**Cascading Style Sheets**

<http://www.csszengarden.com/> http://cssreference.io/  
<https://www.w3.org/TR/CSS22/> <https://www.w3.org/TR/CSS/#css>  
<https://medium.com/@ckor/make-full-screen-sections-with-1-line-of-css-b82227c75cbd>  
normalize.css is the style sheet used to reset the default properties of each user agent. This file should be linked before my own style sheet. <https://necolas.github.io/normalize.css/>

The correct HTML for referring to an external CSS is the <link> element within the <head> section <link rel=”stylesheet” type=”text/css” href=”css/myStyleSheet.css”>

**<link>** tag use to include four attributes when matching a CSS file:

* **rel** =”**stylesheet**”
* **type** =”**text/css**”
* **href**: The URL can be relative or absolute pointing to an external or internal resource.
* **media**: indicates the way to apply the styles from the CSS file

|  |  |
| --- | --- |
| Media =” ” | Descripción |
| all | All devices |
| braille | Touch devices using braille system |
| embosed | Braille printers |
| handled | Mobile, PDA, etc |
| print | Printers and browsers with in “preliminary view” mode |
| proyection | Slide |
| screen | Computer desktop screens |
| speech | Voice navigator devices for blind people |
| tty | Textual devices |
| tv | Televisions and low resolution devices |

Other option is using the <style> tag in the heading as follows:

**<style** **type**=**"text/css" media**=**"screen">**

**@import '/css/estilos.css'; </style>**

Rules @import precede any other rules except @charset

To include css styles in the HTML page it is posible to use the special rule @import. @import rules have to be declared before any other rule except @charset rule. The URL of the CSS file is declared by a text string inside of round brackets and with the reserved word **url()**. The following @import rules are equivalent:

@import '/css/estilos.css';

@import "/css/estilos.css";

@import url('/css/estilos.css');

@import url("/css/estilos.css");

@media is a special css rule that specify the media in which the css styles will be applied

**@media screen, print { body {line-height: 1.2} }**

# CSS Rules

Selector {property: value;}

## Selectors

Many selectors separate by a space represent a descendent selector; the rule will be applied on the last selector and the previous selectors indicate the nesting position.

Universal Selector **\*** In order to group selectors, separate each selector with a comma.

Adjacent Sibling Selector (element1 **+** element2)

* Both elements have to be siblings (same father/ same nesting level)
* Element2 has to be defined immediately after the element1

General Sibling Selector A **~** B

This is like the adjacent selector (A+B) except it gets all of the following elements instead of one.

* **A~B** select all sibling B elements that follow A.

Attribute Selectors

**[specificAttribute]**, selects all elements with the specific Attribute regardless its value.

* **a[href] 🡪** selects all a elements that have a **href="anything"** attribute.
* **[type] 🡪** selects all elements that have a **type="anything".** attribute

**[specificAttribute=Value]**, selects all elements with the specific Attribute and the specific value.

* **input[type="checkbox"] 🡪** selects all checkbox input elements.

**[attribute^="val"]**; All elements with an attribute value that starts with specific characters

* **.toy[category^="Swim"] 🡪** selects elements with class=”**toy**” and either **category="Swimwear or category="Swimming"…**

**[attribute$="lue"]**; All elements with an attribute value that ends with specific characters

* **img[src$=".jpg"] 🡪** selects all images display a **.jpg** image.

**[attribute\*="alu"]**; Elements with an attribute value that contains specific characters anywhere

* **img[src\*="/thumbnails/"] 🡪** selects all image elements that show images from the "thumbnails" folder.
* **[class\*="heading"]** 🡪 selects all elements with "heading" in their class, like class="main-heading" and class="sub-heading"

**[specificAttribute~=Value]**, selects all elements with the specific Attribute and at least one of the values of this specific attribute is the specific attribute.

