

CONQUER
BLOCKS

CSS



{CSS}

Clase 01



CONQUERBLOCKS

<Índice>

Introducción a css

¿Qué es CSS?

HTML + CSS

Estructura de una sentencia CSS

Can I use y código de colores

Formas de inclusión de código CSS

CONQUERBLOCKS

¿Qué es CSS?

CONQUERBLOCKS

¿Qué es CSS?

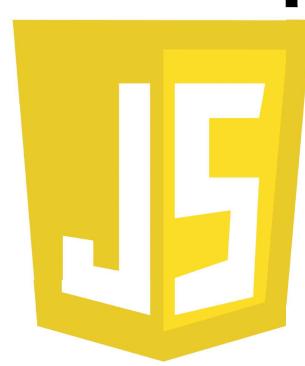
HTML



CSS



JavaScript



CONQUERBLOCKS

¿Qué es CSS?

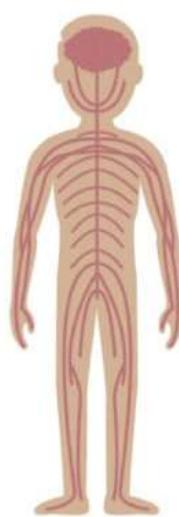
HTML the Skeleton



CSS the Skin



Javascript the Brain



CONQUERBLOCKS

¿Qué es CSS?

What's the Difference?



HTML
Hypertext Markup Language

Create the structure

- Controls the layout of the content
- Provides structure for the web page design
- The fundamental building block of any web page



CSS
Cascading Style Sheet

Stylize the website

- Applies style to the web page elements
- Targets various screen sizes to make web pages responsive
- Primarily handles the "look and feel" of a web page



JavaScript

Increase interactivity

- Adds interactivity to a web page
- Handles complex functions and features
- Programmatic code which enhances functionality

CONQUERBLOCKS

Razones por las que aprender CSS

¿Qué es CSS?

- Es el responsable de indicar cómo se verán los contenidos todos los contenidos de una web
- Trabajes en lo que trabajes en **web** debes conocer CSS en mayor o menor medida
- Toda la web está hecha de HTML + CSS
- Son los cimientos de toda web

CONQUERBLOCKS



Características

1. Texto plano - Cualquier editor
2. Lenguaje nativo de los navegadores
3. No necesita ser compilado ni transpilado
4. No es un lenguaje de programación aunque se va pareciendo
5. Es un lenguaje de marcado de la información
6. Nació con internet

CONQUERBLOCKS

¿Qué es CSS?

CSS se enseña mal

CONQUERBLOCKS

¿Por qué aprender CSS hoy en día?

- Mucho tutorial de un solo uso
- Documentación anticuada
- Se estudian framework de CSS sin saber bien html
- Se necesite un aprendizaje incremental y en detalle
- No mezclar tecnologías, al menos al principio. CSS limpio

CONQUERBLOCKS

¿Por qué aprender CSS hoy en día?

CSS

- 2º Lenguaje más importante en la web
- Se encarga de presentar la información marcada previamente mediante HTML
- Ayuda a poder separar la información de la presentación
- Evoluciona junto a HTML y los diferentes motores de los navegadores

CONQUERBLOCKS

HTML + CSS

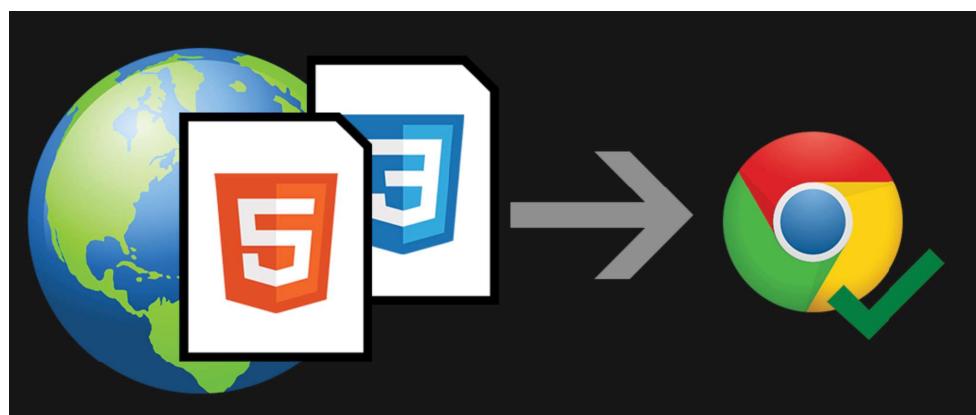
CONQUERBLOCKS

HTML + CSS

- La curva de aprendizaje de CSS suele ser sencilla
- Es fácil aprender cosas básicas, pero complicado dominar
- Puede resultar complejo adaptarse si nunca has programado (o eres totalmente ajeno al código)
- A medida que cometes errores y vas practicando, tu capacidad para escribir código CSS mejorará

CONQUERBLOCKS

HTML + CSS



CONQUERBLOCKS

HTML + CSS



CONQUERBLOCKS

HTML + CSS

¿Qué significa CSS?

- Cascading Style Sheets) significan «Hojas de estilo en cascada»
- Aplican estilos (colores, formas, márgenes, etc...) a uno o varios documentos

CONQUERBLOCKS

HTML + CSS

Ten en cuenta que generalmente tenemos varios documentos HTML en un sitio web, pero solemos tener un sólo documento CSS.

En cada documento HTML enlazamos ese único documento CSS, de modo que si hacemos cambios en él, afecta a todos los documentos HTML relacionados

CONQUERBLOCKS

¿Por qué usarlo?

- Si necesitamos hacer modificaciones visuales, lo haremos en un sólo lugar y se aplica a todo el sitio.
- Se reduce la duplicación de estilos en diferentes lugares. Es más fácil de organizar y hacer cambios.
- La información a transmitir es considerablemente menor (las páginas se descargan más rápido).
- Es más fácil crear versiones diferentes para otros dispositivos: tablets, smartphones, etc...

CONQUERBLOCKS

Estructura de una sentencia CSS

CONQUERBLOCKS

Estructura de una sentencia CSS

Estructura de un documento

- No tiene estructura alguna
- Es una relación de propiedades una detrás de otra sin orden establecido a priori
- Existen diferentes metodologías a la hora de ordenar las propiedades

CONQUERBLOCKS

Estructura de una sentencia CSS

- Los documentos CSS son archivos de texto (igual que los documentos HTML), sólo que en este caso tienen extensión .css en lugar de .html
- En el código de un documento .css se escriben una serie de órdenes y el cliente (navegador) las interpreta y aplica a los documentos HTML.
- En este apartado vamos a ver como es la sintaxis de un archivo .css y como se debe escribir el código CSS, a la vez que ponemos en práctica lo aprendido en los artículos anteriores.

CONQUERBLOCKS

Estructura de una sentencia CSS

Componentes CSS

Concepto	Descripción
Selector	Elemento o elementos del documento que vamos a seleccionar para aplicarle un estilo concreto.
Propiedad	Característica principal que vamos a definir con el valor indicado a continuación.
Valor	Cada propiedad tiene una serie de valores concretos asignables, con los que tendrá uno u otro comportamiento.
Comentario	Fragmento de texto entre /* y */ con anotaciones o aclaraciones para el desarrollador (el navegador los ignorará).
Regla	Par de propiedad y su respectivo valor asociado.

CONQUERBLOCKS

Estructura de una sentencia CSS

Componentes CSS



CONQUERBLOCKS

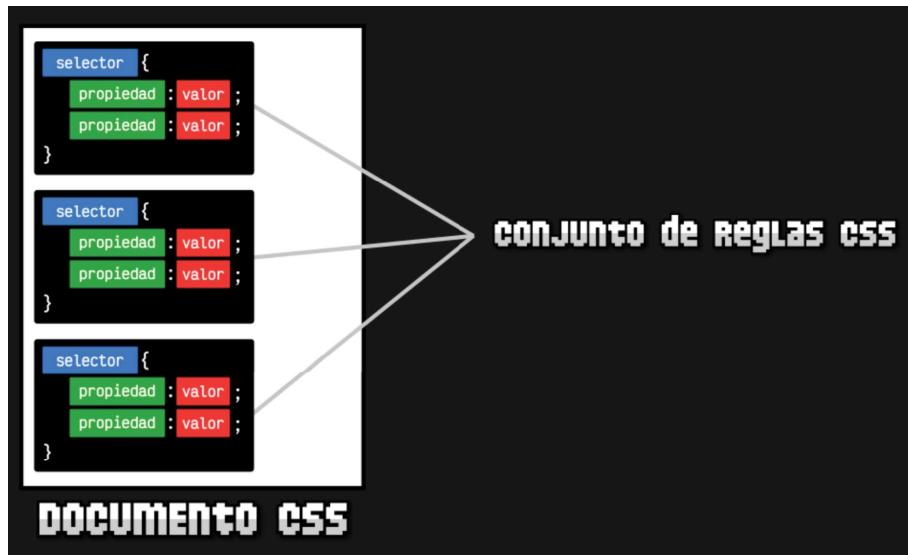
Estructura de una sentencia CSS

Ejemplo

```
p {  
  color: red; /* Color de texto */  
}
```

CONQUERBLOCKS

Frontend



CONQUERBLOCKS

Frontend

Buenas prácticas

- Escribe una regla por línea. Será mucho más fácil de leer y modificar.
- Usa la indentación. Tras escribir el carácter { al empezar un bloque de reglas, las propiedades se separan hacia la derecha.
- El último punto y coma de un bloque de reglas es opcional. Sin embargo

CONQUERBLOCKS

Frontend

Buenas prácticas

- Más tarde veremos técnicas de compresión de código
- Veremos también algún tipo de metodología de orden y nombrado de selectores
- División en ficheros para mejorar la componetización

CONQUERBLOCKS

Can I use (css)

CONQUERBLOCKS

Can I use (css)

Código de colores

CONQUERBLOCKS

Can I use (css)

Evolución de navegadores

CONQUERBLOCKS

Can I use (css)

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Google Chrome	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Internet Explorer	Trident	Completo desde la versión 7.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades a partir de la versión 10.0 del navegador
Firefox	Gecko	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Safari	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades

CONQUERBLOCKS

Can I use (css)

Responsables	Navegador web	Propósito	Motor	Notas	URL
Google	Chrome	Uso habitual	Blink		URL
Google	Chrome beta	Early-adopters	Blink		URL
Google	Canary Chrome	Desarrollador	Blink		URL
Google	Chromium	Open Source	Blink		URL
Mozilla	Firefox	Uso habitual	Quantum		URL
Mozilla	Firefox beta	Early-adopters	Quantum		URL
Mozilla	Firefox Dev Edition	Desarrollador	Quantum		URL
Mozilla	Firefox Nightly	Desarrollador	Quantum		URL
Microsoft	Edge	Uso habitual	Blink		URL
Opera	Opera	Uso habitual	Blink		URL
Opera	Opera beta	Early-adopters	Blink		URL
Opera	Opera developer	Desarrollador	Blink		URL
Opera	Opera Neon	Early-adopters	Blink		URL
Apple	Safari	Uso habitual	Webkit	Sólo Apple	URL
Apple	Safari Tech Preview	Desarrollador	Webkit	Sólo Apple	URL

