

Interactividad I

CSS



CSS

Qué es CSS

Es un lenguaje desarrollado para controlar la presentación de uno o varios documentos HTML. Las hojas de estilo CSS permiten separar la estructura y contenidos de su aspecto.

↳ Acrónimo de Cascading Style Sheets.

CSS

Qué es CSS

Las hojas de estilos CSS utilizan selectores para indicar a qué etiquetas HTML aplicarán las propiedades que definirán su posicionamiento, altura, anchura, fuente, color, bordes, fondo...

<https://uniwebsidad.com/libros/css>



Selector

Declaración



h1 { color: green; }



Propiedad

Valor

```
body { background: black; }
```

```
section { width: 940px; height: 480px; }
```

```
h2 { font-size: 36px; }
```

```
p { font-family: sans-serif; }
```

CSS

Cómo funciona

Los estilos CSS deben guardarse en un archivo independiente del contenido, pudiendo controlar tanto la presentación de una página, como la de un sitio web completo o incluso varios sitios.

↳ Este archivo debe llamarse *style.css* y se enlaza con nuestro código HTML usando la etiqueta `<link>`.

```
<!doctype html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Título de esta página</title>
```

```
    <link rel="stylesheet" href="style.css">
```

```
  </head>
```

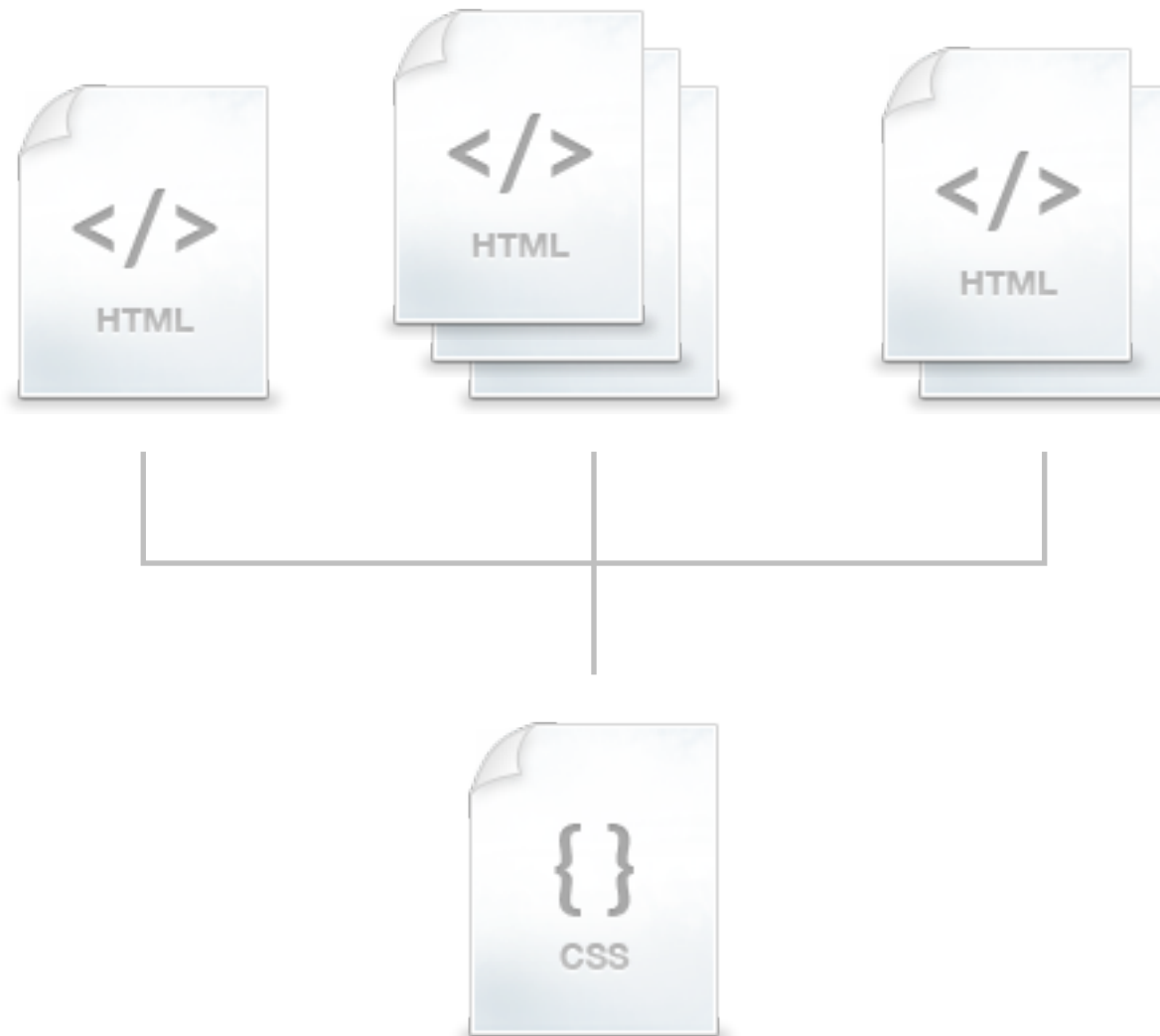
```
  <body>
```

```
    <h1>Heading principal de esta página</h1>
```

```
    <p>Contenido dentro de un párrafo</p>
```

```
  </body>
```

```
</html>
```