First Child Pseudo-Selector

A child element is an element that is directly nested in another element. You can combine this pseudo selector with other selectors:

* **:first-child** **🡪** selects all first child elements
* **p :first-child 🡪** selects all first child elements which are inside <p> elements
* **div p:first-child 🡪** selects all first child <p> elements that are in a div

Only Child Pseudo-Selector **:only-child**

Select elements that are the only element inside of another one

* **:only-child 🡪** selects all only child elements
* **p:only-child 🡪** selects all only child <p> elements
* **div p:only-child 🡪** selects all only child <p> elements which are inside a div

Last Child Pseudo-Selector **:last-child**

Select elements that are the last element inside of another one

Nth Child Pseudo-Selector **:nth-child(xth)**

Select an element by its order in another element. In order to select the element A it is necessary to type **A:nth-child(xth)** and its position taking into account all the siblings of the element, even if they are not the same element.

* **:nth-last-child** **🡪** selects all child elements that are in the third position counting from the back
* **p:nth-last-child** **🡪** selects all child <p> elements that are in the second position counting from the back

Nth Last Child Pseudo-Selector **:nth-last-child(xth)**

Select an element by its order in another element, counting from back! In order to select the element A it is necessary to type **A:nth-child(xth)** and its position takes into account all the siblings of the element, even if they are not the same element, and counting from the back

* **:nth-last-child** **🡪** selects all child elements that are in the third position counting from the back
* **p:nth-last-child** **🡪** selects all child <p> elements that are in the second position counting from the back

First of type Selector

* **span:first-of-type 🡪** selects all first <span> child within another element

Nth of type Selector

Selects a specific element based on its type and order in another element - or even or odd instances of that element.

* **a:nth-of-type(even) 🡪** selects all odd instances of the <a> type

Nth-of-type Selector with Formula

The nth-of-type formula selects every nth element, starting the count at a specific instance of that element.

* **span:nth-of-type(6n+2) 🡪** selects every 6th instance of a <span>, starting from (and including) the second instance. It selects two elements: the 2nd and the 8th

Only of type Selector

Select elements that are the only ones of their type within of their parent element

* **p span:only-of-type** **🡪** selects a <span> within any <p> if it is the only span in there

Last of type Selector

Selects each last element of that type within another element. Remember type refers the kind of tag, so <p> and <span> are different types.

* **div:last-of-type 🡪** selects the last <div> in every element.
* **p span:last-of-type 🡪** selects the last <span> in every <p>

Empty Selector

Selects elements that don't have any other elements inside of them.

* **div:empty 🡪** selects all empty <div> elements.

Negation Pseudo-Class **:not(X)**

Select all elements that don't match the negation selector

* **:not(#fancy) 🡪** selects all elements that do not have **id="fancy"**
* **div:not(:first-child) 🡪** selects every div that is not a first child.
* **:not(.big, .medium) 🡪** selects all elements that do not have **class="big"** or **class="medium"**

**<https://www.w3schools.com/css/css_pseudo_classes.asp>**

In case of property conflicts, CSS apply the following procedure:

* 1. Determinate all rules that applies to the element for the selected CSS media
  2. Order the rules according to its origin (browser, user, designer) and priority (key word **!important**)
  3. Order the rules according to the specificity of the selector. The more generic, less importance
  4. If after appliying this three steps still there are two or more rules with the same priority, the last defined will prevail
* The more specific selector, the more importance for the rule
* In case of similar specificity, the last rule will be applied.

At css, the browser applies the rules from the general to specific. It means that the browser applies in first place those rules which selector is an html tag, and secondly specific selectors with a defined class in html. The class is used to restrict rules.

# Relative measurement units

The great advantage of the relative units is that they always maintain the design proportions of the page.

* **1em 🡪** always refers to the font size of the element. By default, browsers render the text <p> with a 16px size. If the this is changed buy the font-size property, 1em will be the equivalent to the new size value
* **1ex 🡪** its value is approximately the half of the em.
* **px** 🡪 It is also a relative unit as it depends on the resolution of the device on which the page is displayed.

Computer desktop screens with 15 inches (38cm) with HD resolution are: 1280x800 px. However, most screens are full HD, which means 1920x1080 px

|  |  |
| --- | --- |
| Device | Resolution (px) |
| Computer Desktop Screens | 1920 x 1080 |
| Laptop | 1024px (width) |
| Tablet | >576px |
| Mobile | >320px |

To apply relative measurement units it is necessary to set a text base with the rule font-size property into the body (selector). Then, all elements nested in the body will inherit the value of the property unless it would be overwritten by another rule. It is important to note that the relative value is not inherited, but the absolute value.