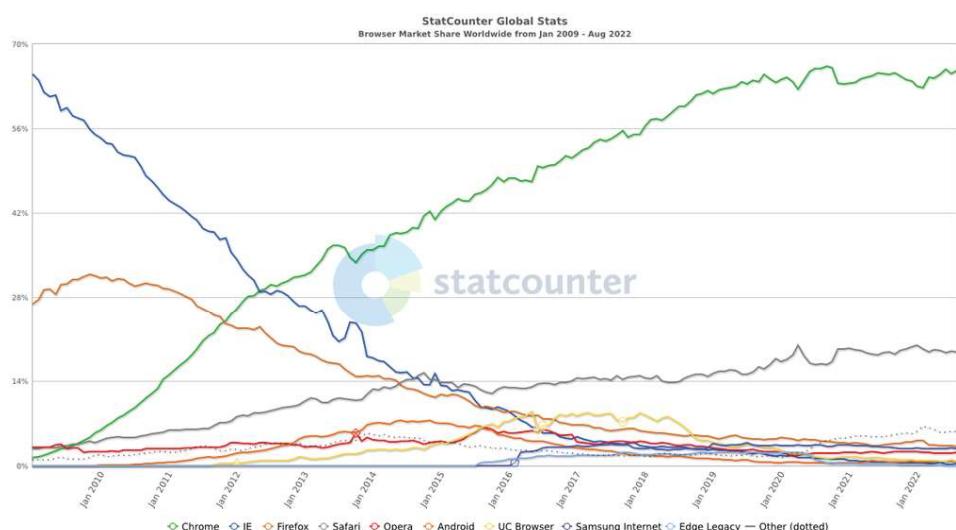
CONQUERBLOCKS

Can I use (css)

Nivel	Año	Descripción
CSS1	1996	Se crean propiedades para tipografías, colores, alineación, etc...
CSS2	1998	Propiedades de posicionamiento, tipos de medios, etc...
CSS2.1	2005	Corrige errores de CSS2 y modifica ciertas propiedades
CSS3	2011	Se comienzan a crear características de CSS como módulos independientes

CONQUERBLOCKS

Can I use (css)



CONQUERBLOCKS

Formas de inclusión de CSS en HTML

CONQUERBLOCKS

Incluir CSS

Nombre	Mediante...	Descripción
Archivo CSS externo	Etiqueta <code><link rel="stylesheet"></code>	El código se escribe en un archivo .css a parte. Recomendado.
Bloque de estilos	Etiqueta <code><style></code>	El código se escribe en una etiqueta <code><style></code> en el documento HTML.
Estilos en línea	Atributo HTML <code>style="..."</code>	El código se escribe en un atributo HTML <code>style</code> en una etiqueta.

CONQUERBLOCKS

Incluir CSS

Archivo CSS externo

CONQUERBLOCKS

Incluir CSS

```
<link rel="stylesheet" href="index.css" />
```

- Va en el head
- Antiguamente se necesitaba indicar el tipo de archivo

CONQUERBLOCKS

Incluir CSS

Bloque de estilos en el mismo documento html

CONQUERBLOCKS

Incluir CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título de la página</title>
    <style>
      div {
        background: hotpink;
        color: white;
      }
    </style>
  </head>
  ...
</html>
```

CONQUERBLOCKS

Incluir CSS

Estilos en línea

CONQUERBLOCKS

Incluir CSS

```
<p>¡Hola <span style="color: red; padding: 8px">amigo lector</span>!</p>
```

CONQUERBLOCKS

Incluir CSS

Somos team CSS Externo

CONQUERBLOCKS

Incluir CSS

Pros y Contras

CONQUERBLOCKS

Incluir CSS

Demo

CONQUERBLOCKS

Incluir CSS

Codepen

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 02



CONQUERBLOCKS

<Índice>

CSS Clase 02: Unidades de medida en CSS

¿Qué son las unidades y cuando se utilizan?

Unidades absolutas

Unidades relativas: %, em y rem

Unidades del viewport

Funciones de cálculo

Colores en CSS

CONQUERBLOCKS

Unidades de medida

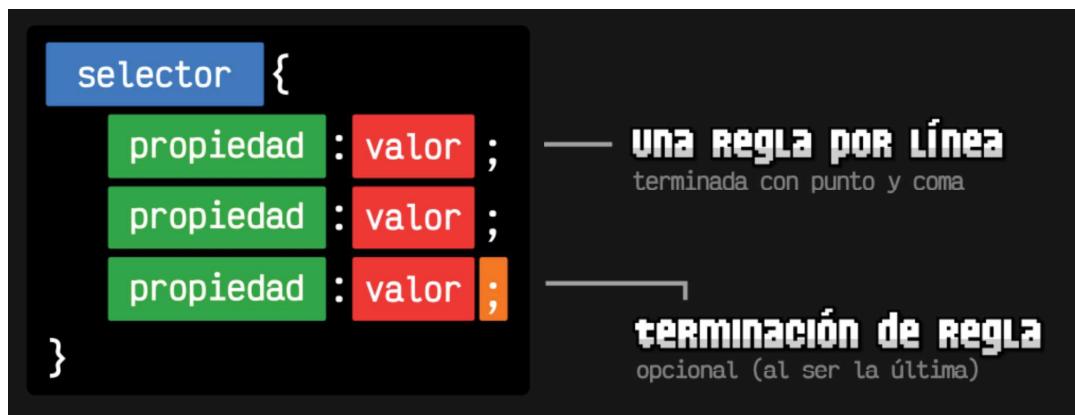
CONQUERBLOCKS

Unidades de medida

- En CSS necesitaremos utilizar tamaños
- Altos, anchos, tamaño de fuente, interlineados etc...
- Antes de empezar a ver propiedades, veremos las unidades
- Ejemplo de propiedad con unidad

CONQUERBLOCKS

Unidades de medida



CONQUERBLOCKS

Unidades de medida

```
p{  
    width: 700px;  
}
```

CONQUERBLOCKS

Unidades de medida

2 Tipos de unidades

Unidades relativas

Unidades absolutas

CONQUERBLOCKS

Unidades de medida

Consejo

Cuando estemos comenzando, empezaremos a usar unidades absolutas.

Conforme nuestra habilidad sea mayor, le cogeremos el gusto a las unidades relativas

CONQUERBLOCKS

Unidades absolutas

CONQUERBLOCKS

Unidades absolutas

- Las unidades absolutas son las más sencillas. Ya que no cambian
- Se trata de unidades que son siempre iguales a una unidad de medida física.
- La más conocida con diferencia es el píxel, una medida fácil para comenzar a trabajar en CSS.

CONQUERBLOCKS

Unidades absolutas

- Son ideales en contextos donde las medidas no varían como pueden ser en medios impresos (documentos, impresiones, etc...)
- Poco adecuadas para la web, ya que no tienen la capacidad de adaptarse a diferentes resoluciones o pantallas

CONQUERBLOCKS

Unidades absolutas

Tipos de unidades absolutas

Unidad	Significado	Medida aproximada
in	Pulgadas	1in = 25.4mm
cm	Centímetros	1cm = 10mm
pc	Picas	1pc = 4.23mm
mm	Milímetros	1mm = 1mm
pt	Puntos	1pt = 0.35mm
px	Píxeles	1px = 0.26mm
Q	Cuarto de mm	1Q = 0.248mm

CONQUERBLOCKS

Unidades absolutas

Ejemplos con unidades absolutas

```
/* El cuerpo de la página debe mostrar un margen de media pulgada */  
body { margin: 0.5in; }  
  
/* Los elementos <h1> deben mostrar un interlineado de 2 centímetros */  
h1 { line-height: 2cm; }  
  
/* Las palabras de todos los párrafos deben estar separadas 4 milímetros entre si */  
p { word-spacing: 4mm; }  
  
/* Los enlaces se deben mostrar con un tamaño de letra de 12 puntos */  
a { font-size: 12pt; }  
  
/* Los elementos <span> deben tener un tamaño de letra de 1 pica */  
span { font-size: 1pc; }
```

CONQUERBLOCKS

Unidades absolutas

La principal ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar, sin necesidad de realizar cálculos intermedios. Su principal desventaja es que son muy poco flexibles y no se adaptan fácilmente a los diferentes medios

CONQUERBLOCKS

Unidades absolutas

De todas las unidades absolutas, las más usadas son **px** para pantalla y **pt** para impresión.

Si, se pueden establecer hojas de estilo para imprimir, para eso está el atributo media.

CONQUERBLOCKS

Unidades absolutas

CSS3 Media Types	
Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smartphones etc.
speech	Used for screenreaders that "reads" the page out loud

CONQUERBLOCKS

Unidades absolutas

```
<link rel="stylesheet" href="css/estilos.css" media="screen">
```

CONQUERBLOCKS

Unidades absolutas

**Demo con width, height y
font-size
article, div y p**

CONQUERBLOCKS

Unidades absolutas

Ojo con densidad de píxeles



CONQUERBLOCKS

Unidades relativas

CONQUERBLOCKS

Unidades relativas

- Las unidades relativas son un tipo de medida mucho más potente y habitual en el CSS que creamos generalmente.
- Al contrario que las unidades absolutas, las unidades relativas dependen de algún otro factor

CONQUERBLOCKS

Unidades relativas

Tipos de unidades relativas

Unidad	Medida aproximada	Ejemplo
%	Relativa a herencia (concretamente, al elemento padre)	$50\% = \text{mitad del parent}$
em / rem	Relativo al <code>font-size</code> en ese elemento	$1.5em = 1.5 * 16px = 24px$
ex / rex	Relativo a la altura del carácter <code>x</code> minúscula	$1ex \sim 0.5em$
ch / rch	Relativo al ancho del carácter <code>0</code>	$1ch \sim 1 \text{ carácter}$
cap / rcap	Relativo a la altura del primer carácter en mayúscula	
ic / ric	Relativo al ancho del glifo 水 CJK (Chino-Japones-Koreano)	
lh / rlh	Relativo al <code>line-height</code> en ese elemento	

CONQUERBLOCKS

Unidades relativas

Probablemente, las más conocidas son %, em y rem, por lo que vamos a empezar con ellas.

CONQUERBLOCKS

Unidades relativas

La unidad de porcentaje %

Los porcentajes llevan mucho tiempo en el universo de CSS, sin embargo requieren cierta destreza y deben usarse con cuidado en nuestras primeras fases de aprendizaje, ya que el valor de porcentaje establecido depende del tamaño del elemento **padre inmediato**.

CONQUERBLOCKS

Unidades relativas

**Demo con width, height y
font-size
article, div y p**

CONQUERBLOCKS

Unidades relativas

La unidad em

La unidad em se utiliza para definir un tamaño en base al tamaño que tiene establecido el navegador en la tipografía (tamaño de la fuente) en el elemento HTML actual donde se aplica. Por defecto, y para simplificar, se puede asumir que 1em es un valor aproximado a **16px**.

CONQUERBLOCKS

Unidades relativas

La unidad em



CONQUERBLOCKS

Unidades relativas

La unidad em

Imaginemos que el tamaño de la fuente establecida en el navegador del usuario es exactamente 16px. Una cantidad 1em equivaldría a 16px, mientras que una cantidad de 2em sería justo el doble: 32px. Por otro lado, una cantidad de 0.5em sería justo la mitad: 8px.

CONQUERBLOCKS

Unidades relativas

Demo con em

CONQUERBLOCKS

Unidades relativas

La unidad rem (root element)

Muchas de las unidades que vamos a ver en esta sección, están prefijadas por la letra r. Esta letra simboliza la palabra root (raíz), por lo que en este caso estamos hablando de unidades em en el elemento raíz.

CONQUERBLOCKS

Unidades relativas

La unidad rem (root element)

Esta unidad toma la idea de la unidad em, pero en lugar de tomar el tamaño font-size del elemento actual, toma el tamaño del elemento raíz, es decir, el elemento <html> o la pseudoclase :root. Esta forma nos permite trabajar con múltiplos del tamaño base.

CONQUERBLOCKS

Unidades relativas

La unidad rem (root element)

