



UD03: Introducció a HTML, CSS i PHP

Aplicacions Web (AW)

```
<html>  
<head>  
  <meta name="TITLE" content="...">  
  <meta name="KEYWORDS" content="...">  
  <meta name="DESCRIPTION" content="...">  
  <link rel="stylesheet" type="text/css" href="...">  
  <script language="javascript" src="...">  
  </script>  
</head>  
<body>  
  <div id="container" style="border: 1px solid black; padding: 10px;">  
    <h1>Hello World!</h1>  
  </div>  
</body>  
</html>
```

Qué anem a vore?

- **Conceptes previs**
 - Estàndards, ACID, Browser Wars
 - WYSIWYG, VsCode
 - W3C, Firebug?, W3School
- **HTML**
- **CSS**
- **PHP**

Conceptes Prèvis



Estàndards i accessibilitat web

El W3C és un consorci internacional on diferents organitzacions treballen, conjuntament amb els usuaris que hi volen participar i col·laborar, en el desenvolupament dels estàndards web. La missió del W3C és la d'encaminar la web cap al seu màxim potencial, desenvolupant protocols i pautes que assegurin el seu creixement futur.





Hi havia diverses regles bàsiques de disseny que guiaven el HTML, i algunes opcions pragmàtiques, fins i tot polítiques. Una regla filosòfica era que el HTML havia de contenir l'estructura d'un document d'hipertext, però no detalls de la seva presentació. Aquesta era l'única forma perquè aparegués raonablement en qualsevol d'una àmplia gamma de pantalles diferents i diverses mides de paper.

Tim Berners-Lee, Weaving the WEB, 1999

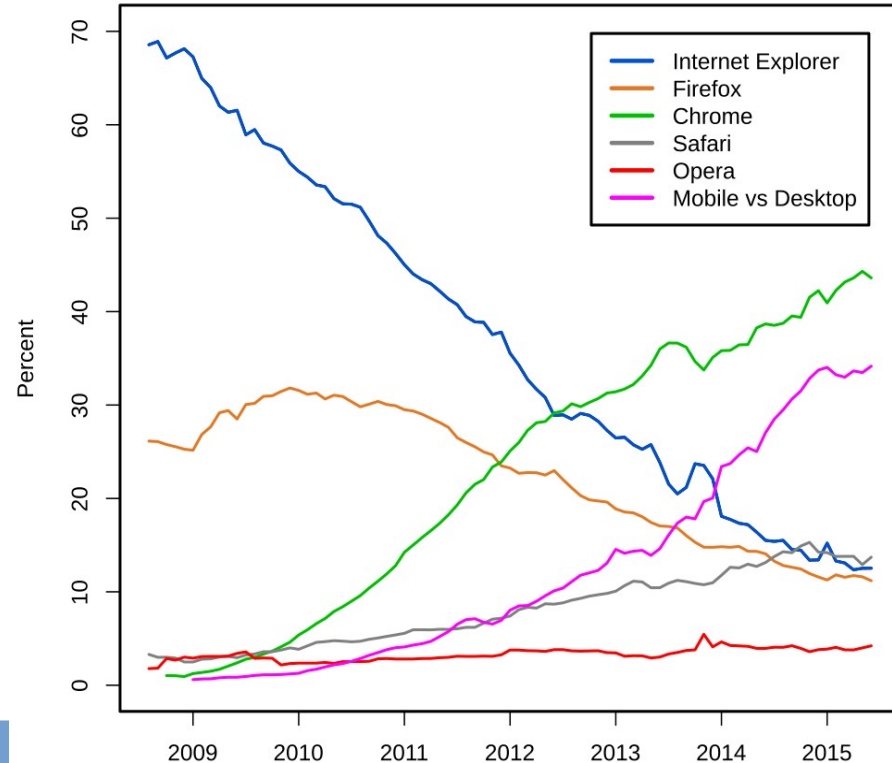


Beneficis dels usos d'estandars web

- Compatibilitat
- Independència del dispositiu
- Control centralitzat
- Facilita la feina als cercadors
- Pàgines més lleugeres
- Accessibilitat



Suport dels navegadors als estàndards

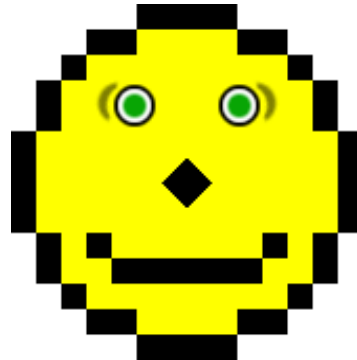


Font: Wikipedia



Acid test

L'Acid2 és una prova, que consisteix en una imatge (veure il·lustració) basada en la icona "àcida" de l'Acid Music, creada pel Web Standards Project, per mostrar les mancances dels navegadors web actuals a l'hora de complir amb els estàndards.





Acid test fallats:



Explorer 7.0

Hello World!



Firefox 2.0



Editors WYSIWYG

WYSIWYG, en anglès allò que veus és el que obtens, és un terme que s'aplica als processadors de text i altres editors de text amb format, com els editors de HTML, que permeten escriure un document veient el resultat final. En el cas dels editors d'HTML, aquest concepte s'aplica a aquells programes que permeten escriure la pàgina sobre una vista preliminar semblant a la d'un processador de textos, ocupant-se el programa, en aquest cas, de generar el codi font en HTML.

Exemples d'editors d'HTML del tipus WYSIWYG són:

- Adobe Dreamweaver
- Adobe GoLive
- NVU/Kompozer
- FrontPage
- Amaya



VSCode

Ajudes a l'HTML:

<https://code.visualstudio.com/docs/languages/html>

Complement Live Server:

<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>



Web de validació de la W3C

El W3C ofereix un servei gratuït de validació de documents HTML, XHTML i CSS. Aquest servei conegut com a Markup Validation Service inclou tres variants:

- Per mitjà d'una URI vàlida
- Pujant un arxiu
- Entrada directa

The screenshot shows the W3C Markup Validation Service interface. At the top, there is a blue header with the W3C logo and the text 'Markup Validation Service' and 'Check the markup (HTML, XHTML) of Web documents'. Below the header, there are three tabs: 'Validate by URI', 'Validate by File Upload', and 'Validate by Direct Input'. The 'Validate by URI' tab is selected. Under this tab, there is a section titled 'Validate by URI' with the text 'Validate a document online:'. Below this, there is a label 'Address:' followed by a text input field. At the bottom right of the form, there is a 'Check' button. A link labeled 'More Options' is also visible below the input field.



