

Apunts UD3: CSS

lloc: CIFP Francesc de Borja Moll
Curs: Llenguatges de marques i sistemes de gestió d'informació
Llibre: Apunts UD3: CSS
Imprès per: Alejo Morell Bethencourt
Data: dilluns, 26 octubre 2020, 16:12

Taula de continguts

- 1. Fulls d'estil: CSS
 - 1.1. La sintaxi
 - 1.2. Atributs globals: class i id
 - 1.3. Selectors
 - 1.4. Ubicació dels CSS

1. Fulls d'estil: CSS

Per tal d'evitar que l'HTML fos el responsable de la part estètica i visual de la web, es van idear els que s'anomenen, fulls d'estil, és a dir, es va dissenyar el llenguatge CSS (*Cascading Style Sheets*).

En el codi HTML col·loquem el contingut, és a dir, què ha de visualitzar-se, mentre que amb CSS definim la presentació, és a dir, com ha de visualitzar-se.

Recomanació: L'adequat quan treballem amb CSS, és escriure el codi en fitxers independents, que tindran extensió **.css**. No convé col·locar codi CSS per tant dins d'arxius HTML. Hem d'evitar col·locar estils en etiquetes `<style>` en el propi codi HTML.

Introducció

El CSS, per poder fer el disseny de la pàgina, utilitza:

- Selectors. Mitjançant els quals podem especificar a quins elements de la pàgina ens estem referint. Són les referències a les etiquetes, als id's o a les classes HTML.
- Declaracions. Són les unitat bàsiques de CSS, el que significa que no es pot emprar res més petit. Consisteix bàsicament en l'assignació d'un valor a una propietat. Es podria dir, de forma col·loquial que és la resposta a una pregunta: De quin color hauria de ser els fons? per exemple: propiedad color: valor red;

Com seria una regla CSS?

En primer lloc, si volem mitjançant CSS modificar l'estil d'una o diverses etiquetes HTML en una pàgina web, haurem d'indicar a quina etiqueta o etiquetes afecta la modificació. Aquesta part del codi són els selectors. Per a indicar-los, excepte casos especials, s'escriurà el nom de l'etiqueta o una referència a ella. per exemple, si en aquesta part escrivim h1, l'estil que apliquem afectarà a totes les etiquetes h1 que hi hagi en la pàgina.

Després, s'haurà d'escriure què és el que volem canviar d'aquesta etiqueta i com. Tot això formarà un bloc, que anomenarem declaració. En CSS s'empren els signes { i } per a delimitar aquest bloc; Dins de les claus escriurem què és el que volem canviar i el nou valor que li donarem. Al que volem canviar li diem propietat, i ho indiquem mitjançant una paraula clau. Cada propietat té un valor (o varis). El valor indica la variació de la propietat, és a dir, quan o en quina ha canviat la propietat.

Exemple:
Si volem canviar la grandària dels títols amb etiqueta h1: h1 { font-size : 12px } En aquest exemple h1 indica les etiquetes a les quals hem d'aplicar aquest estil; font-size indica la propietat (la grandària de lletra), mentre que 12px és el valor, és a dir la variació de la propietat (el canvi de grandària). Com pot observar-se la propietat i e valor estan separats pel signe de dos punts.

Dins d'una mateixa declaració (signes de claus), pot haver-hi més d'una propietat amb el seu valor, aquestes han d'anar separades pel signe de punt i coma.