```
:root {  
    font-size: 22px;      /* Tamaño base */  
}  
  
h1 {  
    font-size: 2rem;     /* El doble: 2 x 22px = 44px */  
}  
  
h2 {  
    font-size: 1rem;     /* El mismo: 1 x 22px = 22px */  
}
```

CONQUERBLOCKS

Unidades relativas

Demo con rem

CONQUERBLOCKS

Unidades relativas

La unidad ex

La unidad ex es una unidad menos conocida, que hace referencia al tamaño de alto de la primera letra minúscula de la tipografía. Por simplificar, podemos asumir que 1ex es aproximadamente 0.5em, es decir, la mitad del tamaño del font-size del elemento actual.

CONQUERBLOCKS

Unidades relativas

La unidad ex

La unidad ex es una unidad menos conocida, que hace referencia al tamaño de alto de la primera letra minúscula de la tipografía. Por simplificar, podemos asumir que 1ex es aproximadamente 0.5em, es decir, la mitad del tamaño del font-size del elemento actual.

CONQUERBLOCKS

Unidades del viewport

CONQUERBLOCKS

Unidades del viewport

- Existen unas unidades de nueva generación que resultan muy útiles, porque dependen del **viewport** (región visible de la página web en el navegador). Con estas unidades podemos hacer referencia a un porcentaje concreto del tamaño específico que tengamos en la ventana del navegador.

CONQUERBLOCKS

Unidades del viewport

- Es decir, cada vez que hagamos referencia a una unidad precedida por v estamos haciendo referencia a un % del tamaño total de la ventana. De esta forma, podemos hacer que elementos concretos tengan valores proporcionales al tamaño de la ventana del navegador.

CONQUERBLOCKS

Unidades del viewport

Unidad	Significado	Medida aproximada	Más info
vw	viewport width	1vw = 1% del ancho del navegador	
vh	viewport height	1vh = 1% del alto del navegador	
vmin	viewport minimum	1vmin = 1% del alto o ancho (el mínimo)	
vmax	viewport maximum	1vmax = 1% del alto o ancho (el máximo)	
vi	viewport inline	Versión lógica inline de vw y/o vh	Ver propiedades lógicas
vb	viewport block	Versión lógica en bloque de vw y/o vh	Ver propiedades lógicas

CONQUERBLOCKS

Unidades del viewport

La unidad vw (viewport width)

La unidad vw es un tipo de unidad que hace referencia al ancho del viewport (región visible de la página en el navegador). Por ejemplo, si definimos 50vw, estamos indicando un 50% del ancho actual del navegador.

CONQUERBLOCKS

Unidades del viewport

Demo

CONQUERBLOCKS

Unidades del viewport

La unidad vh (viewport height)

Por su parte, la unidad vh hace referencia de la misma forma al alto del viewport. Si con la unidad vw podemos hacer referencia al ancho, con vh hacemos referencia al alto.

CONQUERBLOCKS

Unidades del viewport

En este ejemplo, mientras 50vw referencia al 50% del ancho del navegador, con la propiedad height a 75vh estaremos haciendo referencia al 75% del alto del navegador. Si nuestro navegador tiene un tamaño de 1080px de alto, con 75vh estaríamos indicando unos 810px.

```
.container {  
    width: 50vw;  
    height: 75vh;  
}
```

CONQUERBLOCKS

Unidades del viewport

La unidad vmin (viewport min) y vmax (viewport max)

CONQUERBLOCKS

Unidades del viewport

Demo

CONQUERBLOCKS

Unidades del viewport

Consejo

CONQUERBLOCKS

Unidades del viewport

La elección de unidades es algo muy personal, pero en estos apuntes se intentará utilizar siempre las siguientes unidades:

- **px** para distancias pequeñas (anchos de bordes, márgenes, etc.).
- **rem** para tamaños de fuente, salvo que se quiera aprovechar en el diseño la composición de tamaños, en cuyo caso se usarán porcentajes %.
- **vw, vh y porcentajes %** para tamaños de elementos grandes y distribución de elementos por la pantalla.

CONQUERBLOCKS

Funciones de cálculo

CONQUERBLOCKS

Funciones de cálculo

FUNCIONES CSS MATEMÁTICAS

En este artículo vamos a ver ciertas funciones CSS matemáticas, que no son más que funciones de apoyo que podemos utilizar en CSS para realizar cálculos u operaciones sencillas de una forma fácil y sencilla, sin tener que abandonar CSS y/o recurrir a Javascript.

CONQUERBLOCKS

Funciones de cálculo

Función CSS	Descripción
Funciones de cálculos	
<code>calc()</code>	Permite calcular operaciones con unidades CSS como px, %, vw, vh u otras (incluso combinadas)
Funciones de comparación	
<code>min() / max()</code>	Permite calcular el valor mínimo o máximo de las unidades indicadas.
<code>clamp()</code>	Permite calcular valores «ajustados». Equivalente a <code>max(MIN, min(VAL, MAX))</code> .
Funciones escalonadas	
<code>round() / mod() / rem()</code>	Funciones que permiten redondear, obtener el módulo o el resto.
Funciones trigonométricas	
<code>sin() / cos() / tan()</code>	Permite obtener el seno, coseno o tangente de un valor.
<code>asin() / acos() / atan()</code>	Permite obtener el arcoseno, arcocoseno o arcotangente de un valor.
<code>atan2()</code>	Aplica la función arcotangente de dos parámetros.
Funciones exponentiales	
<code>pow() / sqrt() / hypot()</code>	Permite realizar potenciación, raíces cuadradas o hipotenusa.
<code>log() / exp()</code>	Permite realizar logaritmos o potencias de e.
Otras	
<code>e / pi / Infinity / NaN</code>	Palabras clave que representan valores matemáticos. Ver constantes o NaN
<code>attr()</code>	Permite obtener el valor de un atributo desde CSS.

CONQUERBLOCKS

Funciones de cálculo

Función calc()

Es posible que en algunas ocasiones necesitemos indicar valores precalculados por el navegador, ya porque sea más cómodo o porque simplemente queremos hacer el código más organizado o fácil de mantener. Por ejemplo, la suma de dos valores que a priori desconocemos o no sabemos exactamente cuantos suman, pero que el navegador si puede calcularlo fácilmente.

CONQUERBLOCKS

Funciones de cálculo

```
.elemento {  
    width: calc(200px + 1em);  
    height: 300px;  
    background: blue;  
}
```

CONQUERBLOCKS

Funciones de cálculo

Como se puede ver, se pueden usar operaciones como: sumas (+), restas (-), multiplicaciones (*) o divisiones (/) que utilicen alguna de las unidades soportadas por CSS, como por ejemplo, números, dimensiones, porcentajes, tiempos, ángulos, etc...

CONQUERBLOCKS

Funciones de cálculo

Función min() y max()

En algunas ocasiones, necesitaremos que en una propiedad se aplique un valor entre varias posibilidades posibles, y lo que te interesa es el valor más pequeño.

CSS permite realizar esto utilizando la función min() que nos permite elegir el valor más pequeño de 2 o más posibilidades aplicadas por parámetro:

```
.elemento {  
    width: min(200px, 25%);  
    height: 200px;  
    background: red;  
}
```

CONQUERBLOCKS

Funciones de cálculo

Esto aplicaría un width de 200px, siempre y cuando el 25% del elemento padre sea más grande de 200px. Todo este cálculo, generalmente realizado en Javascript debido a su calidad de lógica de cálculo, puede realizarse desde CSS si sólo se utiliza exclusivamente para esta finalidad.

Recuerda que tanto la función min() como la función max() permite tener más de 2 parámetros, no tiene que tener obligatoriamente 2 parámetros.

CONQUERBLOCKS

Colores en CSS

CONQUERBLOCKS

Colores en CSS

Hay múltiples formas de definir un color en CSS, algunas más sencillas, otras más complejas.

CONQUERBLOCKS

Colores en CSS

Antes de nada vamos a ver 2 propiedades que os va a servir para poder hacer pruebas.

color
background-color

CONQUERBLOCKS

Colores en CSS

Propiedad	Valor	Significado
color	COLOR	Cambia el color del texto que está en el interior de un elemento.
background-color	COLOR	Cambia el color de fondo de un elemento.

```
.element {  
    background-color: black; /* Color de fondo */  
    color: white;           /* Color de texto */  
}
```

CONQUERBLOCKS

Colores en CSS

Formas de establecer colores

Esquema	Descripción	Más info
Espacio de color RGB		
red	Establece un color mediante palabras clave.	Ver palabras clave predefinidas
rgb()	Usa una función <code>rgb()</code> (rojo, verde y azul).	Ver función RGB
#rrggbb	Notación RGB abreviada en hexadecimal. Notación recomendada	Ver formato Hexadecimal
hsl()	Usa una función <code>hsl()</code> (color, saturación y brillo).	Ver función HSL
hwb()	Usa una función <code>hwb()</code> (color, claridad y oscuridad).	Ver función HWB
Espacio de color independiente del dispositivo		
lab()	Usa una función <code>lab()</code> y <code>oklab()</code> (luminosidad CIE, eje A y eje B).	Ver función LAB y OKLAB
lch()	Usa una función <code>lch()</code> y <code>oklch()</code> (luminosidad CIE, saturación, color).	Ver función LCH y OKLCH

CONQUERBLOCKS

Colores en CSS

Palabras clave de color

CONQUERBLOCKS

Colores en CSS

Función RGB

CONQUERBLOCKS

Colores en CSS

Formato Hexadecimal

CONQUERBLOCKS

Colores en CSS

Demo

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 03



CONQUERBLOCKS

<Índice>

Clase 03

Tipos de etiquetas y CSS del navegador

Etiquetas de bloque

Etiquetas de línea

Navegadores y motores de render

Inspector de elementos a fondo

CSS por defecto

Normalizar o Restablecer CSS

CONQUERBLOCKS

Etiquetas de bloque

CONQUERBLOCKS

Etiquetas de bloque

- Cómo son renderizadas
- Tipos de etiquetas de bloque
- Dos etiquetas de bloque juntas
- Una etiqueta de bloque dentro de otra
- Una etiqueta de bloque con una de línea

CONQUERBLOCKS

Etiquetas de bloque

CONQUERBLOCKS

Etiquetas de bloque

DEMO

CONQUERBLOCKS

Etiquetas de línea

CONQUERBLOCKS

Etiquetas de línea

- Cómo son renderizadas
- Tipos de etiquetas de línea
- Dos etiquetas de linea juntas
- Una etiqueta de linea dentro de otra
- Una etiqueta de bloque con una de línea

CONQUERBLOCKS

Etiquetas de línea

Los elementos en línea definidos por HTML son:

a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

CONQUERBLOCKS

Etiquetas de línea

DEMO

CONQUERBLOCKS

Navegadores y motores de render

CONQUERBLOCKS

Navegadores y motores de render

En un mundo ideal, todas las páginas webs se verían correctamente y de la misma forma en todos los navegadores web disponibles, sin embargo, y una de las cosas que más llama la atención del diseño web cuando estamos empezando, es que no sólo debemos construir una web correctamente, sino que además debemos ser conscientes de los navegadores más utilizados, así como de sus carencias y virtudes.

CONQUERBLOCKS

Navegadores y motores de render

En un principio, el consorcio W3C se encarga de definir unas especificaciones y «normas» de recomendación, para que posteriormente, las compañías desarrolladoras de navegadores web las sigan y puedan crear un navegador correctamente.

CONQUERBLOCKS

Navegadores y motores de render

No estamos en un mundo perfecto

CONQUERBLOCKS

Navegadores y motores de render

Hoja de ruta de cambios

Responsables	Producto	Página de desarrollo
Microsoft	Navegador Microsoft Edge	Edge Platform Status
(Proyecto open source)	Motor Webkit	Webkit Feature Status
Fundación Mozilla	Navegador Mozilla Firefox	Mozilla Standards Positions
Google	Navegador Google Chrome	Chrome Platform Status

CONQUERBLOCKS

Navegadores y motores de render

Prefijos de navegador

- Algunas de las propiedades que veremos no están definidas por completo en todos los motores.
- Usando prefijos de motor de render, podemos utilizar varios prefijos para asegurarnos que aunque dichas funcionalidades tengan un comportamiento o sintaxis diferente en cada navegador, podemos hacer referencia a cada una de ellas por separado:

CONQUERBLOCKS

Navegadores y motores de render

Prefijos de navegador