Web de validació de la W3C

| This Page Is Valid XHTML 1.0 Strict! | |
|--------------------------------------|---|
| Result: | Passed validation |
| File: | practica_4.html |
| Encoding: | utf-8 |
| Doctype: | XHTML 1.0 Strict |
| Root Element: | html |
| Root Namespace: | http://www.w3.org/1999/xhtml |

S'ha validat correctament

| This page is not Valid XHTML 1.0 Strict! | |
|---|---|
| Result: | Failed validation, 8 Errors |
| File: | practica_4.html |
| Encoding: | utf-8 |
| Doctype: | XHTML 1.0 Strict |
| Root Element: | html |
| Root Namespace: | http://www.w3.org/1999/xhtml |

Ha fallat la validació



Icones de validació de la W3C

"valid" Icon(s) on your Web page

To show your readers that you have taken the care to create an interoperable use to add this icon to your Web page:



```
<p>  
<a href="http://validator.w3.org/check?uri=referer"></a>  
</p>
```



```
<p>  
<a href="http://validator.w3.org/check?uri=referer"></a>  
</p>
```



Web de validació de la W3C (També per a CSS)

El W3C també disposa d'una utilitat web gratuïta que permet validar la sintaxi dels fulls d'estil (CSS) associades als documents HTML/XHTML. Inclou igualment tres variants:

- Mitjançant una URI vàlida
- Pujant un arxiu
- Entrada directa

The screenshot shows the W3C CSS Validation Service interface. At the top, there's a blue header with the W3C logo and the text 'CSS Validation Service' and 'Verifica Hojas de Estilo en Cascada (CSS) y documentos (X)HTML con hojas de estilo'. Below the header, there are three tabs: 'mediante URI', 'mediante Carga de Archivo', and 'mediante Entrada directa'. The 'mediante Entrada directa' tab is selected. The main area is titled 'Validar mediante entrada directa' and contains the instruction 'Introduce el código CSS que desees validar:'. Below this is a large text input field. At the bottom, there is a 'Check' button and a link that says '► Más opciones'.



Abans FireBug, ara?

El Firebug és un potent complement per al navegador Mozilla Firefox que permet avaluar i supervisar el CSS, l'HTML i les accions per part de codi Javascript a temps real. [Web](#).





Eines d'anàlisi d'accessibilitat web

TAW: és una eina disponible com aplicació web, independent del sistema (standalone) o com a extensió del navegador Mozilla Firefox.

Web: <https://www.tawdis.net/>



W3Schools

És una web de referència al món del desenvolupament web, amb tutorials i documentació detallada i amb exemples.

Web: <https://www.w3schools.com/>

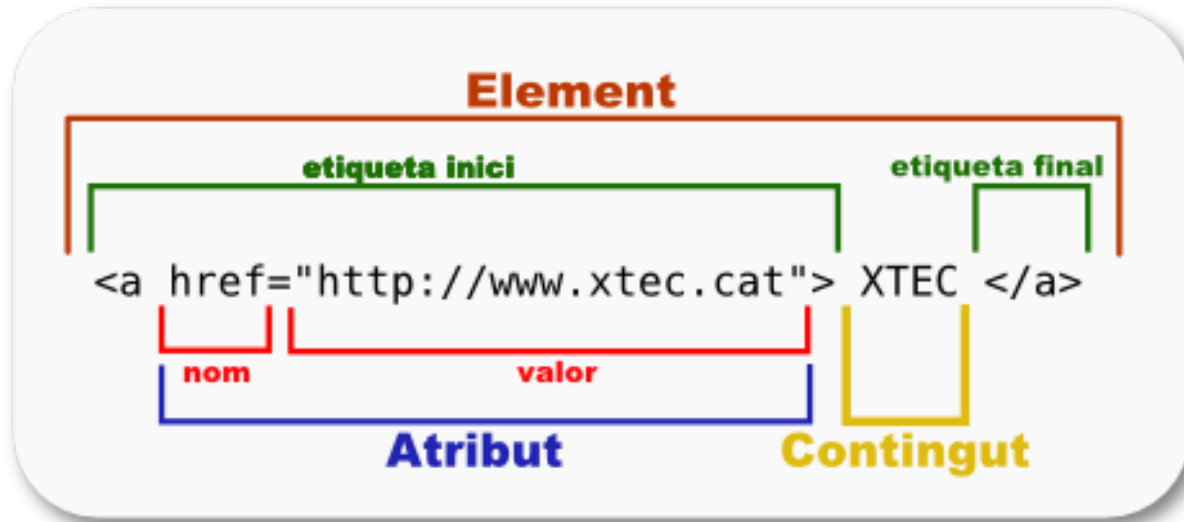
HTML

Què és XHTML?

- XHTML respon a l'acrònim eXtensible HTML, és a dir, a una revisió o millora de les especificacions del HTML, que és el llenguatge de marcat que transforma els documents de text en pàgines web. Aquesta nova orientació del llenguatge HTML ha sorgit de la necessitat de permetre una correcta interpretació de la informació independentment del dispositiu des del que s'accedeix a ella, segons paraules de la W3C
- El pas del HTML a XHTML suposa l'acceptació d'uns requisits bàsics en quant a l'estructuració i definició del document web que assegurin que aquest document, al respectar els estàndards, resulta llegible per qualsevol dispositiu present o futur.

Elements, etiquetes i atributs

- Quan treballem amb llenguatge XHTML ho farem amb elements, etiquetes i atributs:



Etiquetes i atributs d'especial interès

- Al marge de les etiquetes i atributs que determinen el valor semàntic de cada element del document XHTML, que s'aniran veient al llarg del curs, hi ha dos etiquetes i dos atributs que requereixen una atenció especial, són les etiquetes `<div>` i ``, i els atributs `id` i `class`

Etiquetes div i span

- Les etiquetes **div** i **span** no tenen un valor semàntic específic. La seva funció és purament estructural: agrupar parts del codi XHTML per facilitar la seva identificació i posterior associació a una regla d'estil.
- Segons les especificacions del W3C: *Els elements div i span, en combinació amb els atributs id i class, ofereixen un mecanisme per afegir estructura als documents. Aquests elements defineixen un determinat contingut per ser presentats en línia (span) o com a bloc (div).*
- L'etiqueta <div> indica una divisió i té valor de bloc: agrupa una part de l'estructura o contingut del document.

