Style sheets

Consorci d'Educació
de Barcelona
Generalitat de Catalunya
Ajuntament de Barcelona

XHTML & CSS

Initially, HTML was easy to learn and very limited in terms of marks and structure.

Everything changed when the first modern browsers appeared, they were able to represent graphical resources and animations.

XHTML & CSS

The number of websites began to grow and with it the number of HTML tags. The main goal was to create more attractive websites, and for that, HTML had to include new tags for the visual effects.

To prevent that (X)HTML was responsible of sthetic and visual part of the web, the **CSS** (Cascading Style Sheets) language was designed.

XHTML & CSS

Using style sheets gives us the following benefits:

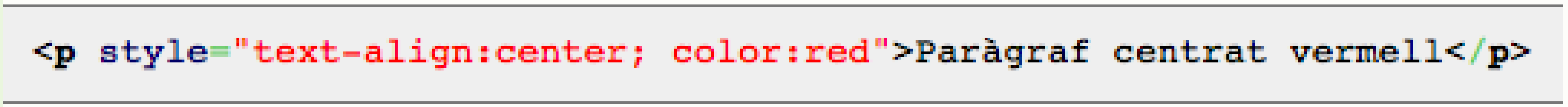
- Possibility to keep the HTML code clear.
- For design, CSS is more powerful than (X)HTML.
- CSS is a simple language.
- Style sheets can be reused from different (X)HTML documents.

The big problem of style sheets is that **not all browsers behave the same way** in front of the same style sheet; because some browsers do not follow the established standards, forcing the programmers to encode different style sheets (one for each browser). However, in recent years, browsers are closer with the proposed standards.

XHTML & CSS

Where we define the CSS properties:

- ***Inside the XHTML tag***, using the **style attribute**.

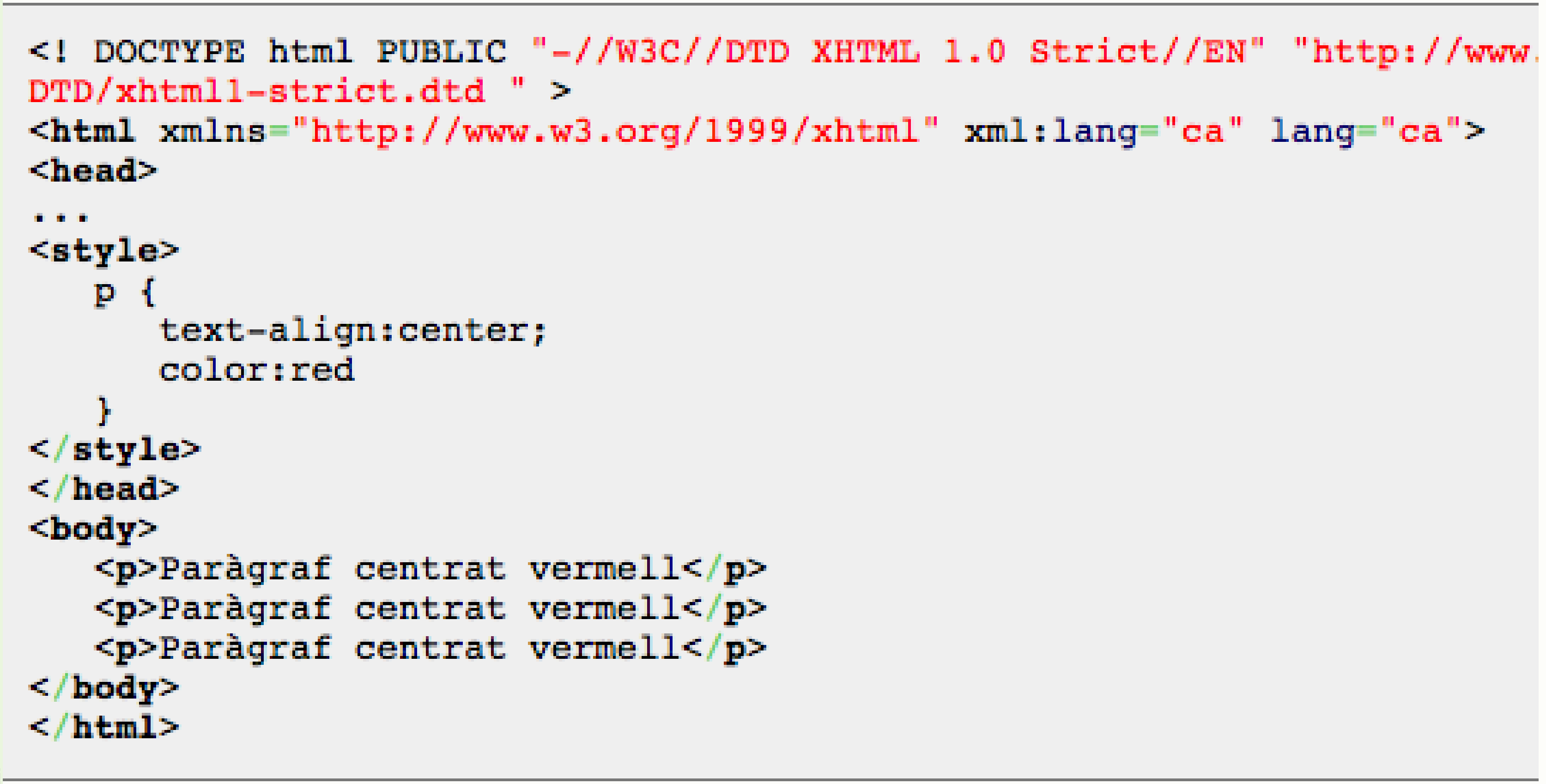


```
<p style="text-align:center; color:red">Paràgraf centrat vermell</p>
```

XHTML & CSS

Where we define CSS properties:

- In the **XHTML header**, adding different CSS properties inside the <*style*> *element* located in the <*head*> *element*.

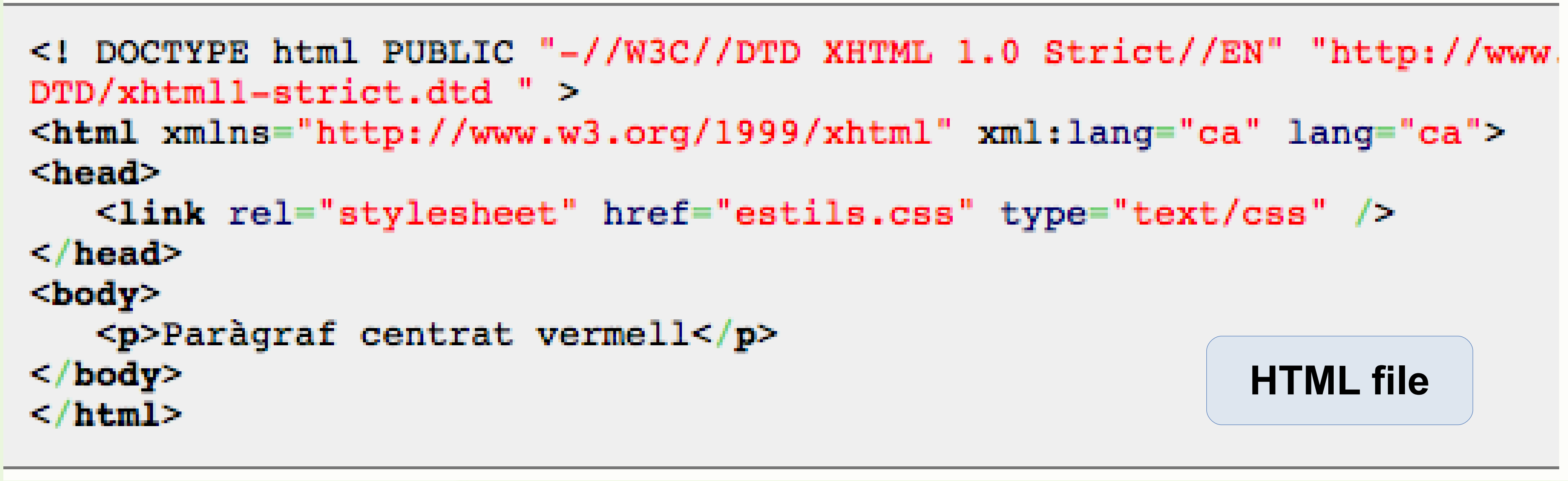


```
<! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.
DTD/xhtml1-strict.dtd " >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">
<head>
...
<style>
   p {
      text-align:center;
      color:red
   }
</style>
</head>
<body>
   <p>Paràgraf centrat vermell</p>
   <p>Paràgraf centrat vermell</p>
   <p>Paràgraf centrat vermell</p>
</body>
</html>
```