Exemple:

```
h1 { font-size : 12px; color : blue; }
```

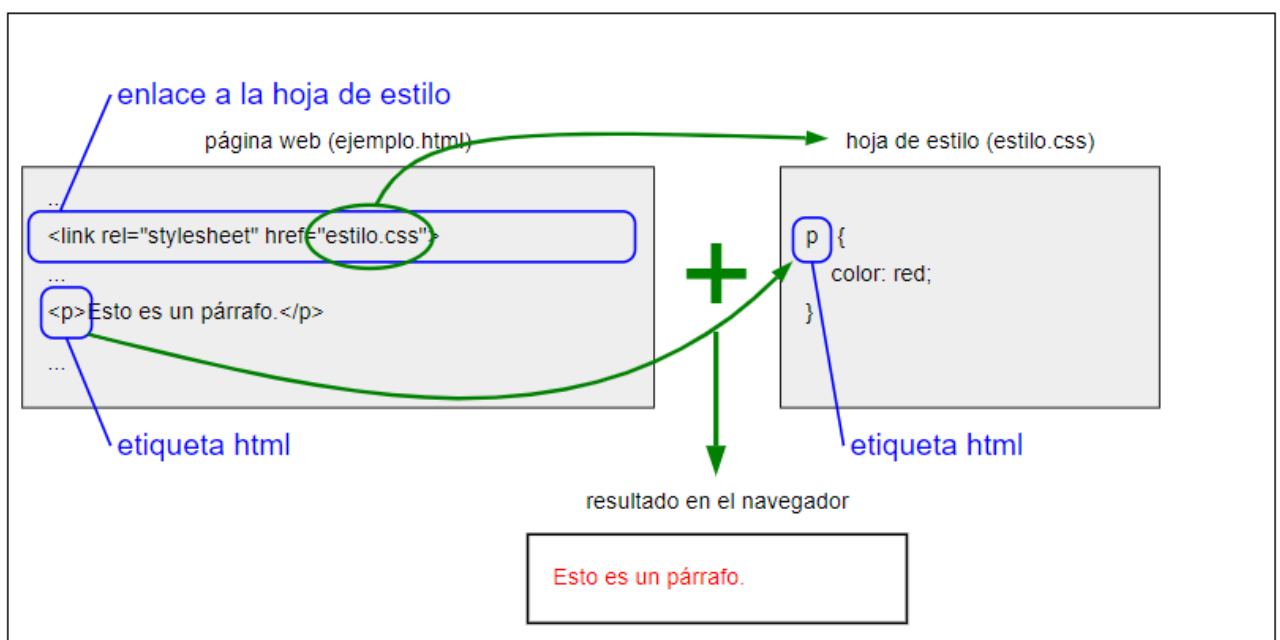
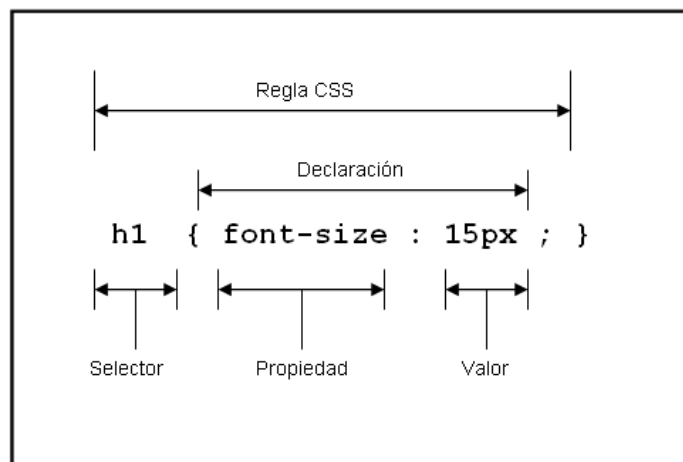
En aquest exemple les etiquetes h1 reben dues propietats, amb la primera canviem la grandària de lletra, i amb la segona canviem el color de lletra.

El signe de punt i coma, es pot posar al final de l'última propietat, encara que no és obligatori resulta convenient, per si volem afegir alguna propietat més.

El llenguatge CSS es basa en una d'estructura, que podem resumir de la següent forma: **selector { propietat : valor; propietat : valor; ... }**