[**http://librosweb.es/libro/css/capitulo\_3.html**](http://librosweb.es/libro/css/capitulo_3.html)

[**https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Your\_first\_HTML\_form**](https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Your_first_HTML_form)

[**https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Styling\_HTML\_forms**](https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Styling_HTML_forms)

# Responsive Design - Elastic Example

**<!DOCTYPE html>**

**<html>**

**<head>**

**<style>**

**body { font-size:1.0em; }**

**div.contenedor { font-size: 1.25em; }**

**p.letra-pequeña { font-size: 0.9em; }**

**@media only screen and (max-width:500px) {**

**/\* For mobile phones: \*/**

**body { font-size:140%; }**

**}**

**@media only screen and (max-width:800px) {**

**/\* For tablets: \*/**

**body { font-size:120%; }**

**}**

**</style>**

**</head>**

**<body>**

**<div class="contenedor">**

**<p>.contenedor tiene un tamaño de 16px \* 1.25 = 20px.</p>**

**<p class="letra-pequeña">.letra-pequeña tiene un tamaño de 20px \* 0.9 = 18px. Toma 20px porque el em toma el último valor definido en alguno de sus contenedores.</p>**

**</div>**

**<p class="letra-pequeña">.letra-pequeña, ahora que no está dentro de .contenedor, tiene un tamaño de 16px \* 0.9 = 14.4px. Toma 16px porque no se ha definido un font-size en alguno de los contenedores, por lo que toma el valor del navegador.</p>**

**</body>**

**</html>**

# Secure Fonts

The typography extension files are: **.otf .ttf .woff .svg .eot**

Upload a file within any extension at [www.font2web.com](http://www.font2web.com) and it returns or gets back the typography at 5 file extensions and a css style sheet with the association **@font-face {**

**@font-face {**

**font-family: ‘Typography Name’;**   
**Src**: **url(’../fonts/name.extension);**  
**font-style:** **normal** (bold, italic…)**;**   
 **font-weight:** thin (300) | **normal** (500) | bold (700) | ultra-bold (900)**;**

**}**

**../** means to go up one level in the folder directory structure.

[www.dafont.com](http://www.dafont.com)

[www.fontsquirrel.com](http://www.fontsquirrel.com)

[www.icomoon.io](http://www.icomoon.io) Icomoon APP

[www.fontawesome.io](http://www.fontawesome.io)

Ilustrator 🡪 Import icons 🡪 Convert to **.svg**

Click me 🡪 Generate font 🡪 Download

* Put the files in the fonts folder and copy the code at the style sheet .css

Now it is possible to add icons as vectorial typography from the HTML file:

**<span class=”icon”></span> [ icon 🡪 “fa-img” ]**

ACCESSIBILITY 🡪 **Wai aria**

**Selector {**

**aria-hidden: “True”;**  
**aria-label: “ ”;**  
**aria-label-by: “ ”;**

**}**

# Colors

A color can be specified as an RGB value, using this formula: ***rgb(red, green, blue)***. Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255. Shades of gray are often defined using equal values for all the 3 light sources.

The values of these three parameters can be typed in decimal, percentage or hexadecimal: ***#rrggbb*** where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

A color can be specified using hue, saturation, and lightness (HSL) in the form:  
***hsl(hue, saturation, lightness)***. Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue. Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color. Lightness is also a percentage, 0% is black, 50% is either light or dark, and 100% is white.

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all). The α channel specifies the opacity for a color: ***rgbα(red, green, blue, alpha***) or ***hslα( , , , )***

# Float positioning

The components of the container and its visualization order from the user point view are:

* **Content** 🡪 the HTML content of the element
* **Padding 🡪** optional empty space between the content and the border
* **Border 🡪** line that encloses the content and its padding
* **Background image 🡪** image displayed behind the content and the padding space
* **Background color 🡪** color displayed behind the background image
* **Margin 🡪** optional espace between the the container box and the rest of adjacent boxes

Margin and padding are transparent, so they show the image or color of their parent element. If a container box has defined a background image and a background color, the image will be shown as it has more priority. Combining transparent images and background colors it is possible to design interesting graphics.