XHTML & CSS

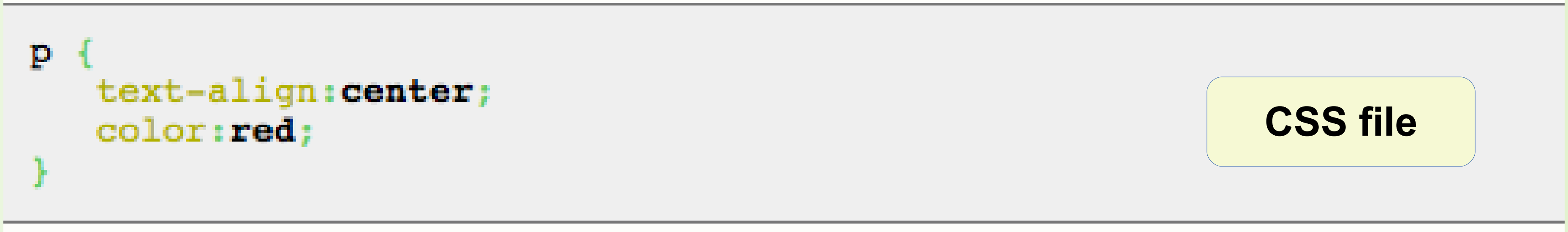
Where we define CSS properties:

- In an **external document**, adding all the properties inside the document with *.css extension*. From the (X)HTML document we link this style sheet with the *<link> tag*, in the *<head> element*.



```
<! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.
DTD/xhtml1-strict.dtd " >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ca" lang="ca">
<head>
   <link rel="stylesheet" href="estils.css" type="text/css" />
</head>
<body>
   <p>Paràgraf centrat vermell</p>
</body>
</html>
```

HTML file

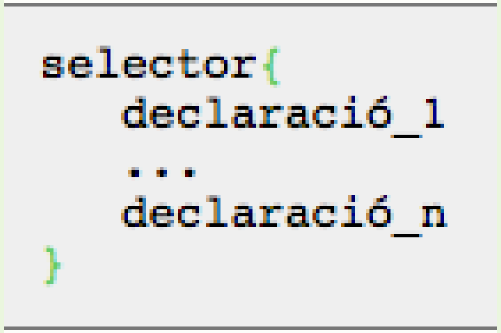


```
p {
   text-align:center;
   color:red;
}
```

CSS file

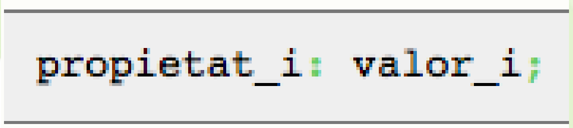
XHTML & CSS

The generic syntax of a rule is:



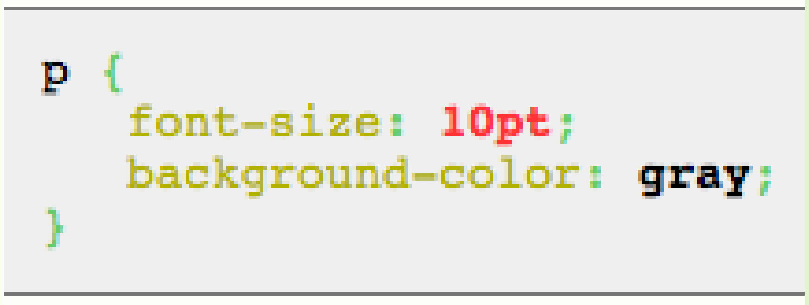
```
selector{
   declaració_1
   ...
   declaració_n
}
```

Where the syntax of the statements are:



```
propietat_i: valor_i;
```

Example:

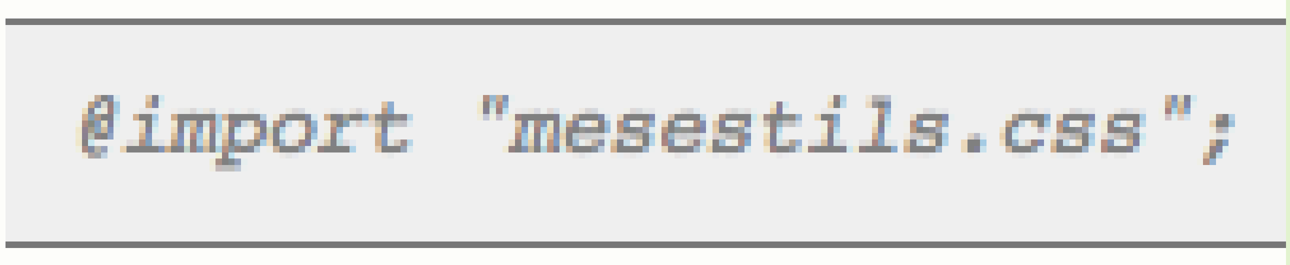


```
p {
   font-size: 10pt;
   background-color: gray;
}
```

XHTML & CSS

There is a set of special rules called ***at-rules***. These rules are characterized because they start with the **@** character.

@import: Used to include external stylesheets in our stylesheet. If, for example, we want to include in our stylesheet all the properties written in another (mesestils.css), we must write the following line:

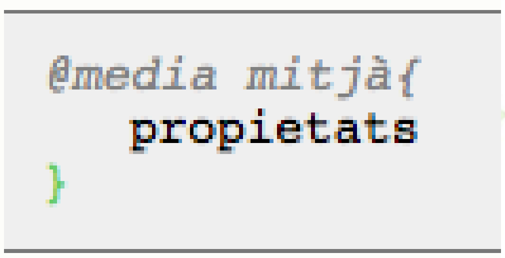


```
@import "mesestils.css";
```

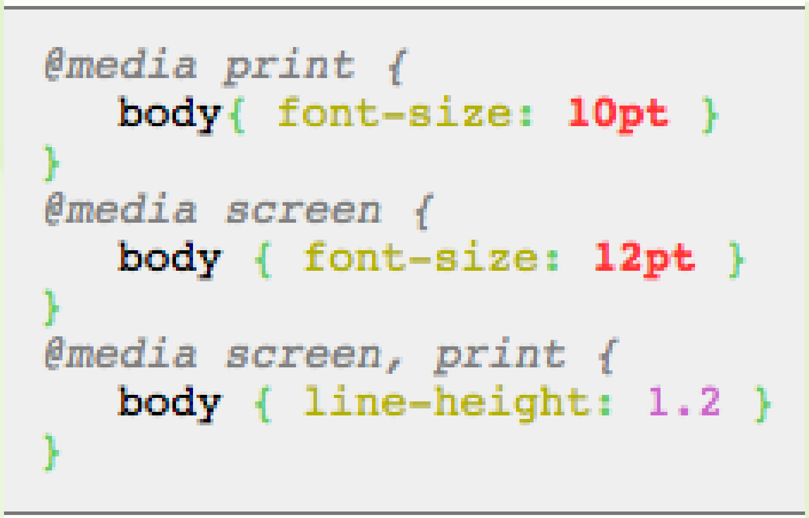
XHTML & CSS

@media: Used to indicate by which device the properties will be offered.

It should be a **print** (to print in paper the content) or **screen** (in different sizes).



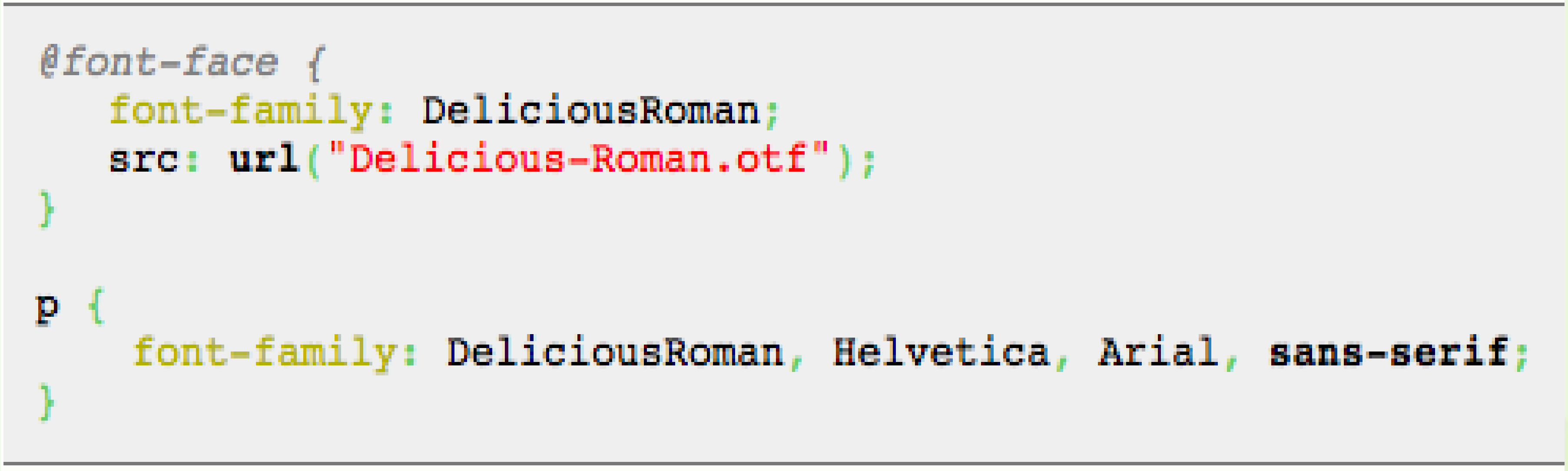
```
@media mitjà{
    propietats
}
```



```
@media print {
    body{ font-size: 10pt }
}
@media screen {
    body { font-size: 12pt }
}
@media screen, print {
    body { line-height: 1.2 }
}
```

XHTML & CSS

@font-face: Specifies a new font type not included in the browser that the user will download.