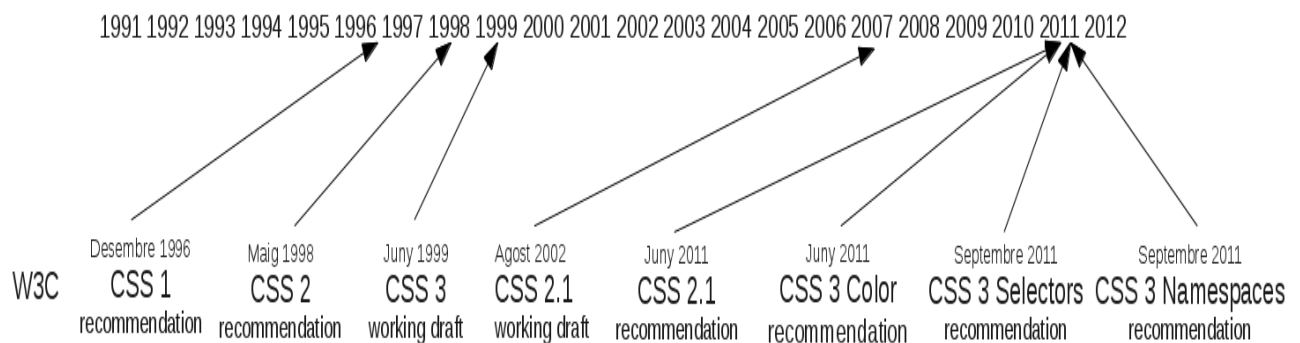
- **Conceptes importants:**

- Regles: Cada regla o conjunt de regles consisteix en un o més selectors i un bloc de declaració (o blocs d'estil).
- Dins dels blocs existeixen les propietats o elements, els quals tenen un valor determinat.
- Els estils s'apliquen als elements del document que compleixin amb el selector que els precedeix.
- Cada bloc d'estils es defineix entre claus, i està format per una o diverses declaracions d'estil amb la sintaxi.



Evolució històrica del CSS

- La primera versió del CSS o Cascading Style Sheets, va aparèixer en 1996.
- Després el maig de l'any 1998 es va publicar l'estàndard CSS versió 2.
- El 2008, se'n va fer una revisió i es va publicar el CSS versió 2, revisió 1 i es va conèixer com CSS2.1.
- Actualment està en vigor la versió 3, el CSS3. En aquest cas l'especificació està dividida en mòduls, alguns dels quals ja han aparegut estàndards i en d'altres encara s'hi està treballant, és a dir, per una part tenim estàndards i per una altra una tecnologia en continu desenvolupament.



Evolució de les versions de CSS

Com podeu observar a partir de la versió CSS3, els estàndards es tracten en mòduls per separat, com ara els , **selectors** o **namespaces**.

Avantatges de l'ús de fulls d'estil

Emprar fulls d'estil ens proporciona els avantatges següents:

- Possibilitat de mantenir el codi.
- En el camp de disseny, el CSS és més potent que les marques de disseny que ofereix L'HTML.
- El CSS és un llenguatge senzill.
- Es poden especificar diferents fulls d'estil per a un sol document HTML. Per exemple, podem tenir l'estil per a la pàgina web quan es visita amb el navegador i l'estil per a quan volem imprimir aquesta pàgina.
- Els fulls d'estil es poden reutilitzar des de diferents documents HTML.

Inconvenient de l'ús de fulls d'estil

El gran inconvenient dels fulls d'estil és que no tots els navegadors es comporten de la mateixa manera davant del mateix full d'estil. Això es deu al fet que alguns navegadors no compleixen els estàndards establerts i, amb això, obliguen el programador a codificar diferents fulls d'estil (un per a cada navegador). Tot i això, en els últims anys els navegadors cada vegada més, s'acosten més a complir els estàndards proposats.

1.1. La sintaxi

Un full d'estil és un conjunt de regles que defineixen l'estètica final dels documents HTML. Com ja hem dit abans aquestes regles estan formades per un selector i per un conjunt de declaracions.

- **Un selector:** Ens serveix per definir a quin o a quins elements volem aplicar les declaracions de la regla.
- **Una declaració:** Està formada per una propietat amb el seu valor associat. Les declaracions són les diverses característiques que han de complir els elements que concorden amb el selector. A cada propietat de les declaracions, s'hi posa un valor.

La sintaxi genèrica:

```
selector{
    declaració_1
    ...
    declaració_n
}
```

on: la sintaxi de cada declaració_i és: propietat_i:valor_i;

Exemple: si volem que tots els paràgrafs tinguin lletra de mida 10pt i un fons de color gris, definirem:

```
p {
  font-size: 10pt;
  background-color: gray;
}
```

En aquesta regla, p és el selector i té dues declaracions: font-size:10pt; i background-color:gray.

En la primera declaració font-size: 10pt, font-size és la propietat i 10pt és el valor.

En la segona declaració background-color:gray, background-color és la propietat i gray és el valor.

Les regles arrova

Hi ha un conjunt de regles especials que s'anomenen regles arrova (*at-rules* en anglès o regles-at). Aquestes regles es caracteritzen perquè comencen pel caràcter @. Algunes d'aquestes regles (@import o @namespace) han d'aparèixer al principi de la fulla d'estil, i les altres (@font-face, @media, etc.), poden posar-se a qualsevol part de la fulla d'estil (dins d'un full extern o incrustat).