<https://css-tricks.com/perfect-full-page-background-image/>

In that site there are some tips to set a background image on a website that covers the entire browser window at all times, filling the entire page with the image, retaining image proportions (aspect ratio) and with the image centered not causing scrollbars:

**Image#id {**

**background: url(image/bg.jpg) no-repeat center center fixed; *shorthand***

**background-repeat: no-repeat;**

**background-position: center center;**

**background attachment: fixed;**

**webkit-background-size: cover;**

**moz-background-size: cover;**

**o-background-size: cover;**

**background-size: cover;**

**}**

CSS defines five properties to set the background of each element (**background-color**, **background-image**, **background-repeat**, **background-attachment,** **background-position**) and the ***shorthand*** property **background**. <http://librosweb.es/libro/css/capitulo_4/fondos.html>

**margin, padding & border-width 🡪** This properties are expecting for 4 values in the order: top, right, bottom left. In case of entering 2 values: the first one is for top and bottom, and the second for the right and left. In case of entering only one value, it applies for all four positions.

**div { border-width: thin; }**  /\* thin thin thin thin \*/

**div { border-width: thin thick; }** /\* thin thick thin thick \*/

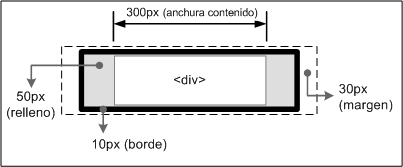
**div { border-width: thin thick medium; }** /\* thin thick medium thick \*/

Other properties are: **border-color border-style**

Expecting for four parameters, although it is always possible to specify only one border/margin/padding:

**border-left-style border-bottom-color**

***Shorthand*** properties are used to set several values simultaneously, for example: **border-top**

**div {**

**width: 300px;**

**padding-left: 50px;**

**padding-right: 50px;**

**margin-left: 30px;**

**margin-right: 30px;**

**border: 10px solid black;**

**}**

When two or more vertical margins are joined, they merge automatically and the height of the new margin will be equal to the height of the highest margin of those that have merged.

# Positioning

# Text

# Images

# Lists

# Tables

# Forms

# Others

# Useful resources

## Properties

**opacity 🡪** 0%=transparente. Si se usa como propiedad, se puede aplicar a todo lo que aparece dentro del selector CSS, pero también se le puede poner a un color: RGB**α** (cuarto canal).

**border 🡪** This property requires three values: width, style and color

**background-color 🡪** Is used to change the background color

**color 🡪** Is used to change the text color of an element

**list-style-type: square; 🡪** Makes a list that lists its items with squares

**font-size 🡪** This property controls the text size

**font-weight: bold; 🡪** Makes bold the text of the respective selector

**a {text-decoration: none;} 🡪** Is used to display links without an underline

**font & font-family 🡪** Both properties are valid to change the font of an element

<https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Styling_HTML_forms>

<https://developer.mozilla.org/en-US/docs/Glossary/CSS>

Hola a todxs.

He subido una buena cantidad de ejercicios de JS para que practiquéis. Os cuento cómo están estructurados (el siguiente orden es en el que considero que deberían hacerse):

  - Ejercicios\_JS\_Links: Se trata de ejercicios de terceros, el primer enlace es de iniciación, el segundo ya van subiendo la dificultad (los 9 primeros no deberían daros problemas, a partir del 10 la cosa se complica).

  - Ejercicios\_JS: Se trata de ejercicios de cosecha propia, alguno ya lo conoceréis o lo habréis hecho ya. Os rogaría que TODXS hicierais al menos los cuatro primeros (SIN jQuery ni otras librerías, solo con JS puro y duro), especialmente el del formulario que es el 4. En cuanto a 5, 6 y 7, estos ya son de una dificultad considerable, pero son especialmente idóneos para practicar el divide y vencerás.