CSS

Cómo funciona

También es posible definir reglas CSS en un archivo HTML, ya sea utilizando la etiqueta *style* en el *head* del mismo, o el atributo *style* en cualquier etiqueta que esté dentro del *body*.

```
<!doctype html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Título de esta página</title>
```

```
    <style>
```

```
      h1 { color: #FF9900; }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Heading principal de esta página</h1>
```

```
    <p style="margin-left: 50px;">Contenido dentro de un párrafo</p>
```

```
  </body>
```

```
</html>
```

CSS

Cómo funciona

El término 'cascada' hace referencia al orden de aplicación de los estilos, siendo siempre las últimas reglas más específicas, las que finalmente definirán la presentación de cada etiqueta.

```
<!doctype html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Título de esta página</title>
```

```
    <link rel="stylesheet" href="style.css">
```

```
    <style>
```

```
      p { color: green; }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Heading principal de esta página</h1>
```

```
    <p style="color: red;">Contenido dentro de un párrafo</p>
```

```
  </body>
```

```
</html>
```

Estilos por defecto

Es importante saber que la mayoría de navegadores aplican al contenido de las páginas web estilos CSS por defecto, con el fin de diferenciar visualmente las etiquetas que las componen.

Estilos por defecto

Para evitar que los estilos aplicados por un navegador alteren la presentación de nuestras páginas, debemos anularlos siempre. ¿Cómo? Haciendo uso del conocido *CSS Reset* de Eric Meyer.

<https://meyerweb.com/eric/tools/css/reset>



CSS

Estilos por defecto

El código de Eric Meyer debe pegarse justo al principio del archivo *style.css*. Tras él, nuestros estilos, que ya no se verán afectados por los estilos por defecto de ningún navegador.

↳ Como el Reset es largo y complejo, es conveniente comprimirlo para que ocupe menos en todos los sentidos. Uno de los minimizadores de CSS más populares es cssminifier.com.

Sintaxis

En CSS, los espacios en blanco entre selector y declaración, o entre las distintas propiedades que componen la declaración de una regla, no modifican la correcta aplicación de la misma.

Sintaxis correcta

```
body { margin: 0; font-family: sans-serif; }
```

```
body  
{ margin: 0; font-family: sans-serif; }
```

```
body {  
    margin: 0;  
    font-family: sans-serif;  
}
```

```
body  
{  
    margin:0;  
    font-family:sans-serif;  
}
```

CSS

Sintaxis

No obstante, escribir incorrectamente un selector, propiedad, valor, u olvidar el cierre entre propiedades, evitará la correcta aplicación de una, varias propiedades o la regla CSS completa.

<https://jigsaw.w3.org/css-validator>



Errores de sintaxis

```
body { margin: 0; font-family: sansserif; }
```

```
body  
{ margin: 0; font-family: sans-serif;
```

```
body {  
    margin: 0  
    fontfamily: sans-serif;  
}
```

```
boddy  
{  
    margin:0;  
    font - family:sans-serif;  
}
```

Errores de sintaxis

`body { margin: 0; font-family: sansserif; }` El valor *sansserif* debería ser *sans-serif*

`body
{ margin: 0; font-family: sans-serif;` Esta declaración debería incluir la llave de cierre

`body {
 margin: 0
 fontfamily: sans-serif;` La propiedad *fontfamily* debería ser *font-family*
`}`

`boddy` El selector *boddy* debería ser *body*
`{
 margin:0;
 font - family:sans-serif;` La propiedad *font - family* debería ser *font-family*
`}`

Selectores

Los selectores funcionan como enlaces entre documento y estilo, especificando cuales serán las etiquetas que se verán afectadas por las propiedades que contiene su declaración.