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

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

Exercise 1

Create a personal resume (a html file with lists), and modify:

- **In each html** tag ‘h1’, ‘h2’…:
  - The font type.
  - The font color.
- **In** the html file **head**:
  - The background color of each ‘li’.
  - The font of each ‘li’.
  - The background color of the web page.
- **In** an **external** html **file**:
  - The background color of the web page for an image.
  - The font size of each ‘li’.

XHTML & CSS

Two of the characteristics that make style sheets so powerful are the cascade and the inheritance.

The **cascade** refers to the possible combination of different style sheets.

Inheritance refers to the ability of the elements in the (X)HTML document to inherit properties from its predecessor elements.

Cascade

If there are conflicting properties, **which property has priority?**

In CSS a hierarchy has been established in the properties:

1) Properties in the **style attribute inside an element**.
2) Properties in the **<style> element** of the (X)HTML document.
3) Properties in an **external stylesheet**.
4) **User-defined properties**. Browsers allow the user to set different style properties.
5) **Browser-defined properties**. Browsers have a default style for each of the (X)HTML elements. If no property has been set in the above items, the browser properties are the prevailing ones

Inheritance

- **Not all** properties are inherited, this feature is described in the corresponding CSS specification.
- If we want to force **force inheritance** in a property we can enter the value inherit.
- If we put a **new value on a property**, that value prevails over the inherited value.

Basic format properties

- **Measures and colors**
- Font types
- Text aspect
- Colors and backgrounds

Measures & colors

- Some of the properties that we can set in the stylesheet have a unit of measure that indicates the size of the property:
  - Font sizes,
  - widths
  - margins.
- These units can be relative or absolute:

- **relative units**: we can measure with **em** and **%**.
- **absolute units**: **px** (pixels) **cm** (centimeters), **mm** (millimeters), **in** (inches) or **pt** (points) can be measured.

Relative mesures

- If we use the measure ***px*** (pixels), a defined distance does not change if the user changes the size of the text in his browser or if the user changes the size of the browser window, but it does change if the user changes the screen resolution.

- Instead, a distance defined in ***em*** or ***percentage*** is proportional to the size of the browser window.

Colors

- Some properties have a **color** as a value (font-color, background-color...).
- **Colors** can be expressed with **keywords** (red, green, blue...) or with the **RGB model**.
- If we use keywords we are **limited** to using a certain number of colors, those that have an associated name. Instead, with the RGB formula, we can get the full range of colors.
- To express a color with the RGB format, we have two ways to do it: for example, if we want to give the **color red**:

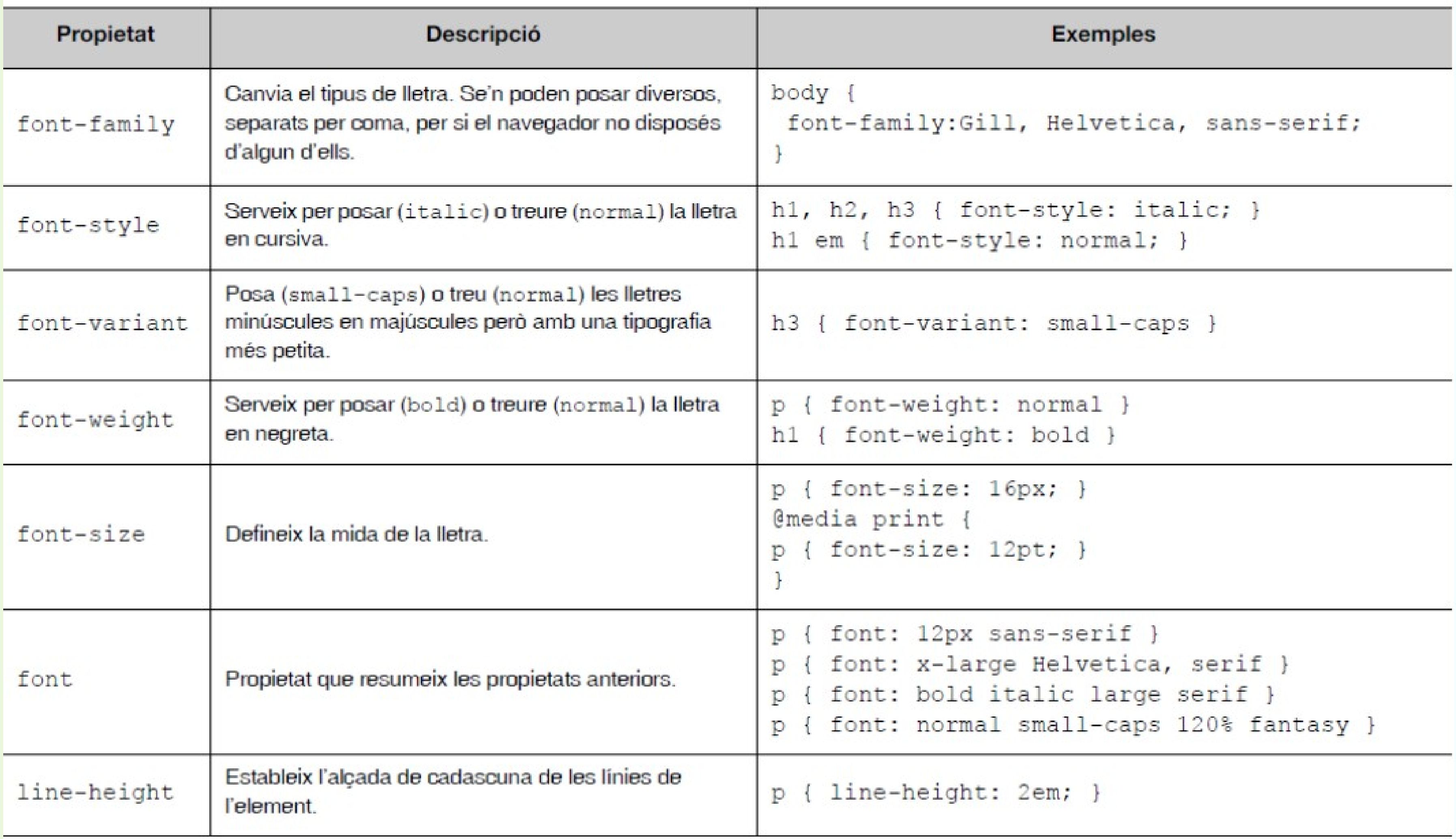
# ff0000

rgb (255,0,0)

Basic format properties

- Measures and colors
- **Font types**
- Text aspect
- Colors and backgrounds

Fonts



Propietat	Descripció	Exemples

font-family	Canvia el tipus de lletra. Se'n poden posar diversos, separats per coma, per si el navegador no disposés d'algun d'ells.	body {
		 font-family:Gill, Helvetica, sans-serif;
		}

font-style	Serveix per posar (italic) o treure (normal) la lletra en cursiva.	h1, h2, h3 { font-style: italic; }
		h1 em { font-style: normal; }

font-variant	Posa (small-caps) o treu (normal) les lletres minúscules en majúscules però amb una tipografia més petita.	h3 { font-variant: small-caps }

font-weight	Serveix per posar (bold) o treure (normal) la lletra en negreta.	p { font-weight: normal }
		h1 { font-weight: bold }

font-size	Defineix la mida de la lletra.	p { font-size: 16px; }
		@media print {
		p { font-size: 12pt; }
		}