Vegem quines són aquestes regles i com s'utilitzen:

- **@import:** La regla @import ens serveix per incloure, en el nostre full d'estil, fulls d'estil externs.

Exemple:

Si, per exemple, volem incloure en el nostre full d'estil totes les propietats que hi ha en un document anomenat "housestils.css", hem d'escriure la línia següent:

```
@import "housestils.css";
```

- **@media**: La regla **@media** serveix per diferenciar per quin mitjà s'ofereixen les propietats que conté aquesta regla. La sintaxi genèrica és la següent:

```
@media mitjà{  
    propietats  
}
```

on mitjà pot ser print (per imprimir) o screen (per mostrar per pantalla), entre d'altres.

Exemple: Volem que, quan imprimim el document HTML, la lletra tenguí una mida de 12pt, però que, quan es vegi per pantalla, tenguí una mida de 14pt. També volem que en els dos casos, l'alçada de la línia sigui d'1.4. En el nostre full d'estil hem d'introduir el codi següent:

```
@media print {  
    body { font-size: 12pt }  
}
```

```
@media screen {  
    body { font-size: 14pt }  
}
```

```
@media screen, print {  
    body { line-height: 1.4 }  
}
```

Avui en dia, amb l'augment dels tipus de dispositius, fa que tinguem moltes mides de pantalles diferents (smartphones, tablets, ordinadors, etc.), i aquesta regla és molt útil per aplicar diferents regles CSS segons el tipus de pantalla o la seva orientació. Aquest tipus de programació se l'ha anomenat Disseny Adaptatiu (en anglès *Responsive Web Design* o RWD), i a la combinació de la regla **@media** amb les condicions que podem afegir, se l'ha anomenat *media-queries*.

- **@font-face**: Especifica una font no inclosa en el navegador, que l'usuari es descarregarà.

Exemple: En aquest exemple, en la regla `@font-face` estem definit una font. Per tal de definir-la li posem un nom. Aquest nom es posa amb la propietat `font-family` i en aquest cas hem triat el nom `DeliciousRoman`. A més, és imprescindible especificar on es troba aquesta font, per tal que el navegador la pugui descarregar i emprar (això es fa amb la propietat `src`)

```
@font-face {
  font-family: DeliciousRoman;
  src: url("Delicious-Roman.ott");
}

p {
  font-family: DeliciousRoman, Helvetica, Arial, sans-serif
}
```

Si hem definida la font, després la podrem emprar en qualsevol regla.

La seqüència de font dins l'atribut `font-family`, vol dir s'intentarà primer aplicar la primera font, si no es pot aplicar, se intentarà amb la segona, i així successivament.

- `@charset`: Especifica quin és el joc de caràcters que farem servir dins del fitxer CSS:
 - `@charset "UTF-8"`: Activa el joc de caràcters pel full d'estil a Unicode UTF-8.
 - `@charset 'ISO-8859-15'`: Activa el joc de caràcters pel full d'estil Latin-9 (Llengües de l'oest d'Europa, amb el símbol de l'euro).
- `@supports`: Ens permet detectar si el navegador de l'usuari suporta o no les noves funcionalitats del CSS. El seu funcionament seria:

Exemple: El CSS aplica les regles només si el navegador suporta `display: flex`. En cas contrari ho ignora per complet.

```
/* si el navegador suporta display: flex */
@supports (display: flex) {
  /* Llavors aplica les regles: */
  .element {
    display: flex;
    ... Més regles
  }
}
```

* Display

- `@page`: La fem servir per modificar propietats CSS a l'hora d'imprimir un document, però hem de tenir en compte que pot actuar sobre un nombre de propietats: sobre els marges, les línies vídues, orfes i els salts de pàgina

Exemple:
<pre>@page { margin-left: 3cm; } @page :left { margin-left: 4cm; }</pre>

Llista de propietats CSS estables

- <https://www.mclibre.org/consultar/htmlcss/css/css-propiedades.html>

Els comentaris

Com en tot llenguatge, la llegibilitat és imprescindible si treballen en grup. És per això que CSS ofereix una manera de comentar els fulls d'estil. Si en un full d'estil volem posar comentaris destinats a l'aclariment del codi CSS, hem de fer servir la sintaxi següent:

/* comentaris */

Exemple:
<pre>/* Estil per a totes les capçaleres del document */ h1 { text-align: center; /* Text centrat per destacar la importància de la capçalera*/ color: red; /* Color de lletra vermella per emfatitzar el text */ }</pre>

1.2. Atributs globals: class i id

Els atributs id i class, serveixen com a identificadors a HTML, són molt importants perquè els elements HTML puguin ser apuntats i personalitzats amb estils mitjançant CSS.

class

Donat que el nombre d'etiquetes d'HTML és reduït, es probable que dins d'una pàgina web, es vulgui donar un format distint a elements amb la mateixa etiqueta. Per fer-ho, podem emprar les classes, que poden assignar a qualsevol element d'una pàgina web.

L'atribut global class, permet especificar les classes que s'aplicaran a un determinat element. Les classes permeten a CSS i Javascript, seleccionar i accedir a elements específics a través dels selectors de classe o funcions com el mètode DOM (es veurà en el segon bloc del curs)

Encara que l'especificació no posa requisits en el nom de les classes, s'anima als desenvolupadors web a utilitzar noms que descriguin el propòsit semàntic de l'element, en lloc de la presentació de l'element (Per exemple, *atribut* per descriure un atribut en lloc de *cursiva*, encara que un element d'aquesta classe pot presentar-se en *cursiva*). Els noms semàntics segueixen sent lògics encara que la presentació de la pàgina canviï.

Class s'empra per donar format a un element d'HTML, i van definides dins les fulles d'estil.

Codi
Fragment HTML: <pre><p>- Haga el favor de poner atención en la primera cláusula porque es muy importante. Dice que ... la parte contratante de la primera parte será considerada como la parte contratante de la primera parte. ¿Qué tal, está muy bien, eh?</p> <p class="aviso">- No, eso no está bien. Quisiera volver a oírlo.</p></pre>
CSS <pre>p.aviso { color: red; }</pre>
Resultat
<p>- Haga el favor de poner atención en la primera cláusula porque es muy importante. Dice que ... la parte contratante de la primera parte será considerada como la parte contratante de la primera parte. ¿Qué tal, está muy bien, eh?</p> <p>- No, eso no está bien. Quisiera volver a oírlo.</p>

Crearem una etiqueta de tipus <button> corresponent a un botó i afegirem l'atribut "class" amb valor vermell (class="vermell").

A continuació crearem la classe corresponent al valor "vermell":

- Per a crear una classe d'estil CSS:
 1. Afegir les etiquetes <style> y </style>. (se col·loquen al <head></head>)
 2. Dins de l'etiqueta <style> crear una classe (un punt seguit del nom de la classe)
 3. Dins de les clau { }, afegir parell(s) atribut/valor (l'atribut sempre a l'esquerra).
 4. Tancam amb punt i coma.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .vermell {
        background-color: red;
      }
    </style>
  </head>
  <body>
    <button class="vermell"> Botó vermell</button>
  </body>
</html>
```

id

L'atribut Id a HTML, és un identificador únic per un element.

L'atribut global id, defineix un identificador únic (ID) que ha de ser únic en tot el document. El seu propòsit és identificar l'element quan es vincula (usant un identificador de fragment), scripts, o estil (CSS).

El valor d'aquest atribut no ha de contenir espais en blanc (espais, tabulacions, etc.). Els navegadors tracten IDs no conformes que contenen espais en blanc com si l'espai en blanc fora part de l'ID. En contrast amb l'atribut class, que permet valors separats per espais, els elements només poden tenir un únic ID.

Exemple:
<pre> <!DOCTYPE> <html> <head> <meta charset="UT F-8"> <title>Aplicació amb id i class </title> <style> div#principal { border: 1px solid blue; padding: 20px; margin: 0 25%; font-family: Tahoma, sans-serif; } .destacat { font-size: 1.3em; color:red; } </style> <body> <div id="principal"> <p>Aquesta és la secció principal. Tot el contingut apareixerà dins un requadre.</p> <p>Seleccionam un <div> amb un atribut id., i un amb un atribut class. Els estils són:</p> Propietat border Propietat padding Propietat margin Propietatfont-family </div> </body> </html> </pre>

L'atribut, al contrari que l'atribut class, ha de tenir un valor únic per un element. A l'atribut Class, es pot aplicar a diferents elements elements.

Exemple:
<pre> <html> <head> <meta charset="utf-8"> <title>Atributs global: id i class</title> </head> <body> <h1 id="titol-principal" class="verd-turquesa">Pàgina sobre bicicletes</h1> <h2 id="subtittol-principal" class="tittol">Anem amb bici per la ciutat</h2> <p class="descripcio">Al mercal trobam molts de tipus de bicicletes.</p> <p class="descripcio">Bicicletes per atletes, per nins, per somniadors, etc.</p> <p class="descripcio">La bicicleta ens permet der lliures</p> </body> </html> </pre>

1.3. Selectors

Al fulls d'estil (CSS), els selectors són la part de les regles que indiquen al navegador a quins elements s'ha d'aplicar les propietats incloses dins de les declaracions.

Selectors bàsics

- **Selector universal:**

S'utilitza per a seleccionar tots els elements de la pàgina. Se indica mitjançant un asterisc (*). És molt senzill, però no s'emptra habitualment, ja que és difícil que un mateix estil es pugui aplicar a tots els elements d'una pàgina. Però si es combina amb altres selectors.

Exemple: eliminar el marge i l'emplenament de tots els HTML:

```
* {  
  margin: 0;  
  padding: 0;  
}
```

- **Selector de tipus o etiqueta:**

Selecciona tots els elements de la pàgina on l'etiqueta indicada coincideix amb el valor del selector. Per utilitzar aquest selector, només és necessari indicar el nom d'una etiqueta HTML (sense els caràcters < i >) corresponent als elements que es volen seleccionar.

Exemple: aplica diferents estils als titulars i als paràgrafs d'una pàgina.

```
h1 {  
  color: red;  
}  
  
h2 {  
  color: blue;  
}  
  
p {  
  color: black;  
}
```

Si es volen aplicar els mateixos estils a dues etiquetes diferents, es poden encadenar els selectors:

```
h1 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
h2 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

En aquest cas, CSS permet agrupar totes les regles individuals en una sola regla amb un selector múltiple. Es posen tots els selectors separats per comes:

```
h1, h2, h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

- **Selector de classe:**

S'utilitzen quan volem que alguns elements amb la mateixa etiqueta a la qual s'ha donat ja un estil, en tenguin un altre de forma excepcional.

Exemple:
<p>Si tenim el següent estil:</p> <pre>h1, h2, h3 { color: #000033; <!-- assigna el color blau a h1,h2,h3--> }</pre> <p>Però volem que alguns dels encapçalaments <h2> es vegin en vermell i negreta. En aquest moment és quan podem emprar els selectors de classe. El procediment seria el següent:</p> <p>1.- En primer lloc, assignarien (dins del codi HTML), a <h2>, l'atribut de classe propi: class= '...': <h2 class="vermell-negreta">Encapçalament vermell i negreta</h2></p> <p>2. En segon lloc, a la fulla d'estil, escrivim la instrucció especial que s'ha d'aplicar a la classe que acabam de definir, és a dir, afegim el nom de la classe al selector de l'element separat per un punt i sense espais en blanc:</p> <pre>h2.vermell-negreta { color: #330000; font-weight: bold; }</pre> <p>Podem anar més enllà, podem generalitzar més l'estil per si a aquest, el volem aplicar a altres etiquetes. Es tracta de no vincular la classe a cap etiqueta:</p> <pre>.vermell-negreta { color: #330000; font-weight: bold; }</pre> <p>I d'aquesta manera, es podrà assignar aquesta classe a qualsevol etiqueta HTML:</p> <pre><p class="vermell-negreta">Aquest paràgraf està en vermell i en negreta</p></pre>

- **Selectors Id:**

A vegades, serà necessari aplicar estils a un únic element de la pàgina. Encara que es podria emprar un selector de classe per aplicar estils a un únic element, els selectors id són més eficients en aquest cas. Aquest selectors permeten seleccionar un element de la pàgina a través del valor del seu atribut. Aquests tipus de selectors només seleccionen un element de la pàgina perquè el valor de l'atribut id no es pot repetir dins dos elements diferents de la una mateix pàgina. La sintaxi és molt similar a la dels selectors de classe, amb la diferència que utilitzen el símbol # en lloc del punt(.).

Exemple:
<pre><h1 id="verd-negreta">Títol en verd i en negreta</h1></pre> <p>La instrucció en el CSS seria:</p> <pre>h1#verd-negreta { color: #003300; font-weight: bold; }</pre>

- **Selector descendent:**

Selecciona els elements que es troben dins d'altres elements.

Exemple: El selector del següent exemple, selecciona tots els elements `` de la pàgina que es troben dins d'un element amb etiqueta `<p>`.