<https://uniwebsidad.com/libros/css/capitulo-2/selectores-basicos>



Selectores

Podrán contener nombres de etiquetas, clases, identificadores, otros atributos, pseudo-clases, pseudo-elementos, operadores... permitiendo control total sobre la aplicación de su declaración.

Selectores globales y descendentes

```
body { }
```

```
h1 a { }
```

```
ul li { }
```

```
section p span { }
```

```
article header figure img { }
```

Selectores de clase

`.destacado { }` Selector genérico

`h2.destacado { }` Selector específico

Selectores de id

`#blog { }`

`section#blog { }`

- ↳ Los selectores genéricos seleccionan cualquier etiqueta HTML a la que se haya asignado la *clase* o *id*. Los selectores específicos seleccionan la etiqueta HTML indicada, si se le ha asignado la *clase* o *id*.

Selectores de atributo

```
input[type="text"] { }
```

```
input[type="submit"] { }
```

```
a[target="_blank"] { }
```

Selectores agrupados

```
p, span, em { }
```

```
.error, .success, .warning { }
```

```
div, #news, .featured { }
```

Selectores de pseudo-clases

```
a:link { }  
a:active { }  
a:visited { }  
a:hover { }
```

```
div:before { }
```

```
section:after { }
```

```
article { }
```

```
footer#bottom_menu ul { }
```

```
.autor * { }
```

```
li a:hover { }
```

```
input[type="text"] { }
```

↳ '*' Selecciona todos los elementos de la página o descendientes de la etiqueta que le precede.

```
section header > ul { }
```

```
#main_menu li + li { }
```

```
div.column p.info.autor span:before { }
```

```
nav li a:hover, nav li a:visited { }
```

```
form div.address input[type="tel"] { }
```

- ↳ '>' Selecciona la etiqueta especificada a su derecha si es descendiente directo de la especificada a su izquierda.
- ↳ '+' Selecciona la etiqueta especificada a su derecha si se encuentra a continuación de la especificada a su izquierda.

```
body.home section#intro > header li + li a:hover { }
```

Comentarios

Debido al elevado número de líneas que puede contener una hoja de estilos CSS, el uso de comentarios se convierte en una práctica vital, ayudando a mantener el código más ordenado.

```
/* Esto es un comentario */
```



```
/* BLOG */
```

```
section#blog { width: 640px; margin: 20px 40px; }  
section#blog header { margin-bottom: 40px; position: relative; }  
section#blog header h1 { font-size: 2em; }  
section#blog p { margin: 20px 0; }  
section#blog p a { color: #DDD; }  
section#blog p a:hover { color: #999; text-decoration: underline; }
```

```
/* SIDEBAR */
```

```
aside { width: 320px; float: left; }  
aside h1 { font-size: 2em; }
```

Herencia

El valor de algunas propiedades relacionadas con las fuentes es heredado por etiquetas HTML descendientes. Esto evita tener que repetir las que controlan familia, tamaño, color, etc.

Herencia

Sin embargo, en muchas ocasiones necesitaremos cambiar el valor heredado de tales propiedades. Para ello bastará con definir un nuevo valor que afecte a la etiqueta descendiente.

```
body
{
    font-family: Helvetica, Arial, sans-serif;
    font-size: 16px;
    line-height: 140%;
    color: gray;
}
```

```
h1, h2
{
    margin: 20px 0;
    font-weight: bold;
    line-height: 120%;
    color: black;
}
```

```
h1
{
    font-size: 24px;
}
```

Specificity

Cuando una o varias reglas CSS afectan a una misma etiqueta, la que finalmente definirá su presentación será la que cuente con un selector más específico, es decir, con mayor *specificity*.

<https://uniwebsidad.com/libros/css/capitulo-2/colisiones-de-estilos>



Specificity

Se considera más específico el selector que contiene una clase, identificador, atributo o construcción más compleja. Cuanto más genérico sea el selector, menor importancia tendrán sus reglas.