  - Práctica\_JS: La joya de la corona. Una señora práctica de las que pongo en la Universidad. No recomiendo hacerla solx sino por parejas. Ya hablamos de palabras mayores, aunque si sabes hacer esto, ya puedes decir que sabes programar, sin lugar a dudas. En cualquier lenguaje.

Todos ellos están en Archivos > 1) Recursos Clase > 1.3 Ejercicios Básicos.

|  |  |
| --- | --- |
| block | inline |
| <div> | <span> |
| <h> | <img> |
| <p> | <video> |
| <form> | <input> |
| <ul> / <ol> , <li> | <a> |
| <table> , <tr> , <td> |  |

Inline elements are normally displayed without starting a new line. Los elementos HTML con salto de línea (retorno de carro) son un bloque. Los elementos de bloque llevan asociado un salto de línea (retorno de carro) y se aplican las mismas reglas CSS de espacio (containers) que a un div. Son elementos de bloque los que llevan asociada la propiedad *display: block* y no requieren ponerles un div.

* Los elementos de bloque ocupan lo que ocupe su padre (inherit). Un elemento de bloque va a ocupar todo lo que ocupa su padre, y, aunque le cambiemos el *width*, el elemento reducirá su tamaño pero el espacio reservado para él seguirá siendo el 100% de su padre.
* Los elementos de línea ocupan lo que ocupe su contenido (sus hijos). Por mucho que se le aplique un *width* a un elemento inline, éste va a ocupar lo que ocupe su contenido.
* *Display: inline* equivale exactamente al modelo de cajas flotantes (div float) siempre que los elementos tengan el mismo tamaño.

A cada elemento se le puede cambiar la propiedad *display* en CSS, excepto a los <div> que se subdividen en sub-bloques o sub-containers. *Display* es una propiedad que establece cómo debe mostrar el navegador a un contenido con respecto de sus hermanos pero cambiarle a un elemento su *display* por defecto no tiene mucho sentido. La propiedad *display: inline-block* evita el salto de línea pero a su vez permite la distribución block, se utiliza para los menús. También se emplea *display: none* para ocultar algo en un caso concreto.

El diseño adaptativo del tamaño se denomina “Responsive”. La altura de una web es la suma de las alturas de sus contenidos, el ancho es el .wrapper. Los elementos heredan la anchura de sus padres pero su altura será la de sus hijos, es decir el contenido. La anchura de un <tag> es la de su padre y la altura la de sus hijos.

CSS: El *width* de un div restringe a sus hijos, pero éste seguirá teniendo la anchura de su padre.

* Full: Hay un “.wrapper” después de cada: <header>, <main> and <footer>
* Box: Solamente hay un .wrapper englobando todo el contenido del <body>

La única caja que vamos a definir el ancho en píxeles es el .wrapper, porque su padre es el <body>, mientras que al resto de los *width* se les define en porcentaje%.

Las propiedades *text-align, margin y padding* se definen al contenedor, no al contenido. La propiedad *padding* es un margen DENTRO de la caja/contenedor. Si a *margin/padding* le damos 4 valores el orden será Norte-Este-Sur-Oeste; con dos valores el primero será para top y bottom y el segundo para right y left; un solo valor lo aplica por los cuatro lados.

## Properties

**opacity 🡪** 0%=transparente. Si se usa como propiedad, se puede aplicar a todo lo que aparece dentro del selector CSS, pero también se le puede poner a un color: RGB**α** (cuarto canal).

**border 🡪** This property requires three values: width, style and color

**background-color 🡪** Is used to change the background color

**color 🡪** Is used to change the text color of an element

**list-style-type: square; 🡪** Makes a list that lists its items with squares

**font-size 🡪** This property controls the text size

**font-weight: bold; 🡪** Makes bold the text of the respective selector

**a {text-decoration: none;} 🡪** Is used to display links without an underline

**font & font-family 🡪** Both properties are valid to change the font of an element

<https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Styling_HTML_forms>

<https://developer.mozilla.org/en-US/docs/Glossary/CSS>

<http://soyfrontend.com/>

Apuntes Guille

<https://docs.google.com/document/d/1aWVxwUwiDQhAMp7N3p1xbO4-kIJ_8cr6IwMOeZ7ST0k/edit?usp=sharing>