```
div {  
    transform: ... /* El navegador implementa la especificación oficial */  
    -webkit-transform: ... /* Chrome antiguo / Safari (Motor Webkit) */  
    -moz-transform: ... /* Firefox antiguo (Motor Gecko) */  
    -ms-transform: ... /* Internet Explorer (Motor Trident) */  
    -o-transform: ... /* Opera antiguo (Motor Presto) */  
}
```

CONQUERBLOCKS

Inspector de elementos a fondo

CONQUERBLOCKS

Inspector de elementos a fondo

El mejor amigo de los desarrolladores web

CONQUERBLOCKS

Inspector de elementos a fondo

I. Introducción al Inspector de Elementos

- A. Definición y propósito del Inspector de Elementos
- B. Acceso al Inspector de Elementos en Google Chrome

CONQUERBLOCKS

Inspector de elementos a fondo

II. Estructura del Inspector de Elementos

- A. Visión general de la interfaz del Inspector de Elementos
- B. Panel de elementos y árbol DOM
 - 1. Explicación de la jerarquía del árbol DOM
 - 2. Navegación y selección de elementos
- C. Panel de estilos y propiedades
 - 1. Visualización de estilos aplicados
 - 2. Modificación de estilos en tiempo real

CONQUERBLOCKS

Inspector de elementos a fondo

III. Utilizando las herramientas del Inspector de Elementos

- A. Inspeccionar y editar elementos
 - 1. Inspeccionar elementos en la página
 - 2. Modificar atributos y contenido de elementos
- B. Depurar y solucionar problemas
 - 1. Identificar errores de CSS y JavaScript
 - 2. Depurar eventos y flujo de código
- C. Optimizar rendimiento y carga de la página
 - 1. Evaluar tiempos de carga y recursos
 - 2. Analizar y mejorar el rendimiento de la página

CONQUERBLOCKS

Inspector de elementos a fondo

IV. Funcionalidades avanzadas y trucos útiles

- A. Uso de atajos de teclado y comandos
- B. Pruebas de elementos y estilos en tiempo real
- C. Simulación de dispositivos
- D. Simulación de conexiones

CONQUERBLOCKS

Inspector de elementos a fondo

Hay más navegadores con sus respectivas developers tools

CONQUERBLOCKS

Inspector de elementos a fondo



CONQUERBLOCKS

CSS por defecto

CONQUERBLOCKS

CSS por defecto

Todos los navegadores tienen una hoja de estilos por defecto

CONQUERBLOCKS

CSS por defecto

Esto hará que los estilos luchen con la cascada que lo veremos en el tema siguiente

CONQUERBLOCKS

CSS por defecto

Demo en varios navegadores
cualquier archivo de formularios o
archivos del proyecto

CONQUERBLOCKS

Normalizar o Restablecer CSS

CONQUERBLOCKS

Normalizar o Restablecer

¿Por qué?

CONQUERBLOCKS

Normalizar o Restablecer

2 opciones



margin-block-start: 0.67em;

CONQUERBLOCKS

Normalizar o Restablecer

Como regla, queremos que los elementos HTML se vean de la misma manera, independientemente del navegador que se use para verlos. Desafortunadamente, este no es el caso debido a la forma en que se ejecutan los navegadores.

CONQUERBLOCKS

Normalizar o Restablecer

La cuestión de qué enfoque usar para eliminar las diferencias entre los estilos de agente de usuario es un debate en curso entre Normalizar CSS y Restablecer CSS.

CONQUERBLOCKS

Normalizar o Restablecer

Hablemos de la hoja de estilos del navegador

```
h1 {                               user agent stylesheet
  display: block;
  font-size: 2em;
  margin-block-start: 0.67em;
  margin-block-end: 0.67em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
  font-weight: bold;
}
```

<h1> estilos predeterminados en Chrome

CONQUERBLOCKS

Normalizar o Restablecer

Normalizar CSS

Es un pequeño archivo CSS que proporciona coherencia entre navegadores en el estilo predeterminado de los elementos HTML

Eso significa que si miramos los estándares W3C de los estilos aplicados por los navegadores, y hay una inconsistencia en uno de los navegadores, los normalize.css estilos arreglarán el estilo del navegador que tiene la diferencia.

CONQUERBLOCKS

Normalizar o Restablecer

Normalize.css es un proyecto de código abierto, continuamente actualizado en Github, creado por Nicolas Gallagher.

CONQUERBLOCKS

Normalizar o Restablecer

Normalize.css

CONQUERBLOCKS

Normalizar o Restablecer

Restablecer CSS

Reset CSS adopta un enfoque diferente y dice que **no necesitamos los estilos predeterminados de los navegadores en absoluto**. Sean cuales sean los estilos que necesitemos, los definiremos en el proyecto según nuestras necesidades. Entonces, "Restablecer CSS" restablece todos los estilos que vienen con el agente de usuario del navegador.

CONQUERBLOCKS

Normalizar o Restablecer

Restablecer CSS

Mucho más agresivo que Normalizar

No heredaremos nada

Tendremos que darle estilo a todo, lo que es normal

CONQUERBLOCKS

Normalizar o Restablecer

reset.css

CONQUERBLOCKS

Normalizar o Restablecer

Normalize CSS

VS

CSS Reset

CONQUERBLOCKS

Normalizar o Restablecer

¿Qué uso entonces?

CONQUERBLOCKS

Normalizar o Restablecer

Demo

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 05



CONQUERBLOCKS

Clase anterior

Clase anterior

Cascada y prioridad

Normalizar o Restablecer CSS

¿Qué significa hojas de estilo en cascada?

Cascada en CSS

Herencia

Especificidad

!important

CONQUERBLOCKS

<Índice>

Clase 04

Clase 5: Selectores I

Selector universal

Selector de etiqueta

Selector de clase

Selector de ID

Selector Mixto

Selector por atributos básicos

Selector por atributos avanzado

CONQUERBLOCKS

Selector

CONQUERBLOCKS

Selector

Componentes CSS



CONQUERBLOCKS

Selector

Como su mismo nombre indica, un selector CSS es una referencia a uno o varios elementos HTML, escrito generalmente desde código CSS (también pueden utilizarse desde ciertos lugares en Javascript).

CONQUERBLOCKS

Selector

Cuando estamos en una fase de aprendizaje muy inicial con CSS, es habitual dar estilo con selectores muy sencillos, donde generalmente utilizamos un selector genérico que representa una etiqueta HTML (por ejemplo div). Sin embargo, con el tiempo aprendemos a utilizar selectores más complejos que nos permitirán hacer más cosas y escribir menos.

CONQUERBLOCKS

Selector universal

CONQUERBLOCKS

Selector universal

*

Se utiliza para seleccionar todos los elementos de la página. El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML

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

CONQUERBLOCKS

Selector universal

- Es el selector menos específico
- A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.
- No obstante, sí que se suele combinar con otros selectores y además, forma parte de algunos hacks muy utilizados, como se verá más adelante.

CONQUERBLOCKS

Selector universal

DEMO

CONQUERBLOCKS

Selector de etiqueta

CONQUERBLOCKS

Selector de etiqueta

nombre_de_etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:

CONQUERBLOCKS

Selector de etiqueta

```
p {  
    ...  
}
```

CONQUERBLOCKS

Selector de etiqueta

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar.

CONQUERBLOCKS

Selector de etiqueta

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

CONQUERBLOCKS

Selector de etiqueta

DEMO

CONQUERBLOCKS

Selector de clase

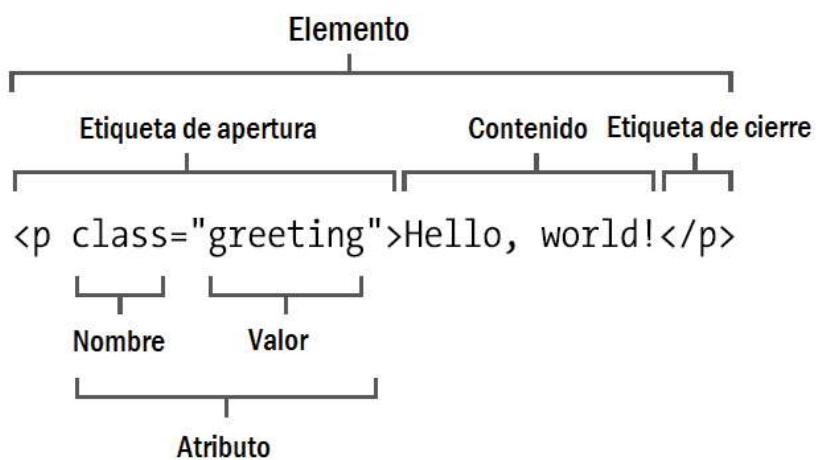
CONQUERBLOCKS

Selector clase

¿Para qué usamos las clases?

CONQUERBLOCKS

Selector clase



CONQUERBLOCKS

Selector clase

Algunos atributos comunes

Atributo	Descripción
class	Permite agrupar elementos. El valor de este atributo es una cadena de caracteres definida por el autor de la página. Si varios elementos comparten el mismo valor, se dice que son de la misma clase. Se utiliza para asignar estilos y durante la programación con JavaScript.
contenteditable	Si existe, el contenido del elemento es editable.
hidden	Si existe, el elemento no se representa en el navegador.
id	Identifica al elemento de manera única en el documento. El valor es una cadena de caracteres definida por el autor de la página. Se utiliza para asignar estilos y durante la programación con JavaScript.
lang	Determina el idioma en el que está escrito el contenido del elemento.
spellcheck	Determina si el elemento debe ser analizado ortográfica y gramaticalmente por el navegador.
style	Permite asignar un estilo al componente.
tabindex	Determina el orden de selección de elementos cuando se pulsa el tabulador.
title	Permite especificar información extra al elemento.
translate	Determina si el elemento debe ser traducido o no.

CONQUERBLOCKS

Selector clase

¿Para qué usaremos las clases?