Etiquetes div i span

- L'etiqueta `` agrupa elements en línia com paraules o frases curtes
- Les dues etiquetes estan mancades de valor semàntic i la seva finalitat és **purament estructural**. Aquesta funcionalitat s'oblida sovint quan es fa servir l'etiqueta ``, en combinació amb els atributs `id` i `class`, per a la creació de nous elements amb significat. Caldria evitar aquesta pràctica en benefici de les etiquetes del propi llenguatge XHTML i no atribuir a l'etiqueta `` un valor d'èmfasi, per exemple, quan es disposa de l'etiqueta ``, que té aquesta finalitat.

Incorrecte

```
01. | <h1>Hem guanyat la <span class="important">Copa d'Europa</span></h1>
```

Correcte

```
01. | <h1>Hem guanyat la <em>Copa d'Europa</em></h1>
```


Atributs id i class

- S'ha vist que els elements `<div>` i `` no tenen cap valor semàntic. Són invisibles en aquest sentit. Els atributs `id` i `class` supleixen aquesta manca de significat.
- L'atribut `class` assigna un valor semàntic a un element definit per l'usuari/a.
- El valor semàntic del paràgraf amplia el seu significat quan se li afegeix l'atribut `class` amb el valor `atencio`.
- Es permet assignar el mateix valor de `class` a diferents elements del document de la mateixa manera que un element pot compartir diferents noms de classe.

```
01. <p class="atencio alumnes">Aquesta nota d'atenció és per als alumnes</p>  
02. <p class="atencio professors">Aquesta nota d'atenció és per als professors</p>  
03. <p class="atencio alumnes nouvinguts">Aquesta nota d'atenció és per als alumnes nouvinguts al centre</p>
```

Atributs id i class

- L'atribut id, a diferència de class, té un valor únic. Repetir el mateix valor de l'atribut id en diferents elements del document invalidarà el document. Un exemple habitual del seu ús és la identificació única de diferents divisions del document.

```
01. <div id="capçalera">  
02.   ...  
03. </div>  
04. <div id="navegacio">  
05.   ...  
06. </div>  
07. <div id="contingut">  
08.   ...  
09. </div>  
10. <div id="peu">  
11.   ...  
12. </div>
```

Estructura bàsica d'un document XHTML

- Un document XHTML , per ser vàlid, ha de tenir una estructura mínima que ha de contenir una sèrie de parts que es tractaran en aquest mòdul.
- A l'exemple següent podeu veure la que seria l'estructura mínima d'un document XHTML vàlid:

```
Veure en text pla  Còpia  Imprimir  ?
01.  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
02.  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
03.  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">
04.  <head>
05.      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
06.      <title>Títol del document</title>
07.  </head>
08.  <body>
09.      <!-- inici del contingut del document -->
10.
11.      <!-- final del contingut del document -->
12.  </body>
13. </html>
```

Parts d'un document XHTML - Declaració del tipus de document

- A la primera línia d'un document XHTML ha d'aparèixer sempre la declaració del tipus de document, dins de l'etiqueta `<!DOCTYPE>`:
- És un requisit obligatori. Sense aquesta declaració un document XHTML no és vàlid, ja que el navegador o dispositiu desconixerà quines són les regles que l'han d'ajudar a interpretar-lo correctament.

```
01. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
02. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Parts d'un document XHTML - Element arrel i idioma

- Un document XHTML ha de tenir sempre un element arrel. Aquest element arrel s'obre amb l'etiqueta <html> i es tanca amb </html> posant-la sempre al final del tot del document. Fixeu-vos en la plantilla de l'inici: immediatament després de la declaració del tipus de document s'obre l'element arrel amb la etiqueta <html>, que ha de contenir els atributs que fan referència a l'espai de noms i a l'idioma.

```
03. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">
```

Parts d'un document XHTML - Capçalera del document

- La capçalera d'un document XHTML s'obre amb l'etiqueta `<head>` i es tanca amb `</head>`. Entre aquestes dues etiquetes se situa tota una sèrie d'informació relativa al document: tipus de contingut, títol, metadades, enllaços, referències,...

```
04. <head>
05.     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
06.     <title>Títol del document</title>
07. </head>
```

Parts d'un document XHTML - Cos del document

- El cos inclou el contingut visible del document XHTML al navegador. S'inicia amb l'etiqueta `<body>` i es tanca amb l'etiqueta `</body>`.
- Qualsevol element, caràcter, ... del contingut que no es trobi entre les dues etiquetes `<body>...</body>`, invalidarà el document.