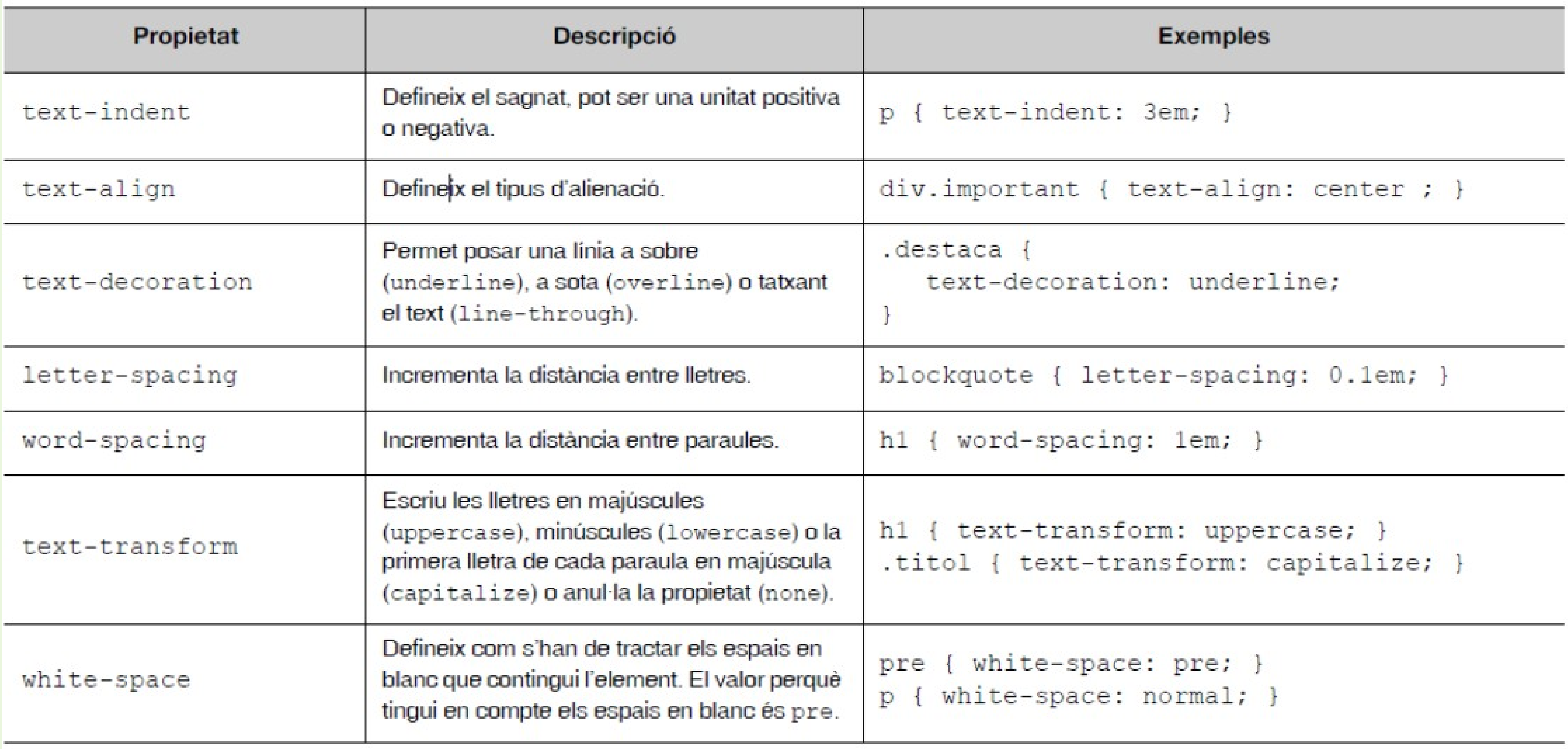
font	Propietat que resumeix les propietats anteriors.	p { font: 12px sans-serif }
		p { font: x-large Helvetica, serif }
		p { font: bold italic large serif }
		p { font: normal small-caps 120% fantasy }

line-height	Estableix l'alçada de cadascuna de les línies de l'element.	p { line-height: 2em; }

Basic format properties

- Measures and colors
- Font types
- **Text aspect**
- Colors and backgrounds

Text aspect



Propietat	Descripció	Exemples
`text-indent`	Defineix el sagnat, pot ser una unitat positiva o negativa.	`p { text-indent: 3em; }`
`text-align`	Defineix el tipus d'alienació.	`div.important { text-align: center ; }`

`text-decoration`	Permet posar una línia a sobre (`underline`), a sota (`overline`) o tatxant el text (`line-through`).	`.destaca {`
		`   text-decoration: underline;`
		`}`

`letter-spacing`	Incrementa la distància entre lletres.	`blockquote { letter-spacing: 0.1em; }`
`word-spacing`	Incrementa la distància entre paraules.	`h1 { word-spacing: 1em; }`

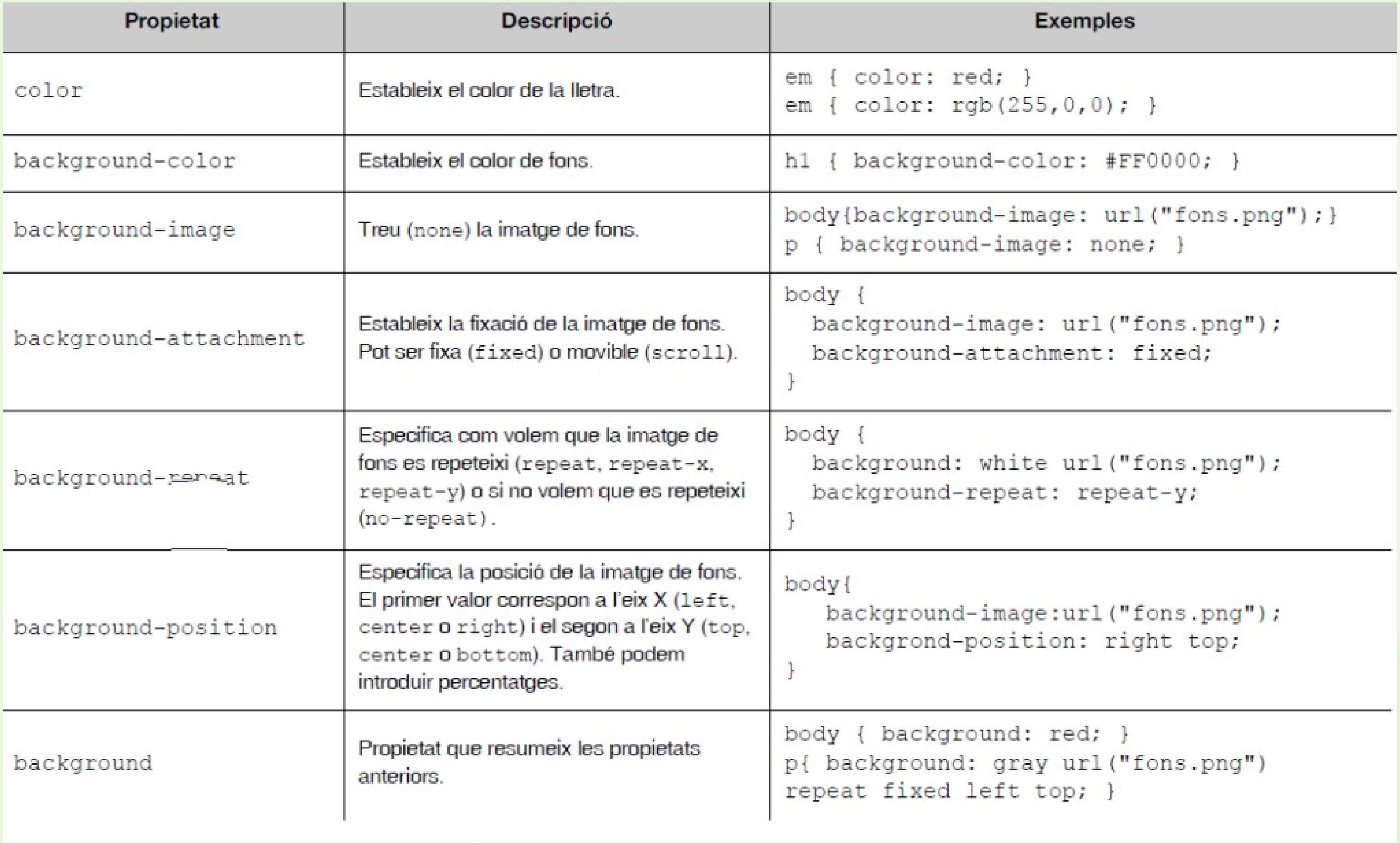
`text-transform`	Escriu les lletres en majúscules (`uppercase`), minúscules (`lowercase`) o la primera lletra de cada paraula en majúscula (`capitalize`) o anul·la la propietat (`none`).	`h1 { text-transform: uppercase; }`
		`.titol { text-transform: capitalize; }`

`white-space`	Defineix com s'han de tractar els espais en blanc que contingui l'element. El valor perquè tingui en compte els espais en blanc és `pre`.	`pre { white-space: pre; }`
		`p { white-space: normal; }`

Basic format properties

- Measures and colors
- Font types
- **Text aspect**
- Colors and backgrounds

Colors & fonts



Propietat	Descripció	Exemples

color	Estableix el color de la lletra.	em { color: red; }
		em { color: rgb(255,0,0); }

background-color	Estableix el color de fons.	h1 { background-color: #FF0000; }

background-image	Treu (none) la imatge de fons.	body{background-image: url("fons.png");}
		p { background-image: none; }

background-attachment	Estableix la fixació de la imatge de fons. Pot ser fixa (fixed) o movible (scroll).	body {
		  background-image: url("fons.png");
		  background-attachment: fixed;
		}

background-repeat	Especifica com volem que la imatge de fons es repeteixi (repeat, repeat-x, repeat-y) o si no volem que es repeteixi (no-repeat).	body {
		  background: white url("fons.png");
		  background-repeat: repeat-y;
		}

background-position	Especifica la posició de la imatge de fons. El primer valor correspon a l'eix X (left, center o right) i el segon a l'eix Y (top, center o bottom). També podem introduir percentatges.	body{
		   background-image:url("fons.png");
		   backgrond-position: right top;
		}

background	Propietat que resumeix les propietats anteriors.	body { background: red; } p{ background: gray url("fons.png") repeat fixed left top; }