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

```
p a { color: green; }
```

```
section p a { color: blue; }
```

```
section#blog p a { color: black; }
```

Propiedades

En CSS, son las propiedades quienes definen la presentación de una etiqueta HTML. A través de su uso podremos alterar el aspecto, dimensiones, colores, fondos, fuentes, posición, etc.

↳ A continuación se muestran algunas de las propiedades más comunmente utilizadas.

Tipografía

font-family
font-size
font-weight
font-style
color
line-height
letter-spacing
text-align
text-decoration
text-transform

Diseño gráfico

background
background-size
border
border-radius
opacity
box-shadow

Tamaño y espaciado

width
max-width
height
max-height
margin
padding
box-sizing

Posicionamiento

display
float
position
top
bottom
left
right
z-index

Flexbox

flex-direction
flex-wrap
justify-content
align-content
align-items

Otras

grid
transition
transform
cursor
content
overflow

<https://cssreference.io>



Ejemplos de propiedades

```
section.blog { width: 600px; padding: 20px; background: #00AAFF; }
```

```
ul#main_menu { margin-top: 100px; display: flex; }
```

```
p { font-size: 1em; line-height: 1.4; }
```

```
input[type="text"] { padding: 5px; border: 2px solid #ef7808; }
```

```
aside { min-width: 200px; position: absolute; top: 0; left: 20px; }
```

Unidades

Existen muchas propiedades CSS que requieren unidades de medida para definir sus valores. Estas unidades pueden ser absolutas o relativas, y especifican dimensiones, márgenes, etc.

<https://uniwebsidad.com/libros/css/capitulo-3/unidades-de-medida>



Unidades de medida absolutas

```
li { padding: 10px; }
```

```
aside { margin: 5pt; }
```

```
hgroup h2 { margin: 2cm; }
```

```
section.featured { width: 50mm; }
```

```
div#bio { line-height: 1.5in; }
```

```
footer { margin: 20pc; }
```

Unidades de medida relativas

```
section p { font-size: 2em; }
```

```
article { margin-left: 10rem; }
```

```
div.info { max-width: 50%; }
```

```
ul#menu { line-height: 1.5; }
```

```
aside { height: 100vh; }
```

```
.cookies-bar { width: 25vw; }
```

Valores predefinidos

Algunas propiedades funcionan únicamente asignándoles un valor predefinido, como *top* o *absolute* por ejemplo. Es de vital importancia conocerlas y tener claro los valores que aceptan.

https://developer.mozilla.org/es/docs/Web/CSS/Referencia_CSS



Algunos valores predefinidos

```
div.info { position: relative; }
```

```
aside { background: url(images/aside-bg.png) no-repeat top left; }
```

```
label { display: block; }
```

```
p.autor span { font-style: italic; }
```

```
header a { text-transform: uppercase; }
```

```
nav { transform: rotate(45deg); }
```

Colores

En CSS podemos especificar un color utilizando su nombre o un valor RGB hexadecimal, numérico o porcentual. La decisión de usar uno u otro sistema dependerá de cada desarrollador.

http://en.wikipedia.org/wiki/Web_safe_colors



Colores

```
a { color: black; }
```

```
section { background: #DDDDDD; }
```

```
span { border: 1px solid #FFD700; }
```

```
article header { background: #F90; } /* Hexadecimal abreviado */
```

```
li { border-top: 2px dotted rgb(255, 100, 50); }
```

```
p { color: rgba(0, 0, 0, 0.6); } /* Permite controlar la opacidad */
```


Fuentes

Hace años, los desarrolladores únicamente podían utilizar en sus hojas de estilos CSS las fuentes instaladas por defecto en los ordenadores: Georgia, Times New Roman, Arial, Verdana...

https://w3schools.com/cssref/css_websafe_fonts.asp



```
p.intro { font-family: Georgia, "Times New Roman", serif; }
```

- ↳ Dado este ejemplo, el navegador intentaría aplicar al párrafo la fuente Georgia. En caso de no contar con ella, intentaría aplicar Times New Roman o, como última opción, su fuente con serifa por defecto.
- ↳ Cuando usemos fuentes con nombres compuestos por varias palabras, como Times New Roman, debemos indicarnos entre comillas dobles o simples.