```
p span {  
color: red;  
}
```

Si el codi HTML és:

```
<p>  
...  
<span>text1</span>  
...  
<a href="">...<span>text2</span></a>  
...  
</p>
```

El selector: `p span`, selecciona tant "text1" com "text2".

El motiu: és que en el selector descendent, un element no ha de ser descendent directe de l'altre.

L'única condició és que un element ha d'estar dins d'un altre element (sense importar el nivell de profunditat en el que es trobi).

A la resta d'elements `` de la pàgina que no es troben dins de l'element `<p>`, no se'ls hi aplicaria.

Els selectors descendents sempre estan formats per dos o més selectors separats entre sí per espais en blanc, on el darrer selector indica l'element on s'apliquen els estils i tots els selectors anteriors indiquen el lloc on s'ha de trobar aquest element.

Exemple: 4 selectors

```
p a span em {  
text-decoration: underline;  
}
```

Els estils s'apliquen als elements de tipus `` que es troben dins d'elements de tipus ``, que a la vegada es troben dins elements de tipus `<a>` els que es troben dins d'elements de tipus `<p>`.

- **Combinació de selectors:**

Es permet la combinació d'un o més tipus de selectors per restringir l'àmbit d'aplicació de les regles del CSS.

Exemple:

Si tenim un estil que només selecciona aquells elements amb una classe anomenada especial (class="especial") que es troba dins qualsevol element amb un classe anomenada avis (class="avis"), tendriem:

```
.avis .especial {
```

```
  ...
```

```
}
```

Si es modifica per:

```
div.avis span.especial {
```

```
  ...
```

```
}
```

Aquest estil, defineix que el selector només selecciona aquells elements de tipus amb un atribut class="especial" que estiguin dins qualsevol element de tipus <div> que tengui un atribut class="avis".

1.4. Ubicació dels CSS

El CSS ens defineix quin aspecte han de tenir els elements HTML. És per això que hem d'associar d'alguna manera aquestes propietats als elements.

Podem ubicar les propietats CSS en diferents localitzacions:

Al mateix document HTML

Afegint les propietats CSS directament a l'element emprant l'atribut style.

Exemple: Volem que un paràgraf estigui centrat amb lletra vermella. Hem de tenir en compte que aquesta forma d'incloure codificació CSS s'ha de fer servir puntualment, o per fer proves, però no permet reutilitzar el codi en diverses etiquetes.

El codi html seria el següent

```
<p style="text-align:center; color:red">Paràgraf centrat vermell</p>
```

Emprant CSS:

A la capçalera del document HTML i podem posar les diferents propietats CSS dins de l'element <style> que està ubicat dins de l'element <head>. Per exemple, si volem que tots els paràgrafs estiguin centrats i amb lletra vermella, el codi és el següent

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" lang="ca">
    <title>Document HTML </title>
    <style>
      p {
        text-align:center; color:red
      }
    </style>
  </head>
  <body>
    <p>Paràgraf centrat vermell</p>
    <p>Paràgraf centrat vermell</p>
    <p>Paràgraf centrat vermell</p>
  </body>
</html>
```

En un document extern

Posam totes les propietats dins d'un document amb extensió .css i des del document HTML enllacem aquest full d'estil amb l'ajuda de l'etiqueta <link>, dins la capçalera <head>.

Exemple: Si volem tenir tots els paràgrafs centrats amb lletra vermella, el document HTML és:

El codi html hauria de ser:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" lang="ca">
    <link rel="stylesheet" href="estils.css" />
  </head>
  <body>
    <p>Paràgraf centrat vermell</p>
  </body>
</html>
```

El fitxer estils.css té el contingut següent

```
p {
  text-align:center; color:red;
}
```

L'avantatge d'emprar documents externs per establir les propietats CSS, és que podem reutilitzar un mateix full d'estil per a diferents documents (X)HTML.