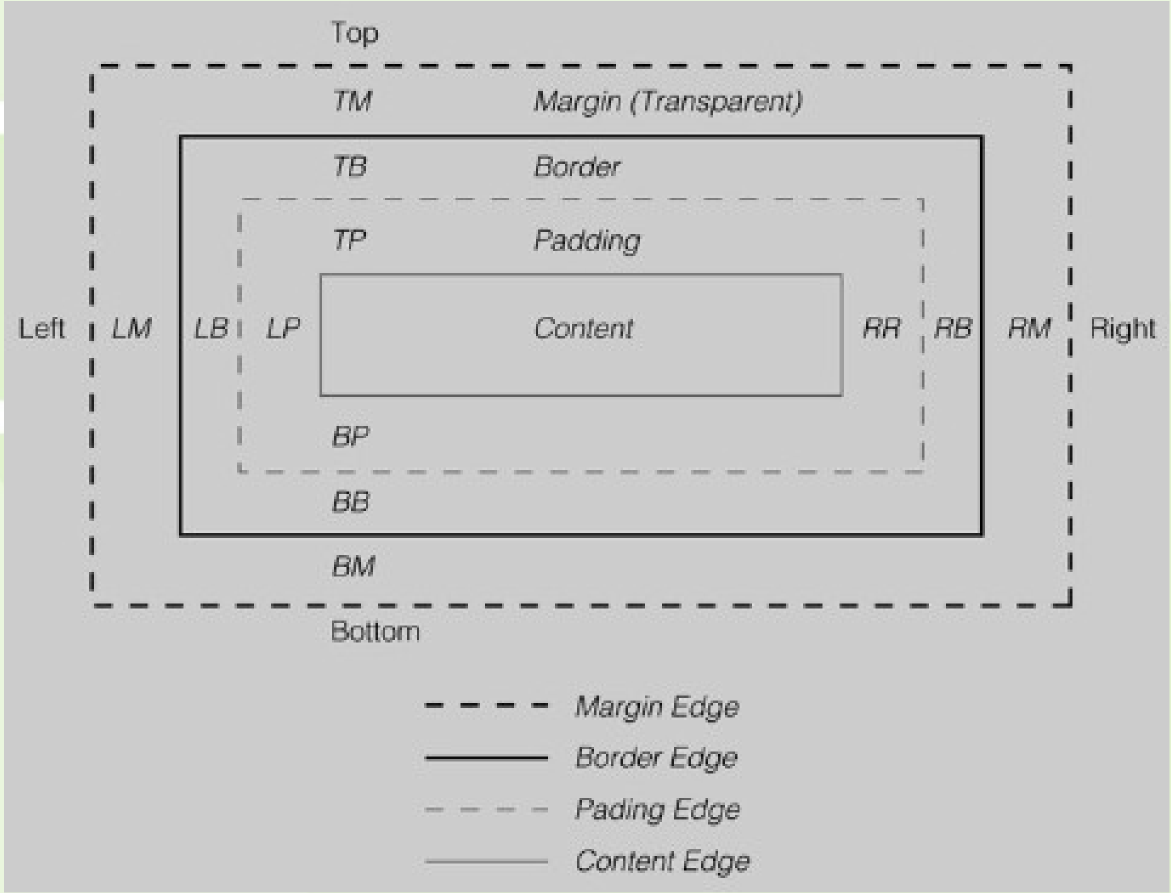
The box model

- To understand this model we have to think in **each element** of the (X)HTML document **as** if it were **a box**.

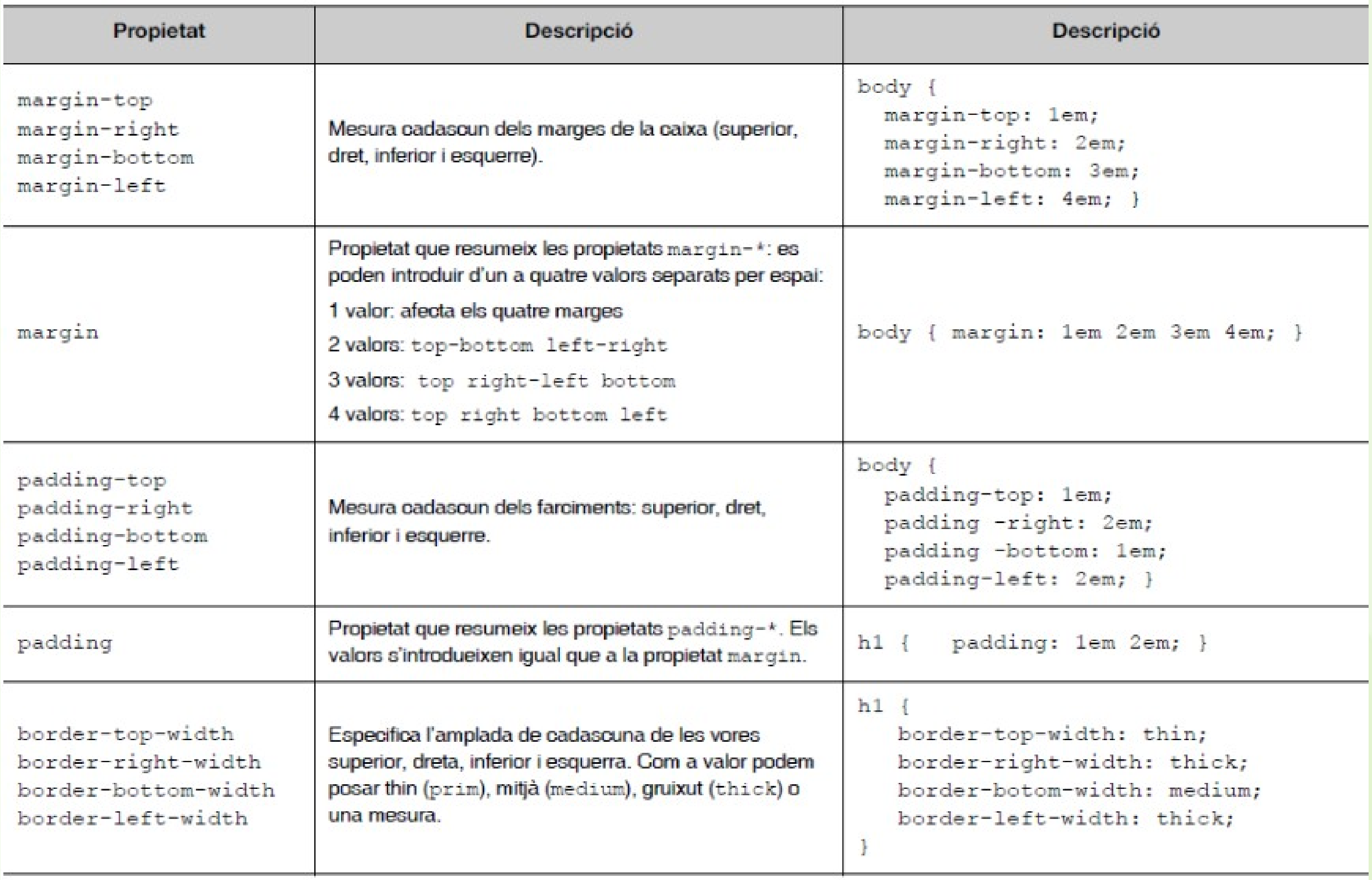
- From each of these boxes we can distinguish the following elements:
  - The **content of the item**: text or other items.
  - The **padding**: space between the contents and the edge of the box.
  - The **border** (border): border that delimits the box.
  - The **margin**: space between the edge of the box and the rest of the elements.

The box model

- In addition, each box has **four segments**: the top (bottom), the bottom (bottom), the left (left) and the right (right).