CSS

Fuentes

Hoy contamos con muuuuchas otras fuentes, *webfonts*, gracias a servicios como Google Fonts u otros donde las fuentes se descargan en varios formatos y se aplican utilizando *@font-face*.

<https://fonts.google.com>

<https://fontsquirrel.com>

<https://fonts.adobe.com>



```
@import url('https://fonts.googleapis.com/css2?family=Work+Sans:wght@300;700&display=swap');
```

```
body  
{  
    font-family: "Work Sans", Helvetica, Arial, sans-serif;  
    font-size: 20px;  
    font-weight: 300;  
}
```

```
h1, h2, h3, h4, h5, h6  
{  
    font-weight: 700;  
}
```

- ↳ Con Google Fonts es muy sencillo importar la familia de fuentes y los pesos deseados, usando la función *@import*.
- ↳ En este ejemplo se importa la familia Work Sans, disponible en fonts.google.com.
- ↳ Si usamos *webfonts*, asignaremos valores como 300, 700... a la propiedad *font-weight*, en lugar de *normal* o *bold*.

```
@font-face
```

```
{  
  font-family: 'DIN';  
  src: url('din.woff2') format('woff2'), url('din.woff') format('woff'),  
  url('din.ttf') format('truetype');  
  font-weight: normal;  
  font-style: normal;  
  font-display: swap;  
}
```

```
body
```

```
{  
  font-family: DIN, Helvetica, Arial, sans-serif;  
  font-size: 20px;  
}
```

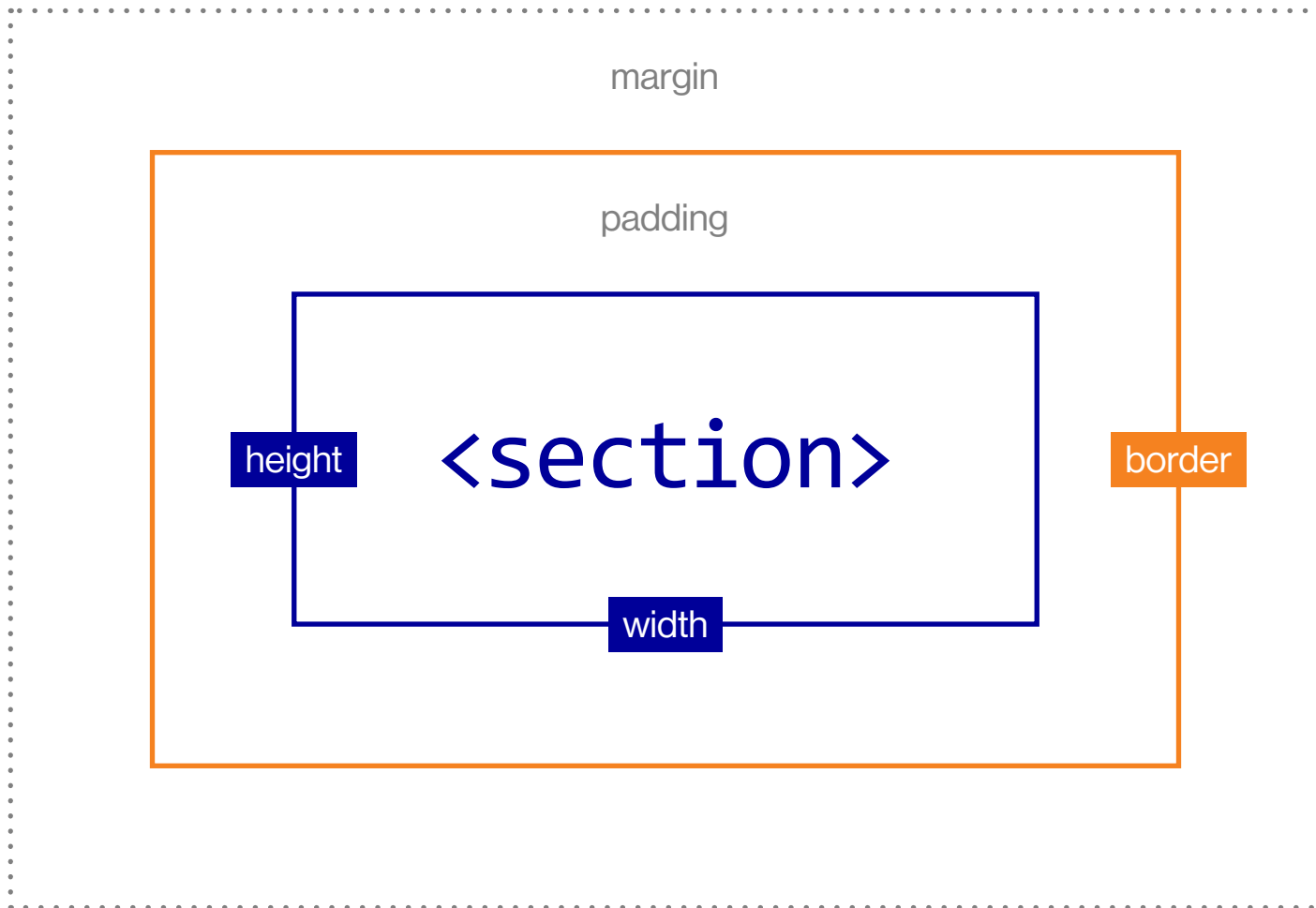
↳ Servicios como Font Squirrel, permiten subir fuentes en formato Open Type y generar los archivos necesarios para utilizarlas en nuestras hojas de estilos CSS a través de la función *@font-face*.