```
08. <body>
09.     <!-- inici del contingut del document -->
10.
11.     <!-- final del contingut del document -->
12. </body>
13. </html>
```

Regles per a la validació correcta de la sintaxi del llenguatge XHTML

Hi ha una sèrie de regles que un document XHTML ha de complir sempre per ser un document vàlid. És molt important tenir-les presents. Són les següents:

- 1) Un document XHTML **requereix una declaració del tipus de document** per a ser correctament interpretat pels navegadors, i, per tant, susceptible de ser validat. La declaració del DTD ha de ser sempre la primera línia del document XHTML i el seu contingut ha de ser el següent: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
- 2) Un document XHTML ha de tenir **totes les etiquetes, atributs i valors en minúscules**. Per exemple s'haurà de posar `<body>` enlloc de `<BODY>`, o `<h2>` enlloc de `<H2>`, i per tant, per exemple, `<table>`, `<Table>` i `<TABLE>` seran interpretats com a elements diferents.
- 3) Un document XHTML, a diferència de l'HTML, **requereix la presència de l'etiqueta `<html>`**, i, a més, aquesta ha d'incloure els atributs `xmlns`, `xml:lang` i `lang`: `<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">`
- 4) **Tots els elements de marcat d'un document XHTML s'han de tancar amb l'etiqueta corresponent** i tots els atributs han de quedar entre cometes. Els elements buits com el salt de línia `
` han de complir també aquesta norma. Així, l'etiqueta `
`, vàlida en HTML, serà vàlida en XHTML afegint-li una barra inclinada abans del signe de tancament: `
`. Altres elements buits que resulten afectats per aquesta norma són: `<col />`, ``, `<input />`, `<link />` i `<meta />`.
- 5) **Els elements del document XHTML han d'estar anidats correctament**, és a dir les etiquetes que obren i tanquen els diferents elements han d'estar en l'ordre correcte. Incorrect: `<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit </p> `.

D'altra banda, la declaració del tipus de document XHTML especifica un conjunt de **regles d'anidació dels elements**:

- Un element **a** no pot contenir altres elements **a**.
- L'element **pre** no pot contenir els següents elements: **img**, **object**, **big**, **small**, **sub**, **sup**, **font**, **applet** o **basefont**.
- Un element **button** no pot contenir els següents elements: **input**, **select**, **textarea**, **label**, **button**, **form**, **fieldset**, **iframe** o **isindex**.
- Un element **label** no pot contenir altres elements **label**.
- Un element **form** no pot contenir altres elements **form**.

CSS

Què és CSS?

- CSS respon a l'acrònim Cascade Style Sheets, és a dir, fulls d'estil en cascada. CSS és un llenguatge orientat a definir l'estil dels documents web.
- Hi ha diferents motius per utilitzar CSS en la creació dels documents web:
 - Respectar el lema del W3C de separar l'estructura i la presentació del document, el que permet una excel·lent millora de l'accessibilitat dels documents web.
 - Frenar l'ús erroni d'elements HTML amb finalitats visuals i no estructurals, que degrada l'accessibilitat del document web, incrementa el pes de l'arxiu i complica el seu manteniment.
 - Tenir un control centralitzat del disseny, la qual cosa comporta una gran millora en el seu desenvolupament, edició i manteniment a llarg termini.

Què és CSS?

- La presentació no es limita només a la pantalla de l'ordinador. Mitjançant els CSS es pot definir la impressió del document, la presentació visual en dispositius mòbils o la pronúncia del document a través de dispositius de lectura.
- De la mateixa manera que l'HTML i l'XHTML, el llenguatge CSS ha anat evolucionant a través de diferents versions segons les directrius del W3C: la CSS 1, la CSS 2, la CSS 2.1 i la darrera, encara en desenvolupament, la CSS 3. Aquest curs seguirà les especificacions de la versió CSS 2.1, que és una revisió, amb correcció d'errors i millores de la versió 2.

Modes CSS

- **Estil Extern:**
 - Arxiu CSS extern – **RECOMANAT!**
- **Estil Local:**
 - Propietats com a atributs en l'etiqueta HTML – **OK per a PROVES!**

```
<html>
<body>
<h1 style="color:green; font-size:x-large;">Donant format amb CSS </h1>
<p style="font-weight:bold; font-size:medium;">Esta pàgina és <em>molt</em> simple</p>
</body>
</html>
```

- **Estils Interns:**
 - Definir propietats a la capçalera del document HTML – **No reutilitzable.**

```
<html>
<head>
  <style type="text/css">
    h1 {
      color: green;
      font-size: x-large;
    }
    #peu {
      color: #FFFF00;
      font-size: small;
    }
  </style>
</head>
```



```
<body>
  <h1> Donant format amb CSS! </h1>
  <p id="peu">Futur Informatic</p>
</body>
</html>
```

Sintaxi del CSS

- La sintaxi del CSS difereix completament de la sintaxi de l'XHTML. Enlloc d'un nombre d'etiquetes que estructurin la informació, en CSS es troba un conjunt de regles formades per selectors, propietats i valors. Aquestes regles determinen l'estil del document. El model que defineix una regla CSS seria:

```
01. SELECTORS {  
02.     PROPIETAT : VALOR;  
03. }
```

- El selector correspon al nom de l'element al que modifica la regla. La propietat indica la característica de l'element que quedarà determinada amb la regla. El valor sempre acompanya a la propietat, després dels dos punts (:). El conjunt format per la propietat i el seu valor es coneix com a expressió. Les expressions han d'anar tancades entre claus ({...}), i han d'acabar sempre amb un punt i coma (;). El conjunt format per selectors i expressions es coneix com a regla.

Sintaxi del CSS

- Fixeu-vos en l'exemple següent:
- El selector correspondria a h2, és a dir, aquesta regla modificarà l'estil de tots els elements h2 del document XHTML al que està associat la regla d'estil.
- font-size representa una propietat que, acompanyada d'un valor, 12px, determinarà, en aquest cas, la mida de la font.

```
01. h2 {  
02.   font-size: 12px;  
03. }
```

Sintaxi del CSS

- Una regla pot contenir diferents expressions:

```
01. h2 {  
02.     font-size: 12px;  
03.     font-family: monospace;  
04.     color: black;  
05.     background-color: white;  
06. }
```

- Diferents selectors poden compartir una mateixa regla:

```
01. h2, h3, h4 {  
02.     font-size: 12px;  
03.     font-family: monospace;  
04.     color: black;  
05.     background-color: white;  
06. }
```


Validació del CSS

- **De la mateixa manera que passava amb la validació d'un document XHTML, és important que un document CSS compleixi unes normes per ser vàlid. Aquestes normes són:**
 - Ha d'estar codificat sota UTF-8.
 - Tots els selectors han d'estar escrits en minúscules (case sensitive). Les propietats i els seus valors poden aparèixer tant en minúscules com en majúscules, però es recomana l'ús de les minúscules com a estil de redacció de l'arxiu.
 - El conjunt d'expressions (propietats i valors) d'una regla sempre han d'estar tancats per claus d'entrada i sortida ({...}). L'absència d'una sola d'aquestes claus provocaria inconsistències en el resultat i, per descomptat, l'arxiu no seria vàlid.
 - Cada expressió d'una regla ha d'acabar obligatòriament amb un punt i coma (;). L'absència d'un sol punt i coma provocaria la no validació de l'arxiu.
 - Els comentaris en un arxiu CSS comencen amb l'expressió /* y acaben amb con */

Tipus de selectors. Selectors XHTML

El selectors de tipus XHTML identifiquen un determinat element de l'XHTML. Alguns exemples de selectors XHTML són:

```
01. p {  
02.   color: red;  
03. }
```

El selector p s'associa a l'etiqueta <p>, és a dir, a tots els paràgrafs del document XHTML. Segons aquesta regla, tots els paràgrafs del document apareixeran de color vermell (color: red;). Un altre exemple:

```
01. h1 {  
02.   color: green;  
03. }
```

El selector h1 s'associa a l'element <h1> (capçalera de primer nivell) del document XHTML. Segons aquesta regla, totes les capçaleres de primer nivell del document apareixeran de color verd (color: green;).

Selectors id

Especifiquen un determinat element del document XHTML que conté el valor de l'atribut id. Veieu el següent exemple:

```
01. <h1 id="principal">Introducció a Lorem Ipsum</p>
```

L'etiqueta `<h1>` conté l'atribut id de valor "vermell". L'especificació del selector es correspondria amb:

```
01. #principal {  
02.     color: red;  
03. }
```

És a dir, es pren el valor de l'atribut id i se li afegeix al començament el caràcter #.

Selectors class

Especifiquen un determinat element del documento XHTML que conté un atribut del tipus class. Veieu el següent exemple:

```
01. <p class="ciutats">Barcelona</p>
```

Si es vol aplicar un determinat estil a la classe "ciutats", s'utilitzarà el valor de l'atribut class afegint-li al davant el caràcter «.» (punt)

```
01. .ciutats {  
02.     color: red;  
03. }
```

Selectors class

Es poden combinar diferents classes en un mateix element XHTML:

```
01. <p class="ciutats europa">Barcelona</p>
```

i aplicar a cadascuna una regla diferent:

```
01. .ciutats {  
02.     color: red;  
03. }  
04.  
05. .europa {  
06.     font-size: 13px;  
07. }
```

Selector universal

El selector universal engloba a tots els elements del document XHTML. Es representa amb el caràcter d'asterisc: *.

```
01. * {  
02.     font-size: 13px;  
03.     font-family: verdana;  
04. }
```

En aquest exemple, tots els elements del document tindran una mida de la font de 13 píxels i la font de lletra correspondrà a la família Verdana.

Pseudo-classes i pseudo-elements

Les pseudo-classes i els pseudo-elements tenen como a objectiu precisar un fragment (pseudo-classe) o establir una condició (pseudo-element) de l'element associat a la regla. Les pseudo-classes i pseudo-elements es declaren després del selector principal, separats pel símbol de puntuació: `::`. Veieu un primer exemple de l'ús de pseudo-classes:

```
01. p {  
02.     font-size: 10px;  
03. }  
04.  
05. p:first-letter {  
06.     font-size: 20px;  
07. }
```

Pseudo-classes i pseudo-elements

S'han especificat dues regles per a l'element `<p>` del document. La primera determina que la mida de la font dels paràgrafs ha de ser de 10 píxels. Contràriament, a la segona regla, el selector ve acompanyat d'un pseudo-element, `first-letter`, que selecciona i modifica la mida d'una part de l'element, en aquest cas, la primera lletra de cada paràgraf.

Ara veieu un exemple de l'ús dels pseudo-elements:

```
01. a:hover {  
02.     text-decoration: none;  
03. }
```

`hover` és un pseudo-element, és a dir, estableix una condició de verdader/fals al selector: si el cursor passa per sobre de l'enllaç (element `<a>` del document), la línia de base de l'enllaç desapareixerà (`text-decoration:none`).

Anidar selectors

És possible anidar diferents selectors amb el propòsit d'associar una regla CSS a un element específic del document XHTML. Cada selector està separat per un espai. Veieu un exemple:

```
01. #informacio p strong {  
02.     color: red  
03. }
```

Els selectors anidats són #informacio, p i strong. Per ordre de jerarquia, el primer element anadat conté al següent, i així successivament fins l'últim selector anadat.

Anidar selectors

Què significa aquesta regla de selectors anidats? Tots els continguts que es troben entre les etiquetes `...` que estiguin continguts en un paràgraf que, alhora, estiguin dins d'un element `<div id="informacio">`, apareixeran de color vermell. Un exemple de codi XHTML per il·lustrar l'anterior:

```
01. <div id="informacio">
02.   <p>Lorem ipsum <strong>és una expressió</strong></p>
03. </div>
```

Unitats de mesura i color

Tipus d'unitats

Es poden identificar tres tipus diferents d'unitats:

- Numèric
- Percentatges
- Longituds

| Unitat | | Sufix | Exemple |
|-------------|------------|-----------------|-------------------------------|
| Numèric | | <i>Cap</i> | <code>width: 100</code> |
| Percentatge | | <code>%</code> | <code>width: 67%</code> |
| Longitud | Em | <code>em</code> | <code>font-size: 1.2em</code> |
| | Píxels | <code>px</code> | <code>font-size: 12px</code> |
| | Punts | <code>pt</code> | <code>font-size: 13pt</code> |
| | Milímetres | <code>mm</code> | <code>width: 356mm</code> |

Unitats absolutes i relatives

Les unitats absolutes són totes aquelles unitats que no varien en funció del dispositiu o context en les que són utilitzades.

Les unitats relatives, en canvi, varien en funció del dispositiu o context en les que són utilitzades. És el cas d'aquells valors representats en percentatges, o el de les unitats em, que s'explicarà a continuació.

Unitats en em

La unitat em, originalment equivalia a l'alçada de la M majúscula. A l'entorn de producció web, la unitat em representa, però, el valor de la mida de font base del navegador (per defecte, 16 píxels). Per exemple: si un paràgraf es representa a 16 píxels, l'equivalent serà d'1 em; per tant 2 em equivaldran a 32 píxels; i 0.5em equival a 8 píxels.

Unitats de color

Hi ha tres tipus d'unitats diferents per a especificar un color:

- **Valor hexadecimal**

```
01. p {  
02.   color: #aaff00;  
03.   background-color: #f00;  
04. }
```

- **Valor RGB**

```
01. p {  
02.   /* valor numèric */  
03.   color: rgb(170, 255, 0);  
04.   /* valor amb percentatges */  
05.   background-color: rgb(100%, 0, 0);  
06. }
```

- **Nom del color**

```
01. p {  
02.   color: lime;  
03.   background-color: red;  
04. }
```

- 17 colors predeterminats: aqua, black, blue, fuchsia, gray, lime, green, maroon, navy, olive, orange, purple, red, silver, teal, white i yellow

Característiques dels fulls d'estils (CSS)

Una de les característiques més importants del CSS és el que es coneix com herència o ordre en cascada. El que significa treballar en cascada es pot entendre fàcilment a partir de l'exemple següent.

Imagineu-vos que voleu crear un document que tingui tota la tipografia de color gris, d'una mida de 12 píxels, sobre un fons clar. Per aconseguir-ho haureu de definir la regla següent:

```
01. * {  
02.   color: #33f;  
03.   background-color: #eee;  
04.   font-size: 12px;  
05. }
```

Aquesta regla de selector universal afectarà a tots els elements del document.

Característiques dels fulls d'estils (CSS)

Però, poseu per cas, que no voleu que la capçalera del primer nivell tingui el mateix color que la resta d'elements, sinó que el que voleu és que sigui de color vermell. Per això caldrà afegir una segona regla:

```
01. h1 {  
02.   color: red;  
03. }
```

El navegador sobreescrirà el valor del color selector h1 sobre el selector universal, però el selector h1 heretarà la resta de propietats de la regla més general, i mostrarà, per tant, la mateixa mida i el mateix color de fons que la resta d'elements.

Es pot dir que les diferents regles competeixen entre elles segons un ordre de jerarquia o prioritats, de manera que, les propietats d'una regla d'un ordre superior, prevaldran sempre sobre les d'un ordre inferior.

Aquest ordre en cascada separa les regles en sis grups diferents segons el tipus de selector utilitzat. El llistat dels sis grups està ordenat de major a menor prioritats. En termes generals, tindran sempre preferència els tipus de selectors de més precisió davant dels selectors més generals.

Característiques dels fulls d'estils (CSS)

Els sis grups en què queden agrupades les regles segons el tipus de selector utilitzat, ordenats de major a menor prioritat són els següents:

1. Propietats que continguin l'expressió !important darrere del seu valor:

```
01. h1 {  
02.   color: red !important;  
03. }
```

2. Estils declarats com a valor de l'atribut style de l'element XHTML:

```
01. <h1 style="color: red;">Lorem Ipsum</p>
```

3. Regles definides per un o més selectors del tipus ID:

```
01. #vermell {  
02.   color: red;  
03. }
```


Característiques dels fulls d'estils (CSS)

4. Regles definides per una o més classes, atributs o pseudo-selectors:

```
01. .vermell {  
02.     color:red;  
03. }  
04.  
05. p:first-letter {  
06.     color: red;  
07. }  
08.
```

5. Regles que contenen un o més selectors XHTML:

```
01. h1 {  
02.     color: red;  
03. }
```

6. Regles que contenen el selector universal:

```
01. * {  
02.     color: red;  
03. }
```

Característiques dels fulls d'estils (CSS)

Què passa quan es troben dues regles del mateix nivell i del mateix grup?

Prioritzarà sempre la regla que contingui més selectors d'una jerarquia superior. Veieu un exemple:

```
01. #contingut .expressio .irregular  
02.     color: red;  
03. }  
04.  
05. #contingut .expressio p strong {  
06.     font-size: 20px;  
07. }
```

Les dues regles comparteixen el mateix selector id, #contingut, però la primera regla tindrà preferència sobre la segona, degut a que conté dos selectors del tipus class mentre que la segona només conté un del mateix tipus.

En el cas hipotètic de que les dues regles continguin el mateix nombre de selectors d'ordre preferent, la prioritat vindrà donada per la seva ubicació. Una regla situada en una ubicació de prioritat alta s'imposarà sobre una altra d'ubicació de menys prioritat.

Característiques dels fulls d'estils (CSS)

Hi ha sis tipus d'ubicacions, que queden ordenades de major a menor segons el seu nivell de prioritat, de la següent manera:

- 1) Tota regla d'estil ubicada dins de l'element `<style>` tindrà preferència sobre la resta d'ubicacions.
- 2) Tota regla d'estil ubicada en un full d'estil extern invocada mitjançant l'expressió `@import` dins de l'element `<style>` del document.
- 3) Tota regla d'estil invocada mitjançant un enllaç del tipus `<link />`.
- 4) Tota regla d'estil invocada mitjançant una expressió `@import` dins d'una etiqueta `<link />` del document.
- 5) Tota regla d'estil enllaçada manualment per l'usuari.
- 6) Tota regla d'estil proporcionada per defecte pel navegador.

Finalment, quan diferents regles d'estil comparteixen el mateix nivell de prioritat d'ubicació, l'ordre de preferència vindrà determinat per l'ordre d'aparició. Una regla que apareix més tard que una altra tindrà preferència sobre la que ha aparegut prèviament.

Aplicar CSS al document XHTML

Hi ha diferents maneres d'aplicar un full d'estil. Cadascuna d'elles té avantatges i inconvenients. L'elecció més adient vindrà donada per l'escenari o situació en que s'està treballant.

CSS en línia

Mitjançant l'atribut style es poden aplicar regles d'estil directament a un element del document XHTML:

```
01. <p style="color: red; font-size: 20px;">Lorem Ipsum</p>
```

Aquest mètode, com s'ha vist a la pràctica anterior, sobreescriurà els valors de la resta de regles de nivell inferior degut al seu alt nivell de preferència en el llistat d'ordre de cascada (exceptuant aquelles propietats que van acompanyades de l'expressió !important). Per tant, amb aquest mètode, queda assegurat que la regla visual s'aplicarà sense reserves a l'element en qüestió.

El gran inconvenient d'aquesta pràctica és anar contra la pròpia filosofia dels CSS, que és la de separar la presentació del contingut. Treballant d'aquesta manera, s'està barrejant la presentació amb l'estructura del document, la qual cosa complica enormement el manteniment, l'actualització i l'edició del disseny de la pàgina web.

Aplicar CSS al document XHTML

CSS dins de l'element <style> en la capçalera del document

Aquest mètode permet escriure totes les regles CSS en el document XHTML, però tancades a la capçalera del document i, per tant, separades de l'estructura i el contingut.