The box model properties



Propietat	Descripció	Descripció

`margin-top`
`margin-right`
`margin-bottom`
`margin-left`
	Mesura cadascun dels marges de la caixa (superior, dret, inferior i esquerre).	`body {`
		`  margin-top: 1em;`
		`  margin-right: 2em;`
		`  margin-bottom: 3em;`
		`  margin-left: 4em; }`

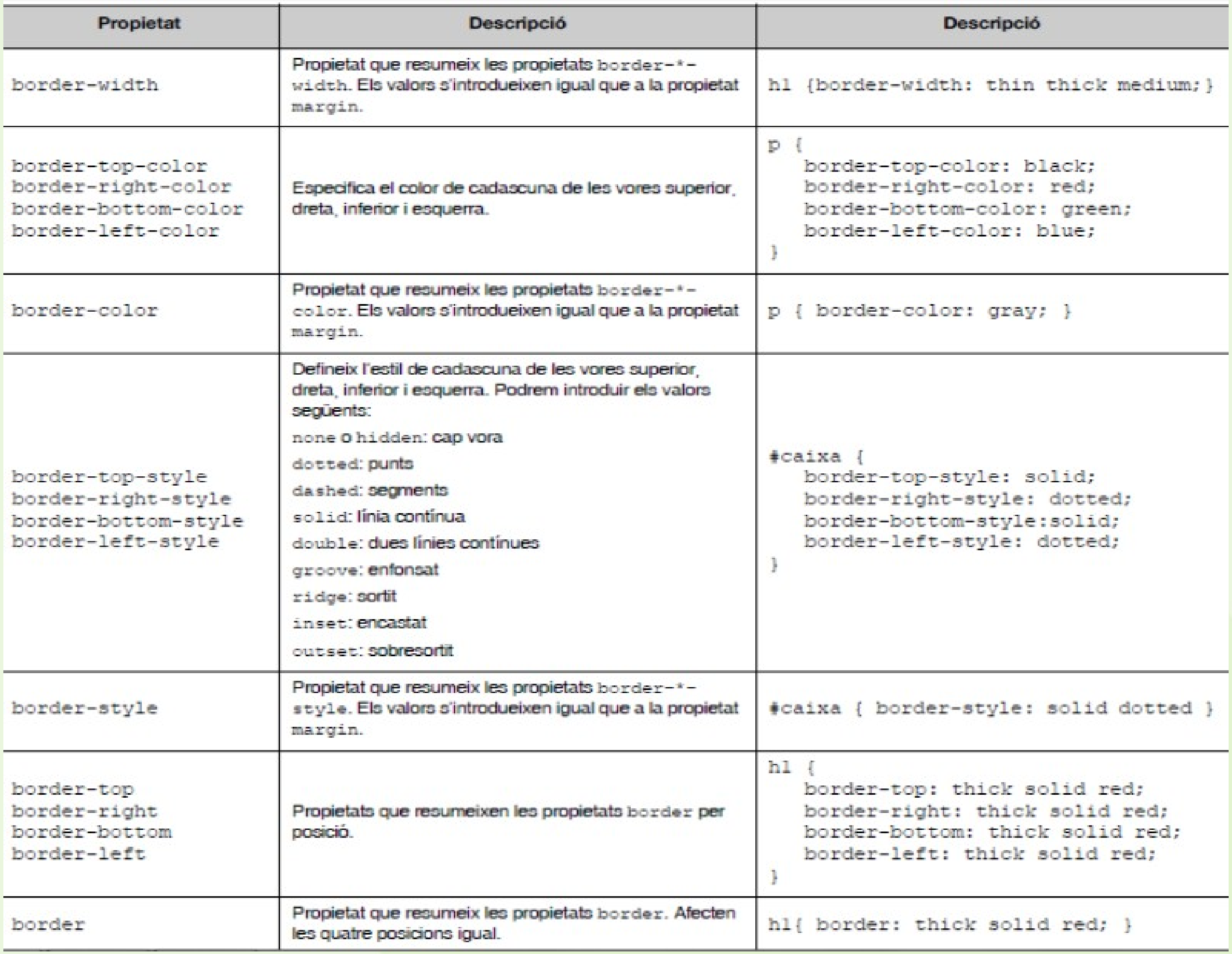
`margin`	Propietat que resumeix les propietats `margin-*`: es poden introduir d'un a quatre valors separats per espai: 1 valor: afecta els quatre marges 2 valors: `top-bottom left-right` 3 valors: `top right-left bottom` 4 valors: `top right bottom left`	`body { margin: 1em 2em 3em 4em; }`

`padding-top`
`padding-right`
`padding-bottom`
`padding-left`
	Mesura cadascun dels farciments: superior, dret, inferior i esquerre.	`body {`
		`  padding-top: 1em;`
		`  padding -right: 2em;`
		`  padding -bottom: 1em;`
		`  padding-left: 2em; }`

`padding`	Propietat que resumeix les propietats `padding-*`. Els valors s'introdueixen igual que a la propietat `margin`.	`h1 {   padding: 1em 2em; }`

`border-top-width`
`border-right-width`
`border-bottom-width`
`border-left-width`
	Especifica l'amplada de cadascuna de les vores superior, dreta, inferior i esquerra. Com a valor podem posar thin (`prim`), mitjà (`medium`), gruixut (`thick`) o una mesura.	`h1 {`
		`   border-top-width: thin;`
		`   border-right-width: thick;`
		`   border-botom-width: medium;`
		`   border-left-width: thick;`
		`}`

The box model properties



Propietat	Descripció	Descripció
border-width	Propietat que resumeix les propietats border-*-width. Els valors s'introdueixen igual que a la propietat margin.	h1 {border-width: thin thick medium;}

border-top-color
border-right-color
border-bottom-color
border-left-color
	Especifica el color de cadascuna de les vores superior, dreta, inferior i esquerra.	p {
		border-top-color: black;
		border-right-color: red;
		border-bottom-color: green;
		border-left-color: blue;
		}

border-color	Propietat que resumeix les propietats border-*-color. Els valors s'introdueixen igual que a la propietat margin.	p { border-color: gray; }

border-top-style
border-right-style
border-bottom-style
border-left-style
	Defineix l'estil de cadascuna de les vores superior, dreta, inferior i esquerra. Podrem introduir els valors següents:
	none o hidden: cap vora
	dotted: punts
	dashed: segments
	solid: línia contínua
	double: dues línies contínues
	groove: enfonsat
	ridge: sortit
	inset: encastat
	outset: sobresortit
		#caixa {
		border-top-style: solid;
		border-right-style: dotted;
		border-bottom-style:solid;
		border-left-style: dotted;
		}

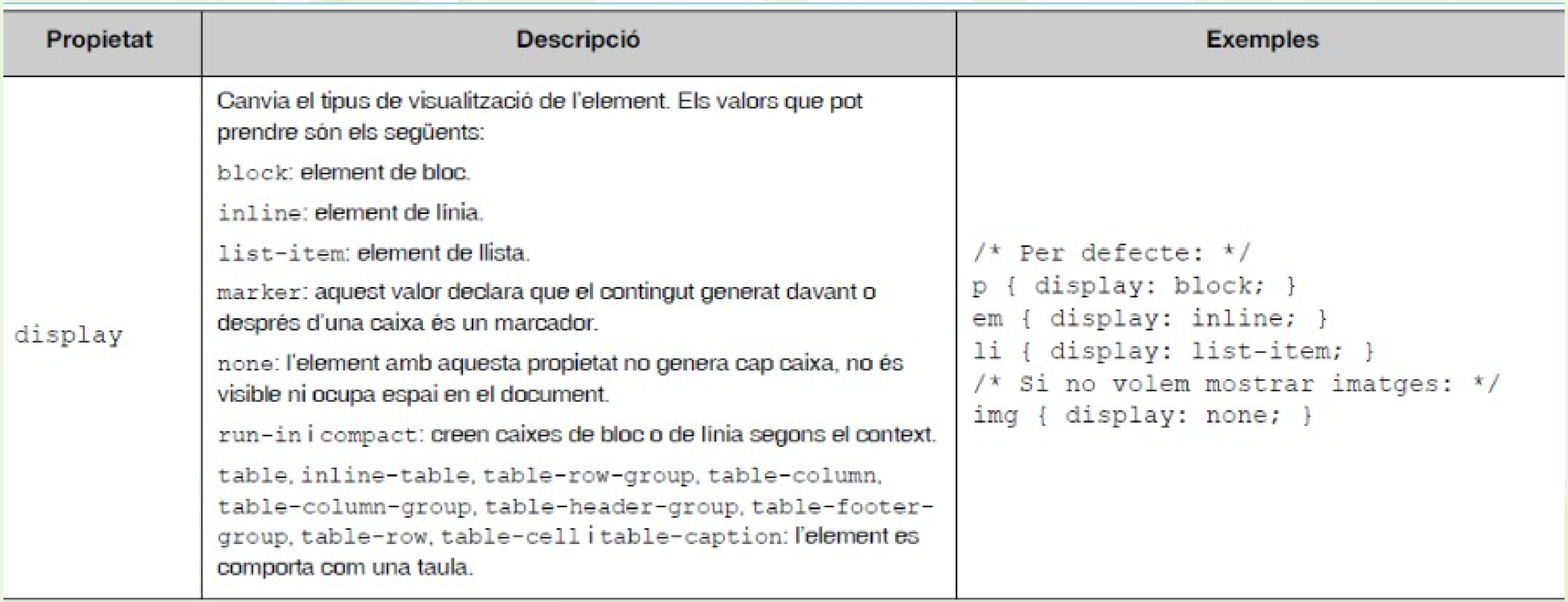
border-style	Propietat que resumeix les propietats border-*-style. Els valors s'introdueixen igual que a la propietat margin.	#caixa { border-style: solid dotted }

border-top
border-right
border-bottom
border-left
	Propietats que resumeixen les propietats border per posició.	h1 {
		border-top: thick solid red;
		border-right: thick solid red;
		border-bottom: thick solid red;
		border-left: thick solid red;
		}

border	Propietat que resumeix les propietats border. Afecten les quatre posicions igual.	h1{ border: thick solid red; }