Box model

Sin duda uno de los conceptos más importantes, tal vez el más importante, del lenguaje CSS. Es el 'envoltorio' de cada etiqueta, construido a partir de su *width*, *height*, *padding*, *border* y *margin*.

<https://css-tricks.com/the-css-box-model>





La función *calc*

Añade lógica a propiedades como *width* y *margin*, utilizando sumas, restas, multiplicaciones y divisiones. El resultado del cálculo será aplicado según la propiedad y selector en uso.

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




```
article.project { width: calc(50% - 10px); }
```

Vendor prefixes

Estos prefijos son la forma que los navegadores usan para dar acceso a los desarrolladores a las últimas propiedades CSS, a las más nuevas, a las que todavía no se consideran estables.

```
header
{
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;

  -webkit-box-shadow: 10px 10px 20px #CCC;
  box-shadow: 10px 10px 20px #CCC;
}
```

↳ Este ejemplo muestra la correcta aplicación de las propiedades *flex* y *box-shadow*. Destacar que los *vendor prefixes* pueden aparecer en valores y propiedades, y deben añadirse antes del valor o la propiedad oficial.

CSS

Vendor prefixes

Escribirlos es difícil, ya que los navegadores evolucionan rápido y muchas veces no estaremos seguros de si una propiedad necesita prefijos o no. Para resolver nuestras dudas: Autoprefixer.

<https://autoprefixer.github.io>



Media queries

Introducidas en CSS3, permiten asignar diferentes estilos a un documento, en función de las características del dispositivo que lo consume. Su objetivo, mejorar la experiencia del usuario.

<https://mediaqueri.es>

<https://css-tricks.com/css-media-queries>



Media queries

Para ello, se extienden las propiedades del atributo *media*, al que podremos asignar valores como *screen*, *print*, *min-width*, *max-width*, *min-device-width*, *max-device-width*, *orientation*...

<https://freetanga.es/esat/media-queries>



```
@media (max-width: 1024px)
{
    nav.desktop { display: none; }
    nav.mobile { display: block; }
}
```

```
@media (min-width: 1025px)
{
    article { width: 50%; float: left; }
}
```

- ↳ Los estilos CSS declarados dentro de la primera *media query* se aplicarán mientras la anchura de la ventana del navegador sea de 1024 píxeles o menos. Los estilos declarados dentro de la segunda se aplicarán mientras la anchura de la ventana del navegador sea de 1025 píxeles o más.

```
@media only screen and (min-width: 321px) and (max-width: 480px)
{
    body { font-size: 14px; }
}
```

↳ En este ejemplo, los estilos CSS se aplicarán mientras la anchura de la ventana del navegador sea de 321 píxeles o más, pero no superando los 480 píxeles.

Praxis

Emoji Book CSS

El objetivo de este ejercicio será maquetar el diseño de Emoji Book, un portfolio de interpretaciones de Emojis por distintos artistas, aplicando estilos CSS a la estructura HTML entregada.