```
<body>
  <p>Lorem ipsum dolor sit amet...</p>
  <p>Nunc sed lacus et est adipiscing accumsan...</p>
  <p>Class aptent taciti sociosqu ad litora...</p>
</body>
```

¿Cómo selecciono el primero de los párrafos?

CONQUERBLOCKS

Selector de clase

Ejemplo

```
<body>
  <p class="destacado">Lorem ipsum dolor sit amet...</p>
  <p>Nunc sed lacus et <a href="#" class="destacado">est adipiscing</a> accumsan...</p>
  <p>Class aptent taciti <em class="destacado">sociosqu ad</em> litora...</p>
</body>
```

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

CONQUERBLOCKS

Selector de clase

```
<!DOCTYPE html>
<html>
<head>
  <title>Documento de ejemplo</title>
</head>
<body>
  <div id="main">
    <p>Escoge uno de los siguientes botones:</p>

    <button class="classic">Opción 1</button>
    <button class="classic">Opción 2</button>
    <button class="classic">Opción 3</button>
    <button class="back">Volver</button>
  </div>
</body>
</html>
```

Ejemplo

```
.classic {
  background-color: green;
  color: white;
}

.back {
  background-color: orange;
  color: white;
}
```

CONQUERBLOCKS

Selector de clase

- Selector multiclas
- Ejemplo de colores
- Ejemplo de tamaños
- Combinación

CONQUERBLOCKS

Selector de clase

DEMO

CONQUERBLOCKS

Selector ID

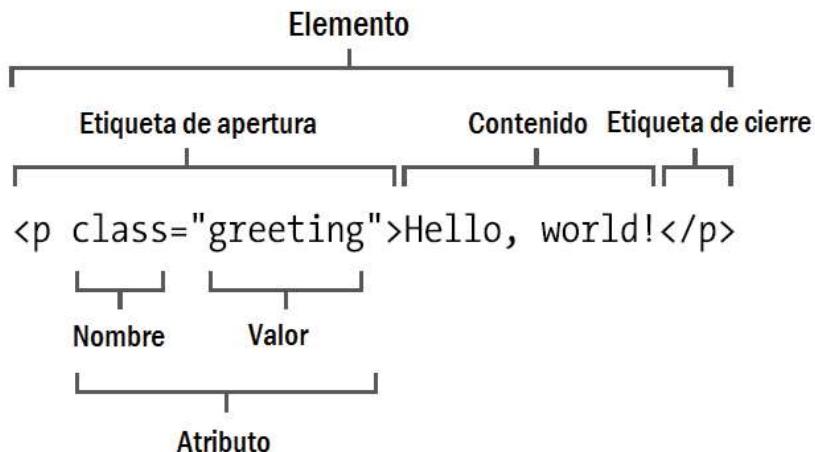
CONQUERBLOCKS

Selector ID

¿Qué es un id?

CONQUERBLOCKS

Selector ID



CONQUERBLOCKS

Selector ID

Algunos
atributos
comunes

Atributo	Descripción
class	Permite agrupar elementos. El valor de este atributo es una cadena de caracteres definida por el autor de la página. Si varios elementos comparten el mismo valor, se dice que son de la misma clase. Se utiliza para asignar estilos y durante la programación con JavaScript.
contenteditable	Si existe, el contenido del elemento es editable.
hidden	Si existe, el elemento no se representa en el navegador.
id	Identifica al elemento de manera única en el documento. El valor es una cadena de caracteres definida por el autor de la página. Se utiliza para asignar estilos y durante la programación con JavaScript.
lang	Determina el idioma en el que está escrito el contenido del elemento.
spellcheck	Determina si el elemento debe ser analizado ortográfica y gramaticalmente por el navegador.
style	Permite asignar un estilo al componente.
tabindex	Determina el orden de selección de elementos cuando se pulsa el tabulador.
title	Permite especificar información extra al elemento.
translate	Determina si el elemento debe ser traducido o no.

CONQUERBLOCKS

Selector ID

¿Para qué usaremos los ID's?

CONQUERBLOCKS

Selector ID

¿Para qué usaremos los ID?

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.

CONQUERBLOCKS

Selector ID

¿Para qué usaremos los ID?

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

CONQUERBLOCKS

Selector ID

Ejemplo

```
#destacado { color: red; }

<p>Primer párrafo</p>
<p id="destacado">Segundo párrafo</p>
<p>Tercer párrafo</p>
```

CONQUERBLOCKS

Selector ID

```
<!DOCTYPE html>
<html>
<head>
  <title>Documento de ejemplo</title>
</head>
<body>
  <div id="saludo">
    <p>¡Hola, visitante! ¡Bienvenido a esta página!</p>
  </div>

  <div id="main">
    <p>En esta página encontrarás los siguientes temas:</p>
  </div>
</body>
</html>
```

Ejemplo

```
#saludo {
  background-color: blue;
  color: white;
}
```

CONQUERBLOCKS

Selector ID

DEMO

CONQUERBLOCKS

Selector mixto

CONQUERBLOCKS

Selector mixto

¿Combinamos etiqueta, id y clases?

CONQUERBLOCKS

Selector mixto

- Selector universal + clase
- Selector universal + id
- Selector de etiqueta + clase
- Selector de clase + clase
- Selector de etiqueta + id ¿?
- Selector de id + clase

CONQUERBLOCKS

Selector mixto

Existen muchas más combinaciones que iremos viendo conforme veamos más selectores

CONQUERBLOCKS

Selector mixto

ESPECIFICIDAD

CONQUERBLOCKS

Selector mixto

DEMO

CONQUERBLOCKS

Selector de atributos básicos

CONQUERBLOCKS

Selector de atributos básicos

Recordemos lo que son los atributos

CONQUERBLOCKS

Selector de atributos básicos

CONQUERBLOCKS

Selector de atributos básicos

Hay muchas formas de seleccionar elementos con determinados atributos. El más sencillo será

etiqueta[atributo]

CONQUERBLOCKS

Selector de atributos básicos

Si se escribe una etiqueta seguida del nombre de un atributo entre corchetes, E[atributo], las propiedades afectan a todos los elementos que tengan establecidos ese atributo. En el ejemplo siguiente, los párrafos con clase, independientemente de la clase asignada, se muestran en mayúsculas.

CONQUERBLOCKS

Selector de atributos básicos

Ejemplo

```
p[class] {  
    color: blue;  
}
```

CONQUERBLOCKS

Selector de atributos básicos

Ejemplo

```
p[class="azul"] {  
    color: blue;  
}
```

CONQUERBLOCKS

Selector de atributos básicos

Ejemplo

```
p[class="azul grande"]  
{  
    color: blue;  
}
```

CONQUERBLOCKS

Selector de atributos básicos

Vale para cualquier tipo de atributo

CONQUERBLOCKS

Selector de atributos avanzados

CONQUERBLOCKS

Selector de atributos avanzados

El tema de los atributos se puede complicar

CONQUERBLOCKS

Selector de atributos avanzados

- [nombre_atributo], selecciona los elementos que tienen establecido el atributo llamado nombre_atributo, independientemente de su valor.

```
[disabled] {  
    background: grey;  
}
```

CONQUERBLOCKS

Selector de atributos avanzados

- [nombre_atributo=valor], selecciona los elementos que tienen establecido un atributo llamado nombre_atributo con un valor igual a valor.

```
a[rel="nofollow"] {  
    background: red;  
}
```

CONQUERBLOCKS

Selector de atributos avanzados

- [nombre_atributo~=valor], selecciona los elementos que tienen establecido un atributo llamado nombre_atributo y al menos uno de los valores del atributo es valor.

```
<div data-mood="happy sad">Uno</div>  
  
<style>  
[data-mood~="happy"] {  
    background: green;  
    color: white;  
    padding: 5px;  
}  
</style>
```

CONQUERBLOCKS

Selector de atributos avanzados

- [nombre_atributo|=valor], selecciona los elementos que tienen establecido un atributo llamado nombre_atributo y cuyo valor es una serie de palabras separadas con guiones, pero que comienza con valor. Este tipo de selector sólo es útil para los atributos de tipo lang que indican el idioma del contenido del elemento.

CONQUERBLOCKS

Selector de atributos avanzados

```
<div lang="es-MX en-EN">English</div>

<style>
[lang|= "es"] {
    background: red;
    color: white;
    padding: 5px;
}
</style>
```

CONQUERBLOCKS

Selector de atributos avanzados

Atributo	¿Cuándo se aplica el estilo al elemento?
Básicos	
[href]	Si tiene un atributo href.
[href="#top"]	Si tiene un atributo href y su valor es #top.
[class~="manzdev"]	Si tiene un atributo class con una lista de valores y uno de ellos es manzdev.
[lang ="es"]	Si tiene un atributo lang con una lista de valores, donde uno empieza por es-.
Avanzados	
[href^="https://"]	Si tiene un atributo href y su valor comienza por https://.
[href\$=".pdf"]	Si tiene un atributo href y su valor termina por .pdf (un enlace a un PDF).
[href*="manzdev"]	Si tiene un atributo href y su valor contiene manzdev.

CONQUERBLOCKS

Selector de atributos avanzados

DEMO

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 06



CONQUERBLOCKS

Clase anterior

[Clase anterior](#)

[Cascada y prioridad](#)

[Selector universal](#)

[Selector de etiqueta](#)

[Selector de clase](#)

[Selector de ID](#)

[Selector Mixto](#)

[Selector por atributos básicos](#)

[Selector por atributos avanzado](#)

CONQUERBLOCKS

<Índice>

Clase 06

Clase 6: Selectores II

Selector descendente: doble, triple, etc...

Selector de hijo

Selector de hermano adyacente

Selector de hermanos

Selector múltiple

La importancia de un espacio y un punto

Especificidad

CONQUERBLOCKS

Selector

CONQUERBLOCKS

Selector

Componentes CSS



CONQUERBLOCKS

Selector

Como su mismo nombre indica, un selector CSS es una referencia a uno o varios elementos HTML, escrito generalmente desde código CSS (también pueden utilizarse desde ciertos lugares en Javascript).

CONQUERBLOCKS

Selector

Cuando estamos en una fase de aprendizaje muy inicial con CSS, es habitual dar estilo con selectores muy sencillos, donde generalmente utilizamos un selector genérico que representa una etiqueta HTML (por ejemplo div). Sin embargo, con el tiempo aprendemos a utilizar selectores más complejos que nos permitirán hacer más cosas y escribir menos.

CONQUERBLOCKS

Selector descendente

CONQUERBLOCKS

Selector descendente

a b

En CSS podemos utilizar lo que se llama el selector descendiente, que no es más que una forma de seleccionar ciertos elementos que **están dentro de otros elementos**. Esto puede parecer sencillo, pero cuidado, ya que puede ser una fuente de problemas si no se comprende bien.

CONQUERBLOCKS

Selector descendente

Su sintaxis se basa en colocar los elementos uno a continuación de otro **separado por un espacio**

CONQUERBLOCKS

Selector descendente

¿Tenemos controlado lo que es un descendiente?

CONQUERBLOCKS

Selector descendente

```
div#page div {  
    background-color: gold;  
}
```

CONQUERBLOCKS

Selector descendente

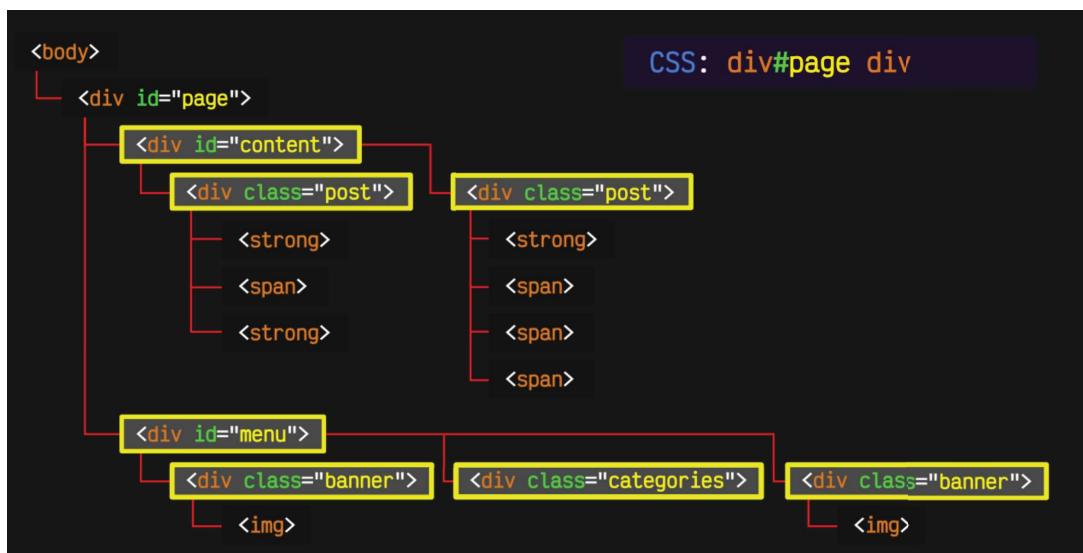
En el ejemplo anterior, aplicamos los estilos CSS
(color azul de fondo)

a todos los elementos `<div>` que estén dentro del elemento
`<div id="page">`

De esta forma, si existe un elemento `<div>` fuera del elemento con
`id="page"`, no se aplicarán los estilos indicados.

CONQUERBLOCKS

Selector descendente



CONQUERBLOCKS

Selector descendente

- Repasemos varios detalles importantes respecto a este combinador CSS:
- Se están seleccionando todos los elementos <div> que están dentro del elemento con id="page".
- Observa que se seleccionan independientemente del nivel al que estén (hijos, nietos, ...)

CONQUERBLOCKS

Selector descendente

- En este caso, el div de div#page es innecesario, ya que habíamos dicho que los id no se pueden repetir. Si ya existe un elemento con id page, no hace falta diferenciarlo también por etiqueta. Si se tratase de una clase, si podría usarse.

CONQUERBLOCKS

Selector descendente

Se pueden construir selectores muy complejos con tantos elementos como se quiera, pero una buena práctica es intentar mantenerlos siempre lo más simples posibles

CONQUERBLOCKS

Selector descendente

Cuántos más elementos descendientes existan en un selector, más complejo será el procesamiento de dicha regla por los navegadores y más específico será el navegador, lo que puede introducir algunas complicaciones.

CONQUERBLOCKS

Selector descendente

```
<div class="menu">
  <div class="options">
    <ul>
      <li><a href="/one">Option 1</a></li>
      <li><a href="/two">Option 2</a></li>
      <li><a href="/three">Option 3</a></li>
    </ul>
  </div>
</div>
```

CONQUERBLOCKS

Selector descendente

```
/* Forma 1: Muy específica */
.menu .options ul li a {
  color: orange;
}

/* Forma 2: Más general, menos específica */
.menu a {
  color: orange;
}
```

CONQUERBLOCKS

Selector descendente

DEMO

CONQUERBLOCKS

Selector de hijo

CONQUERBLOCKS

Selector de hijo

a > b

Es posible también hacer referencia a los elementos hijos de un elemento. Es decir, en el ejemplo de arriba, seleccionaremos los elementos b que sean hijos de los elementos de tipo a.

CONQUERBLOCKS

Selector de hijo

```
#page > div {  
    background-color: gold;  
}
```

CONQUERBLOCKS

Selector de hijo



CONQUERBLOCKS

Selector de hijo

Al contrario que en el caso anterior, no se seleccionan todos los elementos `<div>` descendientes, sino solo aquellos que son hijos directos del primer elemento especificado.

CONQUERBLOCKS

Selector de hijo

DEMO

CONQUERBLOCKS

Selector de hermano adyacente

CONQUERBLOCKS

Selector de hermano adyacente

¿Qué es un hermano?

CONQUERBLOCKS

Selector de hermano adyacente

¿Qué es un hermano adyacente?

CONQUERBLOCKS

Selector de hermano adyacente

a + b

Es posible también hacer referencia a los elementos hermanos, es decir, aquellos elementos que están directamente a continuación del elemento especificado. Mediante el símbolo + del selector hermano adyacente, se pueden seleccionar aquellos elementos hermanos que están seguidos el uno de otro (en el mismo nivel)

CONQUERBLOCKS

Selector de hermano adyacente

Ojo, a + b, selecciona al elemento b no al a

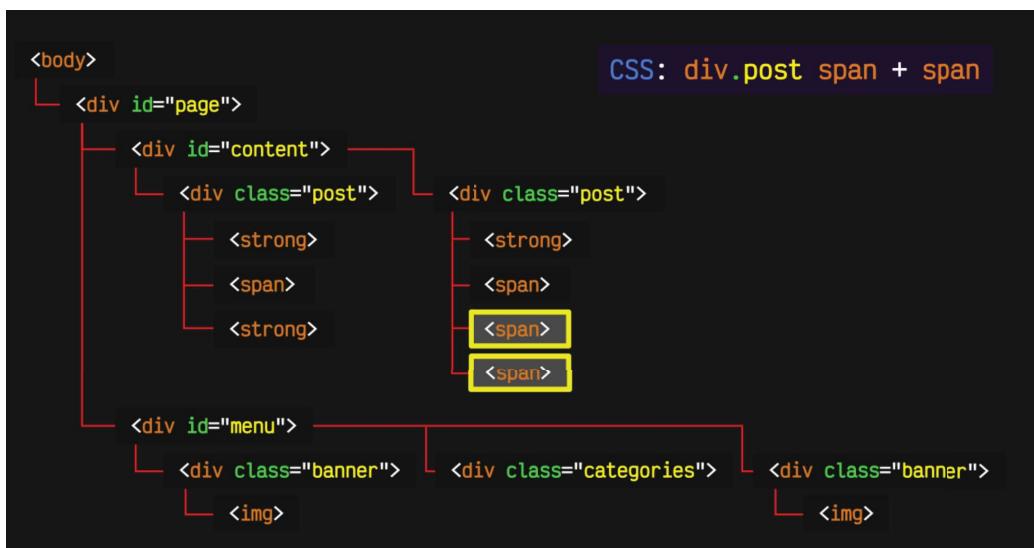
CONQUERBLOCKS

Selector de hermano adyacente

```
div.post span + span {  
    background-color: gold;  
}
```

CONQUERBLOCKS

Selector de hermano adyacente



CONQUERBLOCKS

Selector de hermano adyacente

Observa que el primer elemento `` no es seleccionado, puesto que es el que estamos tomando de referencia de base. Una buena forma para entenderlo es leerlo de la siguiente forma: «todo elemento `` que esté inmediatamente precedido de un ``».

CONQUERBLOCKS

Selector de hermano adyacente

DEMO

CONQUERBLOCKS

Selector de hermanos

CONQUERBLOCKS

Selector de hermanos

Ya sabemos lo que es un hermano :)

CONQUERBLOCKS

Selector de hermanos

Si pensamos otras opciones en el ejemplo anterior, es posible que necesitemos ser menos específicos y en lugar de querer seleccionar los elementos hermanos que sean adyacentes, queramos seleccionar todos los hermanos en general, sin necesidad de que sean adyacentes. Esto se puede conseguir con el selector hermano general, simbolizado con el carácter ~

CONQUERBLOCKS

Selector de hermanos

a ~ b

CONQUERBLOCKS

Selector de hermanos



CONQUERBLOCKS

Selector de hermanos

Como se ve en el ejemplo, no es necesario que el elemento strong se encuentre adyacente al primero, sino que basta con que esté a continuación y sean hermanos en el mismo nivel.

CONQUERBLOCKS

Selector de hermanos

DEMO

CONQUERBLOCKS

Selector múltiple

CONQUERBLOCKS

Selector múltiple

Cuando queremos aplicar los mismos estilos a un conjunto de elementos

CONQUERBLOCKS

Selector múltiple

a, b, c

CONQUERBLOCKS

Selector múltiple

```
.container-logo, .container-alert, .container-warning {  
    border-color: white;  
    background: red;  
}
```

CONQUERBLOCKS

Selector múltiple

Podremos combinar todos los tipos de selectores que hemos visto

CONQUERBLOCKS

Selector múltiple

DEMO

CONQUERBLOCKS

Selector universal (II)

CONQUERBLOCKS

Selector múltiple

Por último, el selector universal se simboliza con un asterisco * y es la forma de aplicar ciertos estilos en TODOS Y CADA UNO de los elementos HTML correspondientes.

CONQUERBLOCKS

Selector múltiple



CONQUERBLOCKS

Selector múltiple

Este ejemplo selecciona todos los elementos dentro de div#menu. Es importante recalcar la diferencia de seleccionar #menu, a seleccionar todos los elementos dentro de #menu, que es lo que estamos haciendo en este caso.

CONQUERBLOCKS

Selector múltiple

Las combinaciones son infinitas

CONQUERBLOCKS

La importancia de un espacio y un punto

CONQUERBLOCKS

Punto y espacio

Un espacio lo cambia todo

CONQUERBLOCKS

Punto y espacio

a .clase

a.clase

a, .clase

CONQUERBLOCKS

Especificidad

CONQUERBLOCKS

Punto y espacio

¿Ahora le veis más sentido a la especificidad verdad?

CONQUERBLOCKS

Ejercicios

CONQUERBLOCKS

Ejercicios

1. Selecciona los elementos div que sean hijos de otro div
2. Selecciona los hijos de los elementos p
3. Selecciona los elementos p que sean hijos de article
4. Selecciona los p que esten dentro de un article
5. Selecciona los titulos h1 y los h3
6. Selecciona los ol que esten dentro de body
7. Cualquier elemento que tenga la clase "amarillo"
8. Los elementos que sean hijos de <body>
9. Los elementos <a> que sean hijos de <p>
10. Selecciona el elemento con id "platano"
11. Todos los elementos <div> que sean hermanos de un elemento <h2>
12. Todos los elementos <p> con clase = paisa, descendientes de un elemento
13. Selecciona los <h1>, <h2>, <h3> que tengan la clase "headertitle", hijos del header

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 07



CONQUERBLOCKS

Clase anterior

Clase anterior

Cascada y prioridad

Selector descendente: doble, triple, etc...

Selector de hijo

Selector de hermano adyacente

Selector de hermanos

Selector múltiple

La importancia de un espacio y un punto

Especificidad

CONQUERBLOCKS

<Índice>

Clase 07

Clase 7: Selectores III

Pseudoclases

Pseudoclases de hijos

Pseudoclases de hijos según tipo de elemento

Pseudoclases de interacción

Pseudoclases de ubicación

Ejercicios

CONQUERBLOCKS

Pseudoclases

CONQUERBLOCKS

Pseudoclases

Las pseudoclases se utilizan para hacer referencia a elementos HTML que tengan un cierto comportamiento concreto

CONQUERBLOCKS

Pseudoclases

Volvamos a recordar el esquema general de sintaxis de CSS, donde ahora añadiremos las pseudoclases, que se definen añadiendo **dos puntos** antes del nombre de la pseudoclase concreta, de la siguiente forma

CONQUERBLOCKS

Pseudoclases



CONQUERBLOCKS

Pseudoclases

De esta forma, podremos seleccionar elementos que en principio parecen iguales, pero tienen diferentes características de comportamiento

CONQUERBLOCKS

Pseudoclases

Para entenderlo bien, vamos a ver las categorías o tipos de pseudoclases y ver cuales se encuentran entre ellas.

CONQUERBLOCKS

Pseudoclases de hijos

CONQUERBLOCKS

Pseudoclases de hijos

Lo usaremos para poder seleccionar cualquier hijo de cualquier elemento sea del tipo que sea

CONQUERBLOCKS

Pseudoclases de hijos

- :first-child
- :last-child

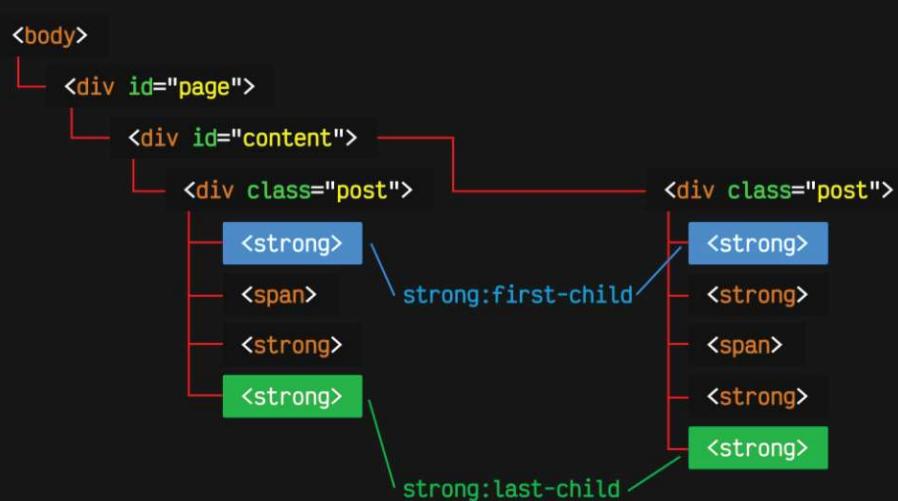
CONQUERBLOCKS

Pseudoclases de hijos

```
strong:first-child {  
    background-color: cyan;  
}  
  
strong:last-child {  
    background-color: green;  
}
```

CONQUERBLOCKS

Pseudoclases de hijos



CONQUERBLOCKS

Pseudoclases de hijos

- :nth-child(n)
- :nth-last-child(n)
- :nth-child(2n)
- :nth-child(2n+1)
- :nth-child(odd): impares
- :nth-child(even): pares

CONQUERBLOCKS

Pseudoclases de hijos

- La pseudoclase :nth-child(A) permite especificar el elemento hijo deseado, simplemente estableciendo su número en el parámetro A. No obstante, hay que tener en cuenta que el parámetro A no es sólo un número, sino que es posible escribir ciertas expresiones:

CONQUERBLOCKS

Pseudoclases de hijos

Selector	Equivalente a...	Significado
strong:nth-child(1)	strong:first-child {	Primer elemento hijo, que además es un
strong:nth-child(2)		Segundo elemento hijo, que además es un
strong:nth-child(3)		Tercer elemento hijo, que además es un
strong:nth-child(n)	strong	Todos los elementos hijos que son
strong:nth-child(2n)		Todos los elementos hijos pares
strong:nth-child(2n-1)		Todos los elementos hijos impares

CONQUERBLOCKS

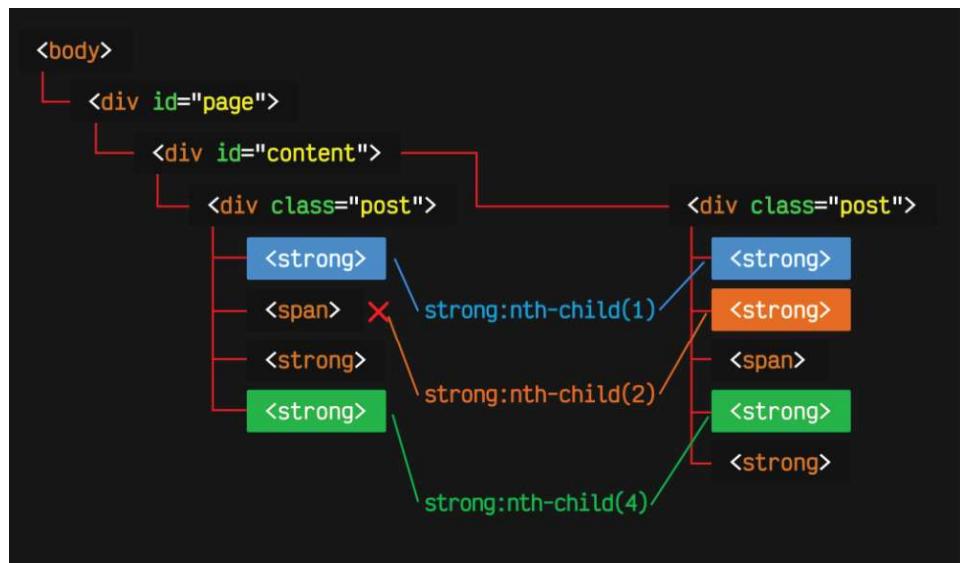
Pseudoclases de hijos

La pseudoclase funcional :nth-last-child(A) funciona de forma muy similar a la anterior, permitiendo también indicarle un parámetro A donde especificar una expresión o número para indicar el hijo concreto.

La diferencia respecto a la anterior, es que comenzamos a contar desde el final

CONQUERBLOCKS

Pseudoclases de hijos



CONQUERBLOCKS

Pseudoclases de hijos

DEMO

CONQUERBLOCKS

Pseudoclases de hijos según tipo de elemento

CONQUERBLOCKS

Pseudoclases de hijos

En los casos anteriores, seleccionamos elementos independientemente de que tipo de elemento sea. Simplemente, hacemos caso a la posición donde está ubicado. Y en algún caso, si no coincide la posición con el tipo de elemento especificado en el selector, simplemente no lo selecciona.

CONQUERBLOCKS

Selector de hijo

Una forma de actuar, quizás, más predecible para nosotros, es que queramos hacer referencia sólo a elementos del mismo tipo, ignorando el resto. Para ello, utilizaremos los selectores siguientes, análogos a los que ya hemos visto, pero haciendo referencia sólo a elementos del mismo tipo

CONQUERBLOCKS

Selector de hijo

- :first-of-type
- :last-of-type

CONQUERBLOCKS

Selector de hijo

Por ejemplo, la pseudoclase :first-of-type es la análoga a :first-child, sólo que tendrá en cuenta sólo elementos de su mismo tipo.

Observa el siguiente ejemplo donde no sólo tenemos <div>, sino que también tenemos un

<p>

CONQUERBLOCKS

Selector de hijo

```
<div class="container">
  <div class="element">Element 1</div>
  <div class="element">Element 2</div>
  <p class="element">Element 3</p>
  <div class="element">Element 4</div>
</div>

<style>
  .container div:first-of-type {
    /* Selecciona "Element 1" */
  }

  .container p:first-of-type {
    /* Selecciona "Element 3" */
  }

  .container :first-of-type {
    /* Selecciona los dos anteriores */
  }
</style>
```

CONQUERBLOCKS

Selector de hijo

- :nth-of-type(n)
- :nth-last-of-type(n)

CONQUERBLOCKS

Selector de hijo

La pseudoclase :nth-of-type(A) es la análoga a :nth-child(A). Se trata de una pseudoclase funcional que admite pasar parámetros, donde le podemos indicar un número (o cierta expresión) para ser mucho más específicos a la hora de seleccionar elementos del mismo tipo.

CONQUERBLOCKS

Selector de hijo

```
<div class="container">
  <div class="element">Element 1</div>
  <div class="element">Element 2</div>
  <p class="element">Element 3</p>
  <div class="element">Element 4</div>
</div>

<style>
  .container :nth-of-type(2) {
    /* Seleccionamos sólo el "Element 2", ya que no hay un segundo <p> */
  }
</style>
```

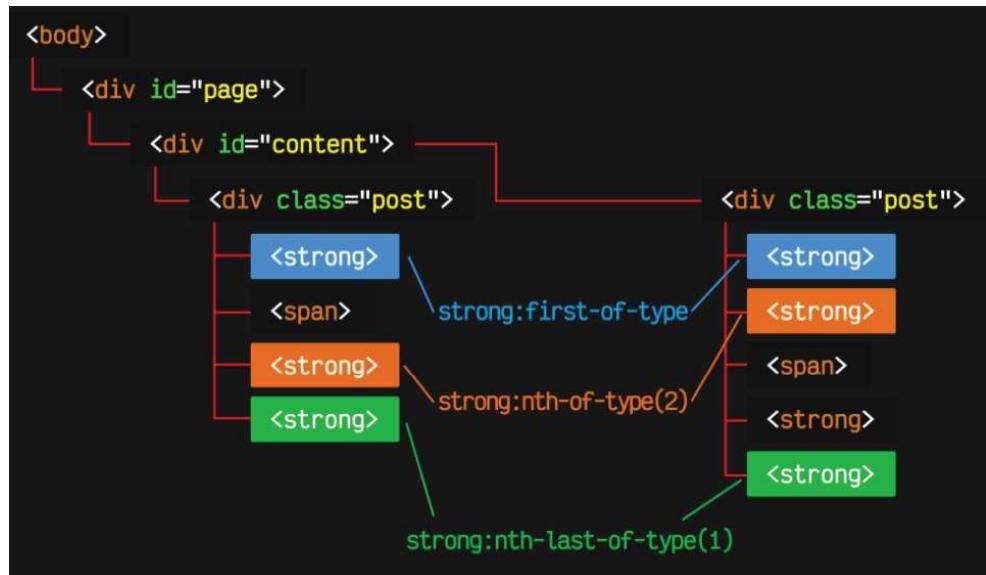
CONQUERBLOCKS

Selector de hijo

La pseudoclase `:nth-last-of-type(A)` es la análoga a `:nth-last-child(A)`. Veamos un nuevo ejemplo sobre el ejercicio anterior, utilizando ahora estas últimas pseudoclases que hemos visto

CONQUERBLOCKS

Selector de hijo



CONQUERBLOCKS

Selector de hijo

En este gráfico, se puede ver como `strong:nth-of-type(2)` selecciona el segundo elemento `strong` en ambos casos, a pesar de que en el primero ocupa la tercera posición. En este caso se selecciona porque es el segundo elemento de su mismo tipo (``).

Por otro lado, `strong:nth-last-of-type(1)` hace una selección de forma inversa, empezando por el último elemento, por lo que elige el último elemento.

CONQUERBLOCKS

Selector de hijo

- :only-child: hijo único de cualquier tipo
- :only-of-type: hijo único de ese tipo concreto
- :empty: elemento sin hijos

CONQUERBLOCKS

Selector de hijo

DEMO

CONQUERBLOCKS

Pseudoclases de interacción

CONQUERBLOCKS

Pseudoclases de interacción

Las pseudoclases de interacción se pueden utilizar en cualquier elemento, aunque lo más frecuente es usarlo en elementos interactivos como enlaces, botones o similares, y pueden seleccionar elementos cuando ocurre una cierta interacción por parte del usuario en ellos.

¿Qué tipo de interacción es esa?

CONQUERBLOCKS

Pseudoclases de interacción

Pseudoclase	Descripción
:hover	Selecciona el elemento si el usuario pasa el ratón sobre dicho elemento.
:active	Selecciona el elemento si el usuario se encuentra pulsando dicho elemento.
:focus	Selecciona el elemento cuando tiene el foco (está en primer plano).
:focus-within	Selecciona el elemento si uno de sus miembros hijos ha ganado el foco.
:focus-visible	Selecciona el elemento cuando tiene el foco sólo de forma visible (TAB , por ejemplo).

CONQUERBLOCKS

Pseudoclases de interacción

hover

La primera de ellas, :hover, es muy útil e interesante, ya que permite aplicar estilos a un elemento justo cuando el usuario pasa el ratón sobre él. Es una de las pseudoclases más utilizadas

CONQUERBLOCKS

Pseudoclases de interacción