Visual format of the elements

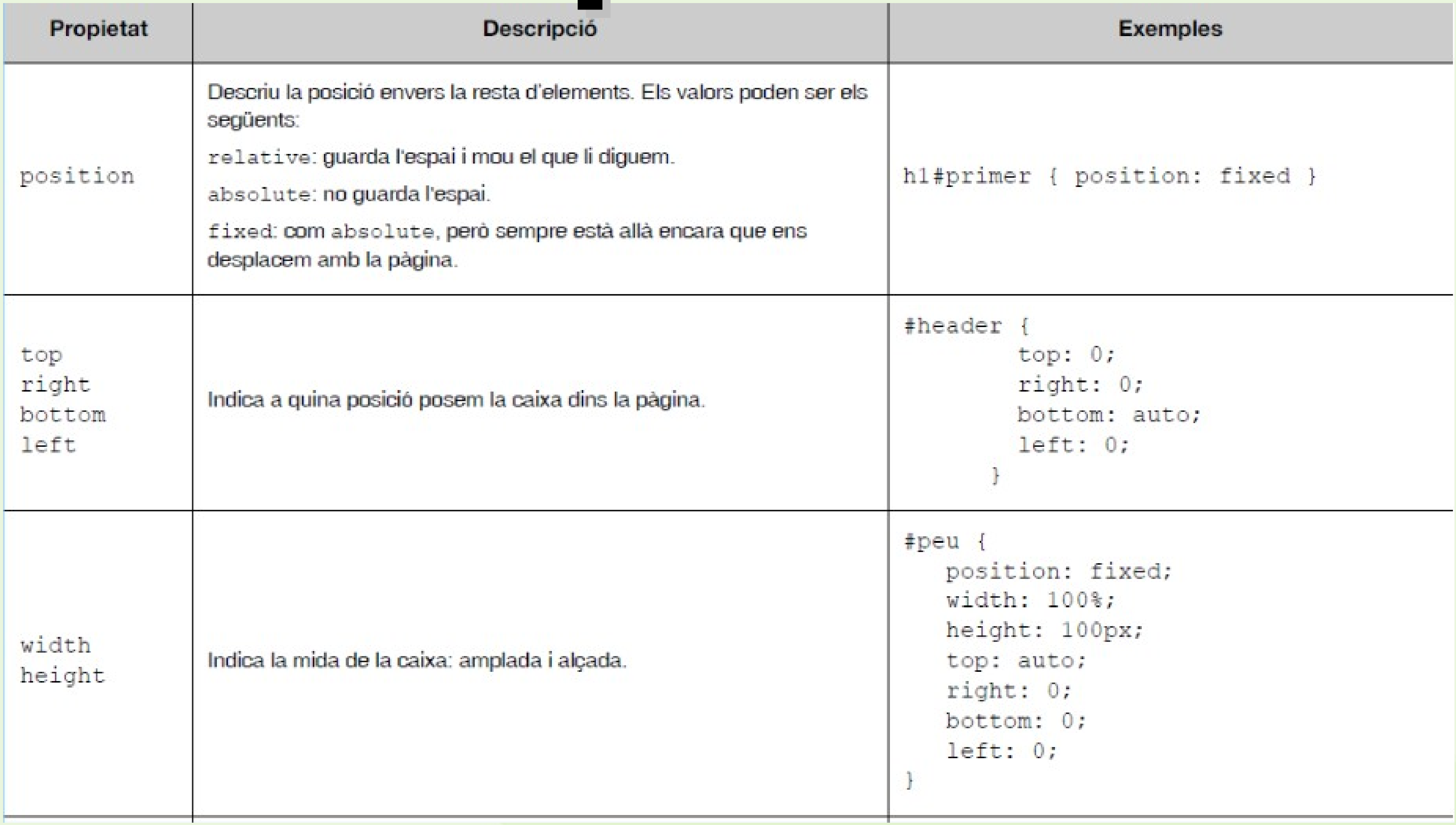
- Defines how the boxes are (that is, the (X)HTML elements seen as boxes) in the document and the behavior of these boxes relative to the other boxes on the page.



Propietat	Descripció	Exemples

display	Canvia el tipus de visualització de l'element. Els valors que pot prendre són els següents:
	block: element de bloc.
	inline: element de línia.
	list-item: element de llista.
	marker: aquest valor declara que el contingut generat davant o després d'una caixa és un marcador.
	none: l'element amb aquesta propietat no genera cap caixa, no és visible ni ocupa espai en el document.
	run-in i compact: creen caixes de bloc o de línia segons el context.
	table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell i table-caption: l'element es comporta com una taula.
		/* Per defecte: */
		p { display: block; }
		em { display: inline; }
		li { display: list-item; }
		/* Si no volem mostrar imatges: */
		img { display: none; }

Properties

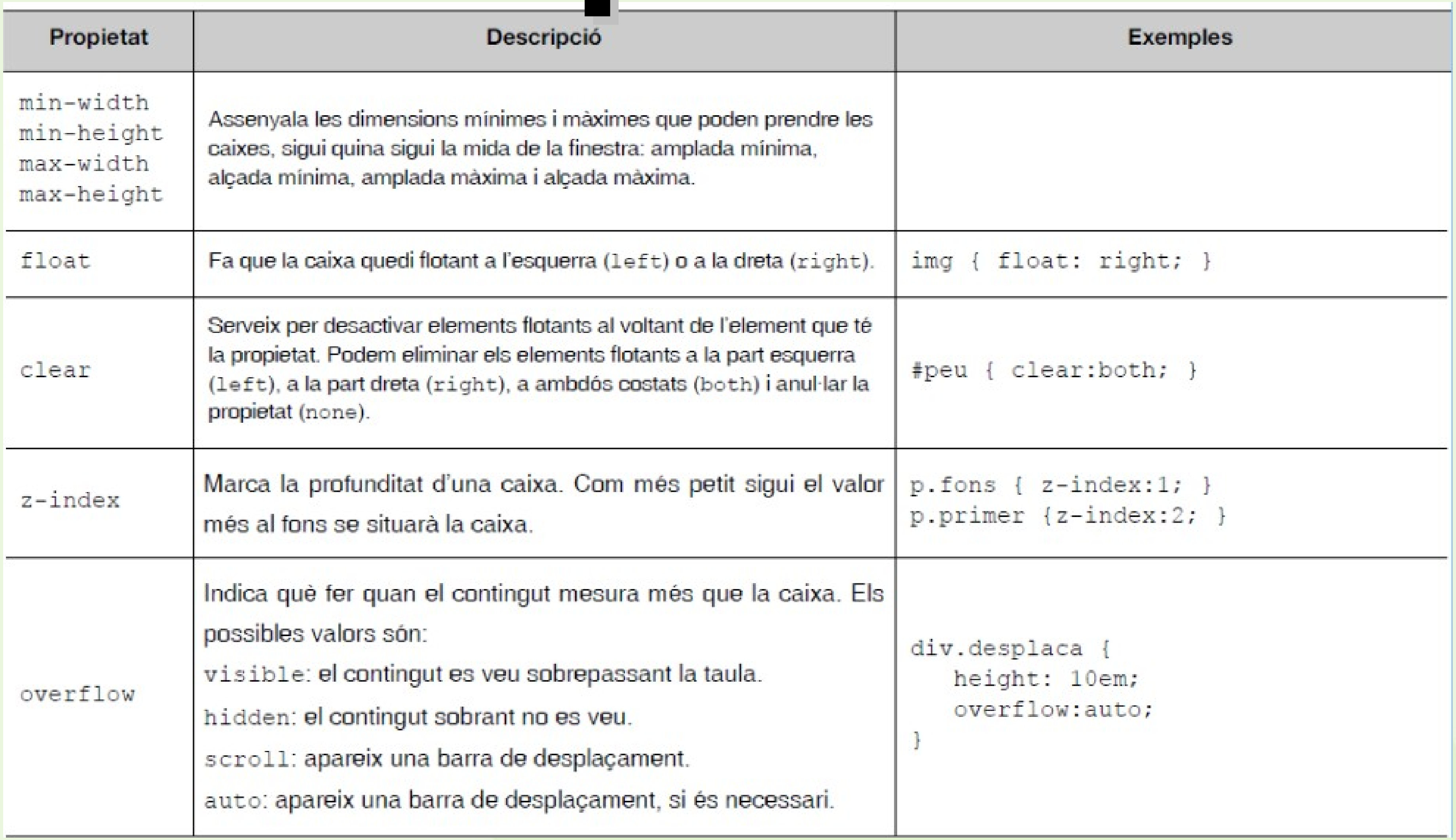


Propietat	Descripció	Exemples
position	Descriu la posició envers la resta d'elements. Els valors poden ser els següents: relative: guarda l'espai i mou el que li diguem. absolute: no guarda l'espai. fixed: com absolute, però sempre està allà encara que ens desplacem amb la pàgina.	h1#primer { position: fixed }

top
right
bottom
left
	Indica a quina posició posem la caixa dins la pàgina.	#header {
		top: 0;
		right: 0;
		bottom: auto;
		left: 0;
		}

width
height
	Indica la mida de la caixa: amplada i alçada.	#peu {
		position: fixed;
		width: 100%;
		height: 100px;
		top: auto;
		right: 0;
		bottom: 0;
		left: 0;
		}

Properties



Propietat	Descripció	Exemples

`min-width`
`min-height`
`max-width`
`max-height`
	Assenyala les dimensions mínimes i màximes que poden prendre les caixes, sigui quina sigui la mida de la finestra: amplada mínima, alçada mínima, amplada màxima i alçada màxima.	

