Mambo) i WordPress, que inicialment es centra en la creació de blogs.

Mitjans de la dècada de 2000: Els CMS comencen a guanyar popularitat com a eines per a la creació de llocs web. Els usuaris poden crear i gestionar continguts sense necessitat de coneixements avançats de programació.

Dècada de 2010: Els CMS continuen creixent en popularitat. WordPress es converteix en el CMS més utilitzat a nivell mundial, impulsat per la seva flexibilitat i una gran comunitat de desenvolupadors i usuaris.

Dècada de 2020: Els CMS segueixen sent una part fonamental de la web. S'han desenvolupat molts altres CMS especialitzats per a diferents necessitats, com Magento per al comerç electrònic i Ghost per a blogs.

Aquesta evolució ha estat impulsada per la necessitat de fer que la creació i la gestió de llocs web siguin més accessibles per a una àmplia gamma d'usuaris. Els CMS han evolucionat per oferir funcionalitats més avançades, plantilles personalitzades i la possibilitat d'afegir extensions i plugins per ampliar les seues capacitats. Huí en dia, els CMS continuen sent una eina essencial per a la creació i la gestió de continguts en línia.

3. Funcionament dels CMS

El funcionament dels CMS (Sistemes de Gestió de Continguts, en anglès Content Management Systems) es basa en proporcionar una plataforma que facilita la creació, l'edició i la gestió de continguts en línia. Aquí tens una descripció del funcionament d'un CMS:

Interfície d'Usuari: Els CMS ofereixen una interfície d'usuari amigable i intuïtiva que es pot accedir a través d'un navegador web. Aquesta interfície permet als usuaris crear, editar i gestionar continguts sense necessitat de coneixements de programació.

Editor de Continguts: Els usuaris poden utilitzar l'editor de continguts per a crear i editar textos, imatges, vídeos i altres elements multimèdia. L'editor sovint és semblant a un processador de textos i facilita afegir format, enllaços i altres elements.

Emmagatzematge en Base de Dades: Totes les dades, com ara textos, imatges i configuracions, es guarden en una base de dades. Això permet una gestió centralitzada i l'organització eficaç del contingut.

Plantilles i Temes: Els CMS ofereixen plantilles i temes predefinits que defineixen l'aparença del lloc web. Els usuaris poden seleccionar o personalitzar aquestes plantilles segons les seues necessitats i preferències visuals.

Gestió de Mitjans: Els CMS faciliten la gestió d'arxius multimèdia com imatges, vídeos i documents. Aquests fitxers es poden carregar i gestionar directament des de la interfície del CMS.

Gestió d'Usuaris i Permisos: Els CMS ofereixen funcionalitats per a la gestió d'usuaris amb diferents nivells d'accés i permissos. Això permet controlar qui pot crear, editar i publicar continguts.

Publicació de Continguts: Una vegada es crea o edita el contingut, es pot programar la seua publicació per a una data i hora determinades. Això permet la planificació de continguts i actualitzacions.

Extensibilitat: Molts CMS ofereixen la possibilitat d'afegir extensions o plugins que amplien les funcionalitats del sistema. Això permet personalitzar i millorar la funcionalitat del lloc web.

SEO i Optimització: Molts CMS inclouen eines per a l'optimització dels motors de cerca (SEO), com la configuració de metadades i la generació d'URLs amigables.

Seguretat: Els CMS incorporen mesures de seguretat per protegir el lloc web contra amenaces com ara atacs de hackers i malware.

Comunitat i Suport: Molts CMS tenen una gran comunitat d'usuaris i desenvolupadors que proporcionen suport i recursos en línia, com fòrums i documentació.

En resum, els CMS simplifiquen la creació i la gestió de llocs web mitjançant una interfície fàcil d'utilitzar i eines per gestionar continguts, disseny i funcionalitats sense requerir coneixements avançats de programació. Aquesta facilitat d'ús fa que els CMS siguin una opció popular per a l'administració de llocs web de tot tipus.

