Selectors

Introducció

Com que no volem estilitzar tots els nostres elements HTML alhora, hem de poder **orientar** un subconjunt d'aquests elements HTML.

Els selectors CSS defineixen *a quins* elements volem que s'apliqui el nostre estil.

Selectors d'etiquetes genèrics

Orientar les etiquetes HTML genèriques és fàcil: només cal que utilitzeu el nom de l'etiqueta.

```
a{ /* Enllaços */ }
p{ /* Paràgrafs */ }
ul{ /* Llistes no ordenades */ }
li{ /* Elements de llista */ }
```

Hi ha una connexió directa entre el *nom* de l'etiqueta HTML i el *selector* CSS utilitzat.

Class (I)

Tenint en compte que probablement no volem estilitzar tots els paràgrafs o tots els títols de manera idèntica, hem de *diferenciar-los*.

De tots els atributs HTML, l'atribut `class` és el més important per a CSS. Ens permet definir un **grup** d'elements HTML als quals podem *orientar específicament*. Només cal que poseu un punt `.` davant del nom de la classe que voleu utilitzar.

Class (II)

```
.data {
  color: red;
}
```

D'una banda, hi ha l'atribut HTML `class` amb el valor `data`. Ha de coincidir amb el nom de la classe CSS.

Podeu utilitzar qualsevol nom per a la vostra classe CSS, sempre que no comenci amb un número.

Class (III)

El selector de classe `.data` orientarà tots els elements HTML que tinguin l'atribut `class="data"`. Per tant, els següents elements HTML tindran un estil, **tots dos**:

```
<p class="data">
  Dissabte 21 de febrer
</p>
<p>
  L'esdeveniment serà el <em class="data">dissabte</em>.
</p>
```

Tingueu en compte que en aquest exemple nom de l'etiqueta és **irrellevant**. Només ho és l'atribut HTML `class`.

Class (IV)

És possible assignar a l'atribut HTML `class` més d'un valor. Per exemple:

```
.error {
  color: red;
}
.destacat {
  background-color: yellow;
}
```

```
<p class="error destacat">Password incorrecte!</p>
```

Aquest paràgraf serà vermell amb groc com a color de fons.

Id

També podeu utilitzar l'atribut `id` al vostre HTML i orientar-lo amb un hash `#` al vostre CSS:

```
#tagline{ color: orange;}
```

```
<h1 id="tagline">Aquest títol serà taronja.</h1>
```

Els id són similars a les classes, ja que es basen en un atribut HTML.

Atenció! L'`id` d'un element és únic dins d'una pàgina!

Combinant selectors (I)

Reutilitzem el nostre exemple anterior on volem que les nostres dates siguin vermelles:

```
.data {
  color: red;
}
```

```
<p class="data">
  Dissabte 21 de febrer
</p>
<p>
  L'esdeveniment serà el <em class="data">dissabte</em>.
</p>
```

Què passa si volem que les nostres dates que estan en elements `em` siguin blaus?

Combinant selectors (II)

Podem **afegir** la següent regla CSS:

```
em.data {
  color: blue;
}
```

El `em.data` combina:

- un selector d'etiquetes `em`
- un selector de classe `.data`

Només s'aplicarà als elements HTML `<em class="data"></em>`. **No** afectarà altres elements `.data` o a les altres etiquetes `em`.

Selectors de jerarquia

Un **espai** en un selector defineix una relació avantpassat/descendent. Suposem que volem que els enllaços de la nostra capçalera estiguin en vermell:

```
header a {
  color: red;
}
```

Això es pot llegir de dreta a esquerra com: *"Seleccioneu tots els elements* `a` *que es troben dins d'un element* `header` *"*. Això evitarà que tots els altres enllaços (que no estiguin al header) es vegin afectats.

Selectors de pseudo-classe

Els elements HTML poden tenir diferents **estats**. El cas més comú és quan passeu el cursor per sobre d'un enllaç. En CSS és possible aplicar un estil diferent quan es produeix un esdeveniment d'aquest tipus.

Els selectors de pseudoclasse s'adjunten als selectors habituals i comencen amb un **de dos punts** `:`:

```
a {
  color: blue;
}

a:hover {
  color: red;
}
```

Herència

Suposem que volem canviar el **color del text** d'una pàgina web. Seria molt pessat especificar un color per a *cada*_ element HTML:

```
p,
ul,
ol,
li,
h1,
h2,
h3,
h4,
h5,
h6{color: grey;}
```

La solució és fer servir l'**herència**

Propagació del valor

El valor `color` es pot heretar d'un **avantpassat**. Tenint en compte que volem alterar la *tota* pàgina web, triarem l'avantpassat de tots els elements HTML, l'etiqueta `body`:

```
body{ color: grey;}
```

Tots els elements fills i descendents **heretaran** el valor `grey` del seu ancestre comú `body`, que de manera natural engloba *tots* els elements.

També podríem utilitzar l'etiqueta `html`.

Propietats heretades

Només algunes propietats CSS es poden heretar dels avantpassats. Són principalment propietats de **text**:

- color del text
- lletra (família, mida, estil, pes)
- alçada de línia

Alguns elements HTML no hereten dels seus avantpassats. Els enllaços, per exemple, no hereten la propietat `color`.

Ordre de les regles CSS

Si hi ha selectors similars al vostre CSS, l'últim definit tindrà prioritat.

```
p{ color: green;}
p{ color: red;}
/* Els paràgrafs seran vermells */
```

Conflictes (I)

Un element HTML es pot estilar mitjançant **diverses regles CSS**. Utilitzem un paràgraf senzill per exemple:

```
<p class="missatge" id="introduccio">
  Benvingut a la meva pàgina web
</p>
```

Podem modificar aquest paràgraf només utilitzant el seu **nom de l'etiqueta**, o el seu **nom de classe** i pel seu **id**:

```
p{ color: blue;}

.message{ color: green;}

#introduccio{ color: red;}
```

Conflictes (II)

Com que el navegador només pot triar **un color** per aplicar en aquest paràgraf, haurà de decidir quina regla CSS té **prioritat** sobre d'altres. D'això es tracta la prioritat CSS (o *especificitat* de CSS).

En el nostre exemple, el paràgraf serà **vermell** perquè un selector `#id` és més *específic* i, per tant, més **important** que altres selectors.

Com evitar conflictes

Mentre escriviu el vostre CSS, és fàcil escriure **regles en conflicte**, on la mateixa *propietat* s'aplica diverses vegades.

Per evitar-ho:

- només utilitzeu **classes**: utilitzeu `.introduction` en comptes de `#introduction`, encara que aquest element només aparegui una vegada a la vostra pàgina web
- eviteu aplicar **diverses classes** en un sol element HTML: no escriviu `<p class="big red important">` sinó `<p class="title">` que és més semànticament descriptiu
- no utilitzeu **estils en línia** com `<div style="background: blue;">`