```
01. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
02. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
03. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">
04. <head>
05.   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
06.   <title>Introducció al XHTML i CSS</title>
07.   <style type="text/css">
08.     <![CDATA[
09.       p {
10.         color: green !important;
11.       }
12.     ]]>
13.   </style>
14. </head>
```

L'atribut type indica al navegador que el contingut respon al llenguatge CSS, i l'etiqueta <![CDATA[...]]> conté en contingut del CSS. Això és necessari segons la clàusula 4.8 de les recomanacions del W3C sobre XHTML 1.

Hi ha, però, un primer problema, que és la invisibilitat de tot el contingut del CDATA per part de navegadors com el Firefox 2.0, la solució al qual passa per eliminar l'expressió <![CDATA[...]]>.

Un altre inconvenient és el fet de no poder associar les regles del CSS a múltiples pàgines web. L'avantatge d'aquest mètode, però, se situa a nivell de desenvolupament i proves

Aplicar CSS al document XHTML

CSS externs

Aquest mètode mostra clarament la teoria de la separació entre contingut i presentació: es tracta de treballar amb dos arxius, un arxiu XHTML, que defineix l'estructura del document i l'altre, CSS, associat al primer, que defineix la seva presentació.

Un CSS extern es pot associar de diferents maneres al document XHTML:

Ús de l'etiqueta <link />

```
01. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
02. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
03. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">  
04. <head>  
05.     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
06.     <title>Curs d27</title>  
07.     <link rel="stylesheet" type="text/css" href="estil.css" />  
08. </head>
```

L'atribut href de l'etiqueta <link /> tindrà com a valor la ubicació física de l'arxiu. Aquest valor podrà ser relatiu, és a dir, en relació a la ubicació del document que el crida, tal com es veu a l'exemple, o absolut, fent servir la URL del CSS.

Aquest mètode té el gran avantatge de que si el CSS està associat a múltiples documents web, només caldrà que sigui descarregat un cop, després es trobarà a la memòria cau (cache) del navegador, i es guanyarà velocitat en la descàrrega dels documents i en la navegació.

Aplicar CSS al document XHTML

Ús de @import

De manera molt semblant al mètode anterior, la regla @import permet cridar un CSS extern. L'adreça de l'arxiu pot ser, igualment, relativa o absoluta respecte al document XHTML:

```
01. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
02. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
03. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">
04. <head>
05.   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
06.   <title>Curs d27</title>
07.   <style type="text/css">
08.     <![CDATA[
09.       @import "estil.css";
10.     ]]>
11.   </style>
12. </head>
```

Aquest mètode es beneficia de tots els avantatges del mètode <link /> amb la diferència que, versions antigues de navegadors com el Netscape o l'Internet Explorer, no suporten la regla @import, per la qual cosa, tot estil definit en @import serà ignorat. Aquesta subtil, però important característica, us permetrà definir aquells estils de les versions superiors o iguals a la 2 del CSS en el @import i deixar les regles d'estil del CSS 1 pel mètode anterior, <link />.

Aplicar CSS al document XHTML

Per tant, un exemple d'esquema óptim d'aplicació dels CSS seria la combinació dels dos darrers mètodes:

```
01. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
02. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
03. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">  
04. <head>  
05.   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
06.   <title>Curs d27</title>  
07.   <link rel="stylesheet" type="text/css" href="navegadors_antics.css" />  
08.   <style type="text/css">  
09.     <![CDATA[  
10.       @import "navegadors_actual.css";  
11.     ]]>  
12.   </style>  
13. </head>
```

En aquest esquema el full d'estil navegadors_antics.css inclouria regles relacionades amb els colors, mida de les fonts o tipus de fonts, elements que poden ser interpretats correctament pels navegadors més antics, i un altre full d'estil, navegadors_actual.css, recolliria totes aquelles regles CSS més actuals, com el posicionament, fons o meta-selectors.

Annexe Selectors CSS

Pàgina de referència sobre selectors CSS:

https://www.w3schools.com/cssref/css_selectors.asp

Es pot fer servir el seu [CSS Selector Tester](#) per a veure els selectors en acció.

| Selector | Example | Example description |
|---------------------------------|----------------------------|---|
| <code>.class</code> | <code>.intro</code> | Selecciona tots els elements amb la classe «intro» |
| <code>.class1.class2</code> | <code>.name1.name2</code> | Selecciona tots els elements que tenen les «name1» i «name2» com a classe |
| <code>.class1 .class2</code> | <code>.name1 .name2</code> | Selecciona els elements amb classe «name2» que es descendent d'un element amb classe «name1» |
| <code>#id</code> | <code>#firstname</code> | Selecciona l'element amb id «firstname» |
| <code>*</code> | <code>*</code> | Selecciona tots els elements |
| <code>element</code> | <code>p</code> | Selecciona tots els elements <code><p></code> |
| <code>element.class</code> | <code>p.intro</code> | Selecciona tots els elements <code><p></code> amb la classe «intro» |
| <code>element,element</code> | <code>div, p</code> | Selecciona tots els elements <code><div></code> i tots els elements <code><p></code> |
| <code>element element</code> | <code>div p</code> | Selecciona tots els elements <code><p></code> que estan dins d'elements <code><div></code> |
| <code>element>element</code> | <code>div > p</code> | Selecciona tots els elements <code><p></code> en que el seu pare és un element <code><div></code> |
| <code>element+element</code> | <code>div + p</code> | Selecciona els elements <code><p></code> que estan situats just després d'elements <code><div></code> |
| <code>element1~element2</code> | <code>p ~ ul</code> | Selecciona tots els elements <code></code> que son precedits per un element <code><p></code> (no necessàriament just abans) |

PHP

Què és PHP?

- PHP és un acrònim de "PHP: Hypertext Preprocessor"
- PHP és un llenguatge de scripts de codi obert àmpliament utilitzat
- Els scripts PHP s'executen al servidor
- PHP es pot descarregar i utilitzar gratuïtament

Seqüència d'execució d'una pàgina PHP

- El navegador de l'ordinador client demana el fitxer «pagina.php» a l'ordinador servidor.
- El servidor web (Apache, NginX, Lighttpd, ...) de l'ordinador servidor examina l'extensió.
 - Si te una extensió «html» busca l'arxiu i l'envia tal qual
 - Si la extensió és «php», aleshores és un fitxer escrit en PHP. El servidor interpreta el fitxer «pagina.php» amb l'interpret de PHP i genera un fitxer completament escrit en HTML.
- El servidor envia l'arxiu resultant (HTML pur) al client.

Com és un arxiu PHP?

- Els fitxers PHP poden contenir text, HTML, CSS, JavaScript i codi PHP
- El codi PHP s'executa al servidor i el resultat es torna al navegador com a HTML simple
- Els fitxers PHP tenen l'extensió ".php"

Què pot fer PHP?

- PHP pot generar contingut de pàgina dinàmic
- PHP pot crear, obrir, llegir, escriure, suprimir i tancar fitxers al servidor
- PHP pot recopilar dades del formulari
- PHP pot enviar i rebre cookies
- PHP pot afegir, suprimir i modificar dades de la vostra base de dades
- PHP es pot utilitzar per controlar l'accés dels usuaris
- PHP pot xifrar dades

Amb PHP no es limita a la sortida HTML. Podeu generar imatges, fitxers PDF i fins i tot pel·lícules Flash. També podeu enviar qualsevol text, com ara XHTML i XML.

Sintaxi bàsica de PHP

Un script PHP es pot col·locar a qualsevol lloc del document.

Un script PHP comença amb `<?php` i acaba amb `?>`:

```
<?php
// El codi PHP va aquí
?>
```

Example PHP

```
1  <!DOCTYPE html>
2  <html>
3
4  <body>
5
6      <h1>My first PHP page</h1>
7
8      <?php
9          echo "Hello World!";
10         ?>
11
12  </body>
13
14  </html>
```


Variables en PHP

Declaració de variables:

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

Regles per a variables PHP:

- Una variable comença amb el signe \$, seguit del nom de la variable
- Un nom de variable ha de començar amb una lletra o el caràcter de subratllat
- Un nom de variable no pot començar amb un número
- Un nom de variable només pot contenir caràcters alfanumèrics i guions baixos (A-z, 0-9 i _)
- Els noms de variables distingeixen entre majúscules i minúscules (\$age i \$AGE són dues variables diferents)

Variables en PHP

```
1  <!DOCTYPE html>
2  <html>
3
4  <body>
5
6      <?php
7          $color = "red";
8          echo "My car is " . $color . "<br>";
9          echo "My house is " . $COLOR . "<br>";
10         echo "My boat is " . $coLOR . "<br>";
11         ?>
12
13     </body>
14
15 </html>
```

Encara que el llenguatge PHP no és «Case sensitive» les variables si ho son, per tant no és el mateix \$color, que \$COLOR, etc.

Eixida HTML de l'exemple:

My car is red
My house is
My boat is

Eixida de variables

Estos dos fragments produeixen la mateixa eixida:

```
<?php  
$txt = "W3Schools.com";  
echo "I love $txt!";  
?>
```

```
<?php  
$txt = "W3Schools.com";  
echo "I love ".$txt."!";  
?>
```

Comentaris en PHP

Comentaris d'una línia:

- `//`
- `#`


Comentaris de bloc

- `/*` (comença)
- `*/` (acaba)

Formularis en PHP (POST)

The example below displays a simple HTML form with two input fields and a submit button:

```
1  <!DOCTYPE html>
2  <html>
3
4  <body>
5
6      <form action="welcome.php" method="post">
7          Name: <input type="text" name="name"><br>
8          E-mail: <input type="text" name="email"><br>
9              <input type="submit">
10     </form>
11
12 </body>
13
14 </html>
```



Name:

E-mail:

Formularis en PHP (POST)

Quan l'usuari emplena el formulari anterior i fa clic al botó d'enviament, les dades del formulari s'envien per processar-les a un fitxer PHP anomenat "welcome.php". Les dades del formulari s'envien amb el mètode HTTP POST.

El fitxer "welcome.php" té aquest aspecte:

```
1  <!DOCTYPE html>
2  <html>
3
4  <body>
5
6      Welcome <?php echo $_POST["name"]; ?><br>
7      Your email address is: <?php echo $_POST["email"]; ?>
8
9  </body>
10
11 </html>
```

Formularis en PHP (POST)

La sortida podria ser una cosa així:

Welcome John

Your email address is john.doe@example.com

Formularis en PHP (GET)

The example below displays a simple HTML form with two input fields and a submit button:

```
1  <html>
2
3  <body>
4
5      <form action="welcome_get.php" method="get">
6          Name: <input type="text" name="name"><br>
7          E-mail: <input type="text" name="email"><br>
8              <input type="submit">
9      </form>
10
11 </body>
12
13 </html>
```



Name:

E-mail:

Formularis en PHP (GET)

Quan l'usuari emplena el formulari anterior i fa clic al botó d'enviament, les dades del formulari s'envien per processar-les a un fitxer PHP anomenat "welcome_get.php". Les dades del formulari s'envien amb el mètode HTTP GET.

El fitxer "welcome_get.php" té aquest aspecte:

```
1  <html>
2
3  <body>
4
5      Welcome <?php echo $_GET["name"]; ?><br>
6      Your email address is: <?php echo $_GET["email"]; ?>
7
8  </body>
9
10 </html>
```

Formularis en PHP (GET)

La sortida podria ser una cosa així:

Welcome John

Your email address is john.doe@example.com

GET vs. POST

Tant GET com POST creen una matriu (per exemple, matriu (clau1 => valor1, clau2 => valor2, clau3 => valor3, ...)). Aquesta matriu conté parells clau/valor, on les claus són els noms dels controls de formulari i els valors són les dades d'entrada de l'usuari.

Tant GET com POST es tracten com `$_GET` i `$_POST`. Es tracta de superglobals, cosa que significa que sempre són accessibles, independentment de l'abast, i hi podeu accedir des de qualsevol funció, classe o fitxer sense haver de fer res especial.

`$_GET` és un conjunt de variables que es passen a l'script actual mitjançant els paràmetres d'URL.

`$_POST` és un conjunt de variables que es passen a l'script actual mitjançant el mètode HTTP POST.

Compare GET vs. POST

From: https://www.w3schools.com/tags/ref_httpmethods.asp

The following table compares the two HTTP methods: GET and POST.

| | GET | POST |
|-----------------------------|--|--|
| BACK button/Reload | Harmless | Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted) |
| Bookmarked | Can be bookmarked | Cannot be bookmarked |
| Cached | Can be cached | Not cached |
| Encoding type | application/x-www-form-urlencoded | application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data |
| History | Parameters remain in browser history | Parameters are not saved in browser history |
| Restrictions on data length | Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters) | No restrictions |
| Restrictions on data type | Only ASCII characters allowed | No restrictions. Binary data is also allowed |
| Security | GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information! | POST is a little safer than GET because the parameters are not stored in browser history or in web server logs |
| Visibility | Data is visible to everyone in the URL | Data is not displayed in the URL |

AMPLIACIÓ

Ampliació

- **Bootstrap:**

<https://getbootstrap.com/docs/4.0/getting-started/introduction/>