4. Tipus de CMS

Els tipus de sistemes de gestió de continguts (CMS) més comuns són:

CMS d'ús general: Aquesta categoria inclou CMS ampliament utilitzats com WordPress, Joomla i Drupal, que són versàtils i poden adaptar-se a una àmplia gamma de necessitats.

CMS especialitzats: Hi ha CMS dissenyats específicament per a àrees com comerç electrònic, com Magento i WooCommerce, o per a blocs, com Blogger.

CMS de codi obert: Aquests CMS són gratuïts i tenen una gran comunitat d'usuaris i desenvolupadors que ofereixen extensions i suport. Exemples són WordPress i Joomla.

CMS comercials: Aquests CMS solen requerir una llicència o una tarifa per a l'ús i el suport. Algunes opcions inclouen Adobe Experience Manager i Sitecore.

CMS basats en la núvol: Aquests CMS es basen en serveis web i emmagatzematge en línia. Google Sites i Wix són exemples populars.

CMS lligats a plataformes de comerç electrònic: Si vols gestionar una botiga en línia, Shopify i BigCommerce són CMS populars que ofereixen eines d'e-commerce.

CMS d'empreses: Empreses més grans podrien utilitzar CMS com Liferay o SharePoint per gestionar el seu contingut i col·laboració interna.

CMS de blocs: Aquest tipus de CMS està dissenyat específicament per a blocs i inclou opcions com Ghost i Medium.

CMS de portals: Aquests CMS estan enfocats en la creació de portals web i intranets, i inclouen opcions com Plone i Liferay.

CMS de microserveis: Aquest tipus de CMS està dissenyat per a entorns de microserveis i distribuïts, com Contentful i Strapi.

CMS sense cap interfície d'usuari (Headless CMS): Aquests CMS ofereixen contingut a través d'API i són molt flexibles. Algunes opcions són Contentful i Kentico Kontent.

Cada tipus de CMS té les seves pròpies característiques i avantatges, i la elecció depèn de les teues necessitats específiques i de les teues habilitats de desenvolupador.

5. Instal·lar WordPress i Drupal

<https://www.youtube.com/channel/UCEONRyyFdRqIEJeu7m6gXBw>

6. Crear un lloc web amb WordPress

Vorem un manual d'ús de la versió més recent i plantejarem un projecte de creació d'un lloc web.

Tema 4: Sistemes de gestió de l'aprenentatge



Índex

- [1. Què són els sistemes de gestió d'aprenentatge LMS](#)
 - [2. Característiques comunes dels LMS](#)
 - [3. Principals LMS](#)
 - [4. Google Classroom](#)
 - [5. Moodle](#)
 - [6. Instal·lar Moodle](#)
 - [7. Aules](#)
-

1. Què són els sistemes de gestió d'aprenentatge LMS

Els Sistemes de Gestió de l'Aprenentatge (LMS, per les seves sigles en anglès) són plataformes o sistemes que faciliten la gestió, l'organització i l'entrega de contingut d'aprenentatge en línia. Aquests sistemes són àmpliament utilitzats en institucions educatives i empreses per a la formació en línia i el seguiment de l'aprenentatge dels estudiants o dels empleats. Les seues funcions principals inclouen:

Gestió de cursos: Permeten crear, organitzar i oferir cursos i contingut d'aprenentatge en línia, incloent-hi material com vídeos, documents i qüestionaris.

2. Característiques comunes dels LMS

Seguiment de l'aprenentatge: Registren i supervisen l'aprenentatge dels usuaris, com ara la progressió dels cursos, les puntuacions dels qüestionaris i el temps dedicat a cada mòdul.

Avaluació i avaluació: Proporcionen eines per a l'avaluació i l'avaluació de l'aprenentatge, com ara qüestionaris, proves i tasques.

Gestió d'usuaris: Administren perfils d'usuaris, registres d'usuaris, inscripcions a cursos i control d'accés.

Comunicació i col·laboració: Faciliten la comunicació entre els usuaris i els instructors, ja sigui a través de xats, fòrums o altres eines de col·laboració.

Informes i analítiques: Generen informes i analítiques per avaluar el rendiment dels usuaris i l'eficàcia del contingut d'aprenentatge.

Personalització: Permeten adaptar l'aprenentatge a les necessitats individuals dels estudiants o els empleats, com ara proporcionar contingut específic basat en el seu progrés.

Integració amb altres sistemes: Poden integrar-se amb altres aplicacions i sistemes, com ara sistemes de gestió de recursos humans (HRMS) o sistemes d'informació de l'institut (SII).

Els LMS són molt útils en àmplies varietats d'entorns d'aprenentatge, des de l'educació formal fins a la formació corporativa i continuada. Ajuden a automatitzar i millorar el procés d'aprenentatge en línia, proporcionant una plataforma central per a la gestió i el seguiment de l'aprenentatge.

3. Principals LMS

Els Sistemes de Gestió de l'Aprenentatge (LMS) són unes eines importants per a l'educació i la formació en línia. Algunes de les principals plataformes LMS són:

Moodle: Moodle és una plataforma LMS de codi obert àmpliament utilitzada a nivell mundial. Ofereix una àmplia gamma de funcionalitats per crear i gestionar cursos en línia.

Canvas LMS: Canvas és una plataforma LMS popular desenvolupada per Instructure. És àmpliament utilitzada en institucions educatives i empreses per a la formació en línia.

Blackboard Learn: Blackboard és un dels noms més coneguts en el món de l'educació en línia. La seva plataforma LMS, Blackboard Learn, ofereix eines avançades per a la gestió de cursos i el contingut d'aprenentatge.

Sakai: Sakai és un altre sistema de codi obert dissenyat per a l'educació superior. Ofereix eines col·laboratives i de gestió de cursos.

Adobe Captivate Prime: Aquest LMS està dissenyat per a la formació corporativa i està relacionat amb les solucions d'Adobe per a la creació de contingut d'aprenentatge.

Totara Learn: Totara Learn és una plataforma LMS de codi obert dissenyada específicament per a la formació corporativa i de recursos humans.

Brightspace by D2L: Brightspace és una plataforma LMS que ofereix una àmplia gamma de funcions per a l'educació i la formació en línia.

iSpring Learn: Aquest LMS es centra en la creació i distribució de contingut d'aprenentatge, especialment per a empreses.

Edmodo: Edmodo és una plataforma LMS que es centra en l'educació primària i secundària, i facilita la comunicació entre estudiants, mestres i pares.

Google Classroom: Aquesta plataforma gratuïta es centra en l'educació i la col·laboració en línia, i està estretament integrada amb les eines de Google.

Aquestes són algunes de les principals opcions LMS disponibles, i la elecció depèn de les necessitats específiques de l'organització o institució que les utilitza. Cada una d'aquestes plataformes té les seves pròpies característiques i avantatges per a la gestió de cursos i l'aprenentatge en línia.

4. Google Classroom

Google Classroom és una plataforma de gestió de l'aprenentatge (LMS) desenvolupada per Google que facilita la creació i administració de cursos en línia, especialment dissenyada per a l'entorn educatiu. A continuació, et proporciono una visió general de què és i com es utilitza Google Classroom:

Què és Google Classroom:

Google Classroom és una eina gratuïta que permet als docents i professors crear un espai virtual on poden compartir materials d'aprenentatge, assignar tasques, rebre treballs dels estudiants, avaluar-los i comunicar-se amb els alumnes. Aquests són alguns dels punts destacats de Google Classroom:

Gestió de cursos: Els docents poden crear cursos en línia i convidar alumnes mitjançant codis d'accés. Cada curs té la seva pròpia pàgina on es poden organitzar totes les activitats relacionades amb el curs.

Materials d'aprenentatge: Els docents poden carregar i compartir materials d'aprenentatge com documents, presentacions, vídeos i enllaços web amb els estudiants.

Assignació de tasques: Els docents poden crear tasques i avaluar-les mitjançant Google Classroom. Les tasques poden incloure preguntes, puntuacions i dates límit.

Rebre treballs dels estudiants: Els estudiants poden enviar els seus treballs a través de Google Classroom, i els docents poden avaluar-los i donar retroacció.

Comunicació: Google Classroom facilita la comunicació entre docents i estudiants a través de comentaris en les tasques i anuncis. També es poden enviar missatges privats.

Calendari integrat: Google Classroom està integrat amb Google Calendar, el que facilita la programació d'activitats i tasques.

Com utilitzar Google Classroom:

Per començar a utilitzar Google Classroom, segueix aquests passos:

Inicia la sessió amb un compte de Google (necessites un compte de Google per a utilitzar Google Classroom).

Accedeix a Google Classroom a través de la pàgina web o de l'aplicació mòbil.

Crea un nou curs i configura la informació del curs, com el nom, la descripció i les dates límit.

Convida els estudiants al curs mitjançant codis d'accés o les seues adreces de correu electrònic.

Carrega materials d'aprenentatge i crea tasques per als estudiants.

Avalua les tasques que envien els estudiants i proporciona retroacció.

Comunica't amb els estudiants a través de les funcions de comentari i missatges.

Utilitza el calendari integrat per programar activitats i recordatoris.

Google Classroom és una eina potent per a la gestió de cursos en línia i és àmpliament utilitzada a nivell mundial, tant en entorns educatius com corporatius. És fàcil d'aprendre a utilitzar i ofereix una manera eficaç d'organitzar i gestionar l'aprenentatge en línia.

5. Moodle

Moodle és una plataforma de gestió de l'aprenentatge (LMS) de codi obert que es fa servir àmpliament en àmbits educatius i corporatius per a la creació i administració de cursos en línia. Aquí tens una descripció i com s'utilitza Moodle:

Què és Moodle:

Moodle és un sistema de gestió de l'aprenentatge que permet als educadors i formadors crear cursos virtuals i gestionar tots els aspectes relacionats amb l'aprenentatge en línia. Moodle és una plataforma de codi obert, la qual cosa significa que es pot personalitzar i adaptar a les necessitats específiques de cada organització o docent. Alguns dels punts destacats de Moodle són:

Gestió de cursos: Amb Moodle, pots crear cursos virtuals i organitzar-los de manera efectiva. Aquesta plataforma ofereix eines per a la creació de cursos, la gestió d'usuaris i la programació d'activitats i materials.

Materials d'aprenentatge: Pots carregar i compartir materials d'aprenentatge, com documents, presentacions, vídeos i enllaços web, per als estudiants.

Comunicació: Moodle facilita la comunicació entre docents i estudiants mitjançant fòrums de discussió, xats en línia, missatges i notificacions.

Avaluació i avaluació: Pots crear tasques i qüestionaris per als estudiants, així com avaluar i donar retroacció sobre els seus treballs.

Calendari i seguiment: Moodle ofereix un calendari integrat que ajuda a programar activitats i tasques i a seguir el progrés dels estudiants.

Com utilitzar Moodle:

Per utilitzar Moodle, pots seguir aquests passos:

Accedeix a un lloc web de Moodle o instal·la la plataforma en el teu propi servidor si tens les habilitats tècniques necessàries.

Crea un compte com a administrador o professor de Moodle. Un cop dins, podràs crear cursos i gestionar continguts.

Crea un curs i personalitza la seva configuració, com el nom, la descripció i les dates límit.

Afegeix contingut al curs, com ara materials d'aprenentatge, tasques, qüestionaris i recursos.

Convida estudiants afegint-los al curs o proporcionant-los l'enllaç d'accés.

Els estudiants podran accedir al curs, accedir als materials d'aprenentatge, realitzar tasques i participar en les activitats del curs.

Els professors podran fer seguiment del progrés dels estudiants, avaluar les seves tasques i comunicar-se amb ells a través de les eines de comunicació de Moodle.

Utilitza el calendari de Moodle per a programar activitats i tasques i mantenir-te organitzat.

Moodle és una eina versàtil i potent per a la gestió de l'aprenentatge en línia i pot ser utilitzada en una àmplia varietat d'entorns educatius i formatius. La seva naturalesa de codi obert i la seva flexibilitat fan que siga una opció popular per a molts docents i institucions.

6. Instal·lar Moodle

<https://www.youtube.com/channel/UCEONRyyFdRqIEJeu7m6gXBw>

7. Aules

La plataforma "Aules de GVA" és una eina educativa en línia que es fa servir a la Comunitat Valenciana, i està dissenyada per a facilitar la gestió i el suport a l'aprenentatge en entorns educatius.

Aules és la plataforma d'aprenentatge en línia de la Generalitat Valenciana, dissenyada per a oferir serveis d'educació virtual als docents i als estudiants de la Comunitat

Valenciana. Aquesta plataforma permet als usuaris, ja siguen professors o estudiants, accedir a cursos virtuals, recursos didàctics, tasques i altres continguts relacionats amb l'educació.

Com utilitzar "Aules de GVA":

Per utilitzar "Aules de GVA", normalment cal seguir aquests passos:

Accés a la plataforma: Docents i estudiants han de tenir accés a la plataforma mitjançant les seves credencials proporcionades per l'administració educativa de la Comunitat Valenciana.

Exploració de cursos: Una vegada dins de la plataforma, els usuaris poden explorar els cursos disponibles i inscriure's als que crega.

Accés als continguts: Una vegada inscrits en un curs, els estudiants poden accedir als continguts del curs, que poden incloure materials d'aprenentatge, tasques, qüestionaris i altres recursos.

Interacció amb els docents i companys: "Aules de GVA" ofereix eines de comunicació i col·laboració perquè els estudiants puguen interactuar amb els docents i amb els altres estudiants. Això pot incloure fòrums de discussió, xats i correus electrònics.

Realització de tasques i avaluació: Els estudiants poden completar tasques, qüestionaris i altres activitats assignades pels docents. Aquests treballs poden ser avaluats i calificados a través de la plataforma.

Seguiment del progrés: Tant els docents com els estudiants poden fer seguiment del progrés en els cursos, veient les puntuacions i el desenvolupament de les activitats.

Accés a recursos i suport addicional: La plataforma pot oferir recursos addicionals i suport en línia, com ara biblioteques digitals i tutorials.

En resum, "Aules de GVA" és una plataforma de gestió de l'aprenentatge dissenyada per a l'educació a la Comunitat Valenciana. Facilita l'accés als cursos i recursos d'aprenentatge en línia, així com la interacció entre docents i estudiants.



Dropbox



Índex

- [1. Serveis de gestió d'arxius web](#)
 - [2. Característiques comunes als serveis de gestió d'arxius web](#)
 - [3. Instal·lació de NextClod i de ownCloud](#)
-

1. Serveis de gestió d'arxius web

Els serveis de gestió d'arxius web són eines en línia que permeten als usuaris gestionar els seus arxius i documents a través d'Internet. Aquests serveis solen oferir funcions com l'emmagatzematge d'arxius, la compartició de documents, la sincronització d'arxius i la col·laboració en temps real. Alguns exemples de serveis de gestió d'arxius web populars inclouen Google Drive, Dropbox i OneDrive. Aquests serveis permeten als usuaris emmagatzemar i accedir als seus arxius des de qualsevol lloc amb connexió a Internet i col·laborar amb altres persones en projectes i documents compartits.

2. Característiques comunes als serveis de gestió d'arxius web

Els serveis de gestió d'arxius web solen compartir diverses característiques comunes que faciliten als usuaris la gestió dels seus arxius i documents a través d'Internet. Algunes d'aquestes característiques inclouen:

Emmagatzematge en línia: Aquests serveis permeten als usuaris carregar i emmagatzemar els seus arxius a servidors en línia, cosa que els permet accedir-hi des de qualsevol dispositiu amb connexió a Internet.

Compartició de documents: És possible compartir arxius i carpetes amb altres persones mitjançant enllaços o convidar-los directament a accedir als arxius. Això facilita la col·laboració en projectes i la distribució de documents.

Control d'accés: Els serveis de gestió d'arxius web permeten als usuaris controlar qui pot veure, editar o descarregar els seus arxius. Això implica l'establiment de permisos i restriccions d'accés.

Sincronització de dispositius: Molts d'aquests serveis ofereixen aplicacions per a diferents dispositius, com ara telèfons mòbils, tauletes i ordinadors. La sincronització entre dispositius permet que els canvis es reflectisquen en temps real en tots els dispositius connectats.

Recuperació de versions anteriors: Molts serveis de gestió d'arxius permeten recuperar versions anteriors dels arxius, el que resulta útil en cas d'errors o canvis no desitjats.

Cerca i organització d'arxius: Proporcionen eines per organitzar arxius en carpetes, cercar ràpidament documents i aplicar etiquetes o metadades per a una millor gestió.

Col·laboració en temps real: Molts serveis permeten la col·laboració en temps real en documents mitjançant l'edició simultània per part de múltiples usuaris.

Seguretat i xifrat: La majoria d'aquests serveis ofereixen seguretat mitjançant xifrat i autenticació de dos factors per protegir els arxius emmagatzemats.

Compatibilitat amb diferents formats de fitxers: Solen ser compatibles amb una àmplia gamma de formats de fitxers, com ara documents de text, imatges, vídeos i altres tipus de documents.

Integració amb altres aplicacions: Molts serveis ofereixen integració amb altres aplicacions i serveis, com ara clients de correu electrònic, sistemes de gestió de projectes i aplicacions d'ofimàtica.

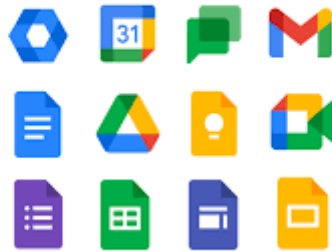
Aquestes característiques fan que els serveis de gestió d'arxius web siguin eines valuoses per a la gestió i col·laboració en documents i arxius, tant per a usuaris individuals com per a equips i empreses.

3. Instal·lació de NextClod i de ownCloud



<https://www.youtube.com/channel/UCEONRyyFdRqIEJeu7m6gXBw>

Google Workspace



Índex

[1. Introducció](#)

[2. Exemples](#)

[3. El més famós és: Google Workspace](#)

1. Introducció

Les "suites d'oficina en el núvol" o "productivitat en el núvol" són conjunts d'aplicacions i serveis de programari relacionats amb la productivitat que són accessibles a través d'Internet i estan allotjats en servidors remots. Aquestes suites ofereixen eines com processadors de text, fulls de càlcul, programari de presentació, correu electrònic, emmagatzematge en línia i serveis de col·laboració. Algunes opcions conegudes inclouen Microsoft 365 (anteriorment Office 365), Google Workspace (abans G Suite) i altres.

Les suites d'oficina en el núvol permeten als usuaris accedir i treballar en els seus documents des de qualsevol lloc amb connexió a Internet. També faciliten la col·laboració en temps real, ja que diverses persones poden treballar en els mateixos documents simultàniament i veure els canvis en temps real. Aquest model d'accés en línia ofereix flexibilitat i facilita la col·laboració en entorns de treball moderns.

2. Exemples

Google Workspace (anteriorment G Suite):

- Google Docs: Per crear i editar documents de text en línia.
- Google Sheets: Per fulls de càlcul i anàlisi de dades.
- Google Slides: Per a la creació de presentacions.
- Gmail: Per al correu electrònic empresarial.
- Google Calendar: Per programar esdeveniments i reunions.

Microsoft Office Online:

- Word Online: Per crear i editar documents de Word al navegador.
- Excel Online: Per fulls de càlcul en línia.
- PowerPoint Online: Per crear i presentar diapositives en línia.
- Outlook.com: Per al correu electrònic i el calendari.

Zoho Office Suite:

- Zoho Writer: Processador de text en línia.
- Zoho Sheet: Per fulls de càlcul.
- Zoho Show: Per a la creació de presentacions.
- Zoho Mail: Per al correu electrònic.

LibreOffice Online:

Proporciona una versió en línia de la coneguda suite d'oficina de codi obert.

OnlyOffice:

Ofereix processador de text, fulls de càlcul i presentacions en línia.

Dropbox Paper:

Per a la col·laboració en temps real en la creació de documents, incloent text i mitjans.

Notion:

Una aplicació de productivitat tot en un que permet crear documents, bases de dades i llistes de tasques.

Airtable:

Combina les característiques de fulls de càlcul i bases de dades per gestionar i col·laborar en dades de manera eficaç.

Trello:

Tot i que no és una suite d'oficina completa, és una eina de gestió de projectes en línia que s'utilitza per organitzar tasques i projectes.

Evernote:

Ideal per prendre notes, organitzar idees i emmagatzemar contingut a la núvol.

Aquestes aplicacions de productivitat web són accessibles des de qualsevol dispositiu amb connexió a Internet i permeten la col·laboració en temps real, el que les fa valuoses per a treballadors remots, equips distribuïts i persones que necessiten accedir als seus documents en qualsevol lloc.

A més, moltes d'elles ofereixen versions gratuïtes amb funcionalitats bàsiques, així com plans de pagament amb característiques addicionals i emmagatzematge ampliat.

3. El més famós és: Google Workspace

Google Workspace és una suite de productivitat a la núvol de Google que ofereix una varietat d'eines i serveis dissenyats per ajudar les empreses a col·laborar, comunicar-se i gestionar la seua feina de manera eficient. Està dirigit tant a xicotetes empreses com a grans corporacions i ofereix diverses aplicacions i característiques que faciliten el treball en equip i la gestió de tasques. Aquí tens alguns detalls sobre el seu funcionament i utilitat:

1. **Correu Electrònic:** Google Workspace proporciona una potent plataforma de correu electrònic a través de Gmail, amb funcions de cerca avançada, etiquetes i capacitat per personalitzar el domini de correu de la teva empresa.
2. **Calendari Compartit:** Google Calendar permet als usuaris programar reunions, compartir esdeveniments i sincronitzar calendari entre els membres de l'equip. També pots rebre recordatoris i notificacions.
3. **Emmagatzematge a la Núvol:** Google Drive ofereix un espai d'emmagatzematge a la núvol per als teus fitxers, permetent l'accés des de qualsevol dispositiu. Pots col·laborar en documents, fulls de càlcul i presentacions en temps real.
4. **Col·laboració en Documents:** Google Docs, Sheets i Slides són eines de processament de text, fulls de càlcul i presentacions, respectivament. Permet l'edició col·laborativa en temps real i el seguiment de canvis.

5. **Missatgeria Instantània i Videotrucades:** Google Chat i Google Meet permeten la comunicació en temps real a través de missatges instantanis i videoconferències, facilitant la col·laboració a distància.
6. **Formularis i Enquestes:** Google Forms permet crear qüestionaris i enquestes personalitzades per recollir dades i retroalimentació de manera eficient.
7. **Administració d'Usuaris i Seguretat:** Google Workspace proporciona eines d'administració d'usuaris i seguretat avançada per protegir les dades i les comptes de la teua empresa.
8. **Integració amb altres eines:** Google Workspace s'integra fàcilment amb altres aplicacions i serveis de Google, com Google Analytics, Google Ads i Google Cloud Platform, així com amb aplicacions de tercers a través de Google Workspace Marketplace.
9. **Accés des de Qualsevol Dispositiu:** Google Workspace és accessible des de pràcticament qualsevol dispositiu amb connexió a Internet, facilitant la feina en moviment i la col·laboració remota.
10. **Escalabilitat:** Pots ajustar la quantitat de llicències de Google Workspace segons les necessitats de la teua empresa, el que el fa escalable per créixer amb la teua organització.

Google Workspace és una eina potent per a empreses que volen millorar la col·laboració, augmentar la productivitat i simplificar la gestió de la informació. Permet als equips treballar junts de manera més eficient, emmagatzemar i compartir fitxers de forma segura i mantenir una comunicació fluida, el que el converteix en una solució versàtil per a empreses de tots els mides.