`float`	Fa que la caixa quedi flotant a l'esquerra (`left`) o a la dreta (`right`).	`img { float: right; }`
`clear`	Serveix per desactivar elements flotants al voltant de l'element que té la propietat. Podem eliminar els elements flotants a la part esquerra (`left`), a la part dreta (`right`), a ambdós costats (`both`) i anul·lar la propietat (`none`).	`#peu { clear:both; }`

`z-index`	Marca la profunditat d'una caixa. Com més petit sigui el valor més al fons se situarà la caixa.	`p.fons { z-index:1; }`
		`p.primer {z-index:2; }`

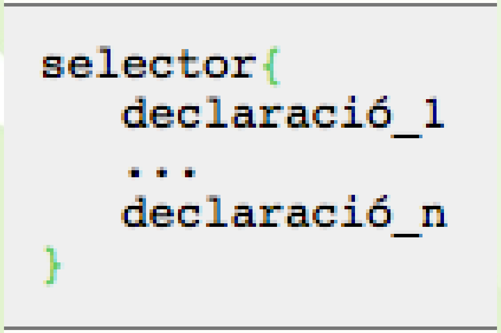
`overflow`	Indica què fer quan el contingut mesura més que la caixa. Els possibles valors són:
	`visible`: el contingut es veu sobrepassant la taula.
	`hidden`: el contingut sobrant no es veu.
	`scroll`: apareix una barra de desplaçament.
	`auto`: apareix una barra de desplaçament, si és necessari.
		`div.desplaca {`
		`   height: 10em;`
		`   overflow:auto;`
		`}`

Selectors

A selector is the part of the CSS rule that identifies the element or elements of the (X)HTML document whose properties we want to specify.

There are different types:

- Of type
- Of classes
- Identifier
- Attributes
- Universal
- Of children
- Of adjacent brothers



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

Type selector

- The item is selected by the (X)HTML tag name. The properties that are defined affect all (X)HTML elements with that tag name.

```
h1{
      color:blue;
}
```

Class selector

- They are used to associate properties to all elements that have the class attribute of equal value:

```
<tag class="ValueClass">...</tag>

tag.ValueClass {                      .ValueClass {
    declaracions                          declaracions
}                                    }
            ------------------------------------------
p{                                 p.destacat {
color:green;                           color:red;
}                                  }
```

Identifier selector

- They are used to associate properties with the element that has a certain identifier.
- The identifier of an (X)HTML element is declared with the ***id*** attribute of the element.
- This *identifier must be unique*.

```
<tag id="ValueId">...</tag>
```

```
tag#ValueId {
    declaracions
}
```

```
#ValueId {
    declaracions
}
```

Consorci d'Educació de Barcelona
Generalitat de Catalunya
Ajuntament de Barcelona

Attribute selector

- They will be used to apply the properties of the rule to elements that have a specific attribute.

```
<tag attr="valor">...</tag>

tag[attr] {
    declaracions
}
```

Example:

```
img[title] {
    margin-right:10px;
}
```

Attribute selector

- For more than one attribute:

```
img[title][class="preferent"] {
    margin-right:10px;
}
```

- For the attribute value:
- **[**attribute**~=**"value"**]:** is used to select items that have a *list of words separated by spaces* as an attribute, one of which is exactly value.

- **[**attribute**|=**"value"**]:** is used to select items whose attribute is a *list of words separated by hyphens*, starting with ‘value’.

Exemples:

```
img[alt~="logo"] {
    border: solid
}


p[lang|="en"] {
    font-family: "Times New Roman", Serif
}
```

Universal selector

Affects all items in the document:

```
* {
    declarations
}
```

Descendant selector

It allows us to select all the items that are contained in another.

```
Tag TagDescendant {
    declaracions
}
```

Example:

```
p em{
    color:blue;
}
```

Child selector

- One element is the child of another if it is a **<u>first-level descendant</u>**.
- Child selectors allow us to select children for a particular element.
- The symbol we need to use is **>**.

```
Tag>ChildTag
{
    declarations
}
```

Child selector

- Examples:
  - body**>**p { font-weight: bold; }
  - div ol**>**li p { text-decoration : underline; }

- In the second case, the selector affects all **<p>** elements that are descended from an element **<li>**, where the element **<li>** must be a child (direct descendent) of an element **<ol>** which must be descended from a element **<div>**.

Adjacent sibling selector

- They allow us, given an element, to confer properties on his brother immediately afterwards.

- The symbol used is **+**:

```
Tag + TagAdjacentSibling {
   declarations
}
```

Example:

- If we want to reduce the vertical space between a **<h1>** and a **<h2>** that follows it immediately, we must define the following rule:

```
h1 + h2 { margin-top: -5mm }
```

Grouping of selectors

- If several selectors have the same properties we can group them separating the selectors by a comma:

```
selector1, selector2, selector3 {
    declaracions
}
```

- Example:

```
h1, h2, h3 { font-family: sans-serif }
```

Exercise 2

Take a the previous resume page **created by you**, and add in a new css file all the modifications that you consider to give it a professional look.

Use **different selectors** to format the contents of the items, and organize them within the file, clearly separating them with comments. Also use the universal selector.

Pseudoclasses & pseudoelements

- They serve to add some additional effects to the items that interest us.

Pseudoclasses

- **:first-child.** Select an item that is the first child of another item.
- **:link.** Lets you define the style of the links on our page, when they haven't been visited yet.
- **:visited.** Lets you define the style of the links on our page, when they have been visited.
- **:active.** Lets you define the style of the items when they are activated (when we click on them).
- **:hover.** Lets you define the style of the elements as we go over them.
- **:focus.** Define the style of items when they receive focus.

Pseudoelements

- **:first-line.** Define the style of the first line of the item.
- **:first-letter.** Lets you add style properties to the first letter of the element.
- **:after.** Allows you to enter content at the end of the item. We must give the element the property happy with the desired value.
- **:before.** Allows you to enter content at the beginning of the item. We must give the element the property happy with the desired value.

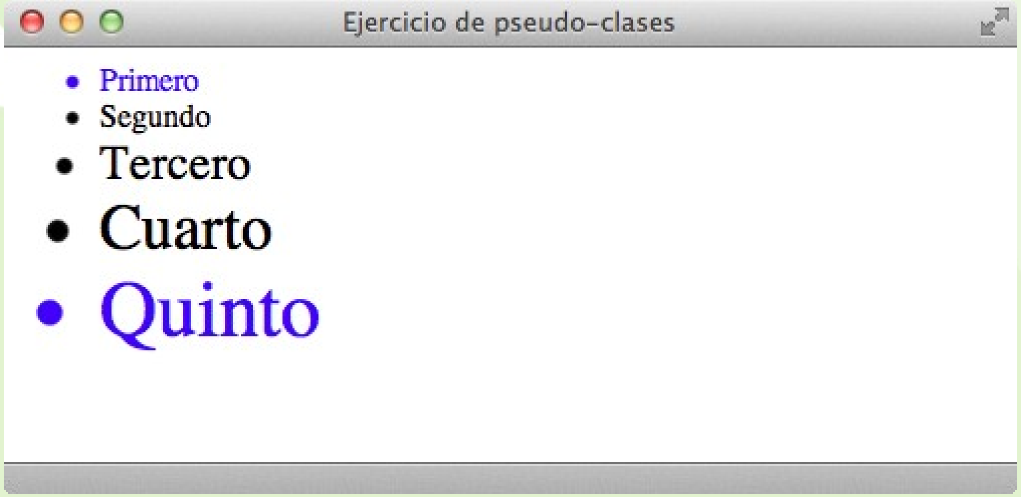
Exercici 3

Given the HTML code, and using the pseudo-classes first-child, first-of-type, nth-of-type, nth-last-of-type, last-of-type i last-child try to obtain the following result:

```
<html>
<head>
   <title>Ejercicio de pseudo-clases</title>
   <style type="text/css">

   </style>
</head>
<body>
   <ul>
      <li>Primero</li>
      <li>Segundo</li>
      <li>Tercero</li>
      <li>Cuarto</li>
      <li>Quinto</li>
   </ul>
</body>
</html>
```

Clue → w3schools



- Primero
- Segundo
- Tercero
- Cuarto
- Quinto