```
/* Usuario mueve el ratón sobre un enlace */
a:hover {
    background-color: cyan;
    padding: 2px
}

/* Usuario mueve el ratón sobre un div y resalta todos los enlaces que contiene */
div:hover a {
    background-color: steelblue;
    color: white;
}
```

CONQUERBLOCKS

Pseudoclases de interacción

active

Por otro lado, la segunda pseudoclase, :active, permite resaltar los elementos que se encuentran activos, o lo que es lo mismo, elementos que están siendo pulsados en ese instante con el ratón por el usuario

CONQUERBLOCKS

Pseudoclases de interacción

```
a:active {  
    border: 2px solid #FF0000;  
    padding: 2px  
}
```

CONQUERBLOCKS

Pseudoclases de interacción

focus

Cuando estamos posicionados en un elemento, se dice que ese elemento tiene el foco, mientras que al pulsar TAB y saltar a otro, solemos decir que pierde el foco. También es posible ganar o perder el foco pulsando con el ratón en un elemento.

CONQUERBLOCKS

Pseudoclases de interacción

```
/* El campo ha ganado el foco */
input:focus {
    border: 2px dotted #444
}
```

CONQUERBLOCKS

Pseudoclases de interacción

focus-within

La pseudoclase :focus-within permite darle estilo no sólo al elemento que tiene el foco, sino también a los elementos contenedores relacionados con el elemento que gana el foco.

CONQUERBLOCKS

Pseudoclases de interacción

```
<style>
  form :focus-within {
    background: yellow;
  }
</style>

<form>
  <label>Name: <input type="text"></label>
  <label>Email: <input type="text"></label>
</form>
```

CONQUERBLOCKS

Pseudoclases de interacción

DEMO

CONQUERBLOCKS

Pseudoselectores de ubicación

CONQUERBLOCKS

Pseudoselectores de ubicación

Existen algunas pseudoclases orientadas a los enlaces o hipervínculos. En este caso, permiten cambiar los estilos dependiendo del comportamiento del enlace

CONQUERBLOCKS

Pseudoselectores de ubicación

- :link
- :visited

CONQUERBLOCKS

Pseudoselectores de ubicación

link

Permite seleccionar enlaces a páginas que aún no han sido visitadas por el navegador del usuario, lo que puede ser interesante para personalizar el color de este tipo de enlaces. Por defecto, estos enlaces sin visitar suelen ser de color azul.

CONQUERBLOCKS

Pseudoselectores de ubicación

visited

También tenemos la pseudoclase :visited, que se utiliza para seleccionar y dar estilo a los enlaces que hayan sido visitados previamente en el navegador del usuario. Por defecto, estos enlaces suelen ser de color violeta.

CONQUERBLOCKS

Pseudoselectores de ubicación

DEMO

CONQUERBLOCKS

Ejercicios

CONQUERBLOCKS

Ejercicios Selectores II

1. El primer elemento `<a>` hijo de `<p>`
2. Selecciona los checkbox que estén marcados, hijos de un `<form>`
3. Selecciona el último `<p>` que sea descendiente de `<article>`
4. Seleccionar el segundo hijo de un elemento `section`
5. Seleccionar los hijos pares de un elemento `ol`
6. Seleccionar el antepenúltimo `li` de un `ul` de longitud indefinida
7. El segundo `<a>` de dentro de un `<p>`
8. Los `button` que sean hijos únicos
9. Los `span` que sean primeros hijos de un `p`
10. Penúltimo hijo de `body`

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 08



CONQUERBLOCKS

Clase anterior

Clase anterior

Selectores III

Pseudoclases

Pseudoclases de hijos

Pseudoclases de hijos según tipo de elemento

Pseudoclases de interacción

Pseudoclases de ubicación

CONQUERBLOCKS

<Índice>

Clase 08

Clase 7: Selectores III

Pseudoclases de formularios

Pseudoelementos

Resolución Ejercicios Selectores II

Resolución Ejercicios Selectores III

CONQUERBLOCKS

Pseudoselectores de formulario

CONQUERBLOCKS

Pseudoselectores de formulario

Existe una serie de pseudoclases en CSS que pueden ser utilizadas para usar en formularios de una página web. Estas pseudoclases permiten seleccionar elementos para darle estilo dependiendo de temas relacionados con los formularios o los campos que están en el interior de un formulario.

CONQUERBLOCKS

Pseudoselectores de formulario

- Pseudoclases de **interacción** en formularios: Seleccionar elementos cuando cambia el estado de un elemento.
- Pseudoclases de **estado** en formularios: Seleccionar elementos cuando se encuentran en un estado concreto.
- Pseudoclases de **validación**: Seleccionar elementos si cumplen o no un cierto criterio de validación.

CONQUERBLOCKS

Pseudoselectores de formulario

De interacción en formularios

- :checked: para seleccionar aquellos input de tipo radio o checkbox que estén marcados
- :indeterminate: Selecciona el elemento cuando la casilla está en un estado indeterminado

CONQUERBLOCKS

Pseudoselectores de formulario

```
input:checked + span {  
    color: green;  
}
```

CONQUERBLOCKS

Pseudoselectores de formulario

DEMO

CONQUERBLOCKS

Pseudoselectores de formulario

De estado

Existen varias pseudoclases para detectar si un campo de un formulario está activado o desactivado, o cierta información relacionada con su estado

CONQUERBLOCKS

Pseudoselectores de formulario

De estado

Pseudoclase	Descripción
:enabled	Selecciona cuando el campo del formulario está activado.
:disabled	Selecciona cuando el campo del formulario está desactivado.
:read-only	Selecciona cuando el campo es de sólo lectura.
:read-write	Selecciona cuando el campo es editable por el usuario.
:placeholder-shown	Selecciona cuando el campo está mostrando un placeholder.
:default	Selecciona cuando el elemento tiene el valor por defecto.

CONQUERBLOCKS

Pseudoselectores de formulario

:enabled

Utilizando la autoexplicativa pseudoclase :enabled, podemos seleccionar elementos que se encuentren activados (comportamiento por defecto)

CONQUERBLOCKS

Pseudoselectores de formulario

:disabled

Sin embargo, lo más interesante de este tema viene al poder darle estilo a elementos desactivados con la pseudoclase :disabled, donde se seleccionan los elementos a los que se le ha añadido el atributo disabled

CONQUERBLOCKS

Pseudoselectores de formulario

:read-only

La pseudoclase :read-only selecciona aquellos elementos <input> de un formulario que están marcados con el atributo de sólo lectura readonly. La diferencia entre un campo con atributo disabled y un campo con atributo readonly es que la información del campo con readonly se enviará a través del formulario, mientras que la del campo con disabled no se enviará.

CONQUERBLOCKS

Pseudoselectores de formulario

:read-write

Por otro lado, la pseudoclase :read-write es muy útil para dar estilos a todos aquellos elementos que son editables por el usuario, sean campos de texto <input> o <textarea>.

CONQUERBLOCKS

Pseudoselectores de formulario

:placeholder-show

Con la pseudoclase :placeholder-shown se nos permite seleccionar y dar estilo a los elementos que están actualmente mostrando un placeholder.

CONQUERBLOCKS

Pseudoselectores de formulario

De validación

En HTML5 es posible dotar de capacidades de validación a los campos de un formulario, pudiendo interactuar con ellos desde Javascript o incluso desde CSS. Con estas validaciones podemos asegurarnos de que el usuario escribe en un campo de un formulario el valor esperado.

CONQUERBLOCKS

Pseudoselectores de formulario

De validación

Pseudoclase	¿Cuándo aplica estilos?
:required	Cuando el campo es obligatorio, o sea, tiene el atributo <code>required</code> .
:optional	Cuando el campo es opcional (por defecto, todos los campos).
:valid	Cuando los campos cumplen la validación HTML5.
:invalid	Cuando los campos no cumplen la validación HTML5.
:user-valid	Idem a :valid, pero cuando el usuario ha interactuado.
:user-invalid	Idem a :invalid, pero cuando el usuario ha interactuado.
:in-range	Cuando los campos numéricos están dentro del rango.
:out-of-range	Cuando los campos numéricos están fuera del rango.

CONQUERBLOCKS

Pseudoselectores de formulario

:required

Por lo general, los campos de un formulario son siempre opcionales. Para hacer obligatorio un campo, tenemos que indicar en el elemento HTML el atributo required, al cuál será posible darle estilo mediante la pseudoclase :required:

CONQUERBLOCKS

Pseudoselectores de formulario

:optional

Por otra parte, los camposopcionales son todos aquellos que no tienen el atributo required. Pueden seleccionarse con la pseudoclase :optional

CONQUERBLOCKS

Pseudoselectores de formulario

:valid

Actualmente, desde HTML5 se brinda un excelente soporte de validaciones desde el lado del cliente sin necesidad de Javascript

```
<input type="text" name="age" pattern="[0-9]+"/>
```

CONQUERBLOCKS

Pseudoselectores de formulario

:invalid

De forma opuesta, se pueden seleccionar elementos y aplicar ciertos estilos si no se cumple el patrón de validación

```
<input type="text" name="age" pattern="[0-9]+"/>
```

CONQUERBLOCKS

Pseudoselectores de formulario

:in-range

Lo ideal sería establecer un rango, algo que se suele hacer muy a menudo si tenemos campos numéricos de formulario mediante los atributos min y max

```
<input type="number" name="age" min="18" max="100" />
```

CONQUERBLOCKS

Pseudoselectores de formulario

:out-of-range

Lo contrario de lo anterior

```
<input type="number" name="age" min="18" max="100" />
```

CONQUERBLOCKS

Pseudelementos

CONQUERBLOCKS

Pseudelementos

Los pseudelementos son una característica de CSS que permite hacer referencias a «comportamientos virtuales no tangibles», dicho de otra forma, seleccionar y dar estilo a elementos que no existen en el HTML, o que no son un simple elemento en sí.

CONQUERBLOCKS

Pseudelementos

La sintaxis de los pseudelementos está precedida de dos puntos dobles (::) para diferenciarlos de las pseudoclases, las cuales sólo tienen dos puntos (:).

CONQUERBLOCKS

Pseudelementos

```
selector #id .clase [atributo] :pseudoclasa :: pseudelemento {  
    propiedad : valor ;  
    propiedad : valor  
}
```

CONQUERBLOCKS

Pseudelementos

Elemento	Descripción
content	Propiedad para generar contenido. Sólo usable en ::before o ::after.
::before	Aplica los estilos antes del elemento indicado.
::after	Aplica los estilos después del elemento indicado.
attr()	Función que hace referencia al atributo HTML indicado.

CONQUERBLOCKS

Pseudoselectores de formulario

::content

La propiedad content admite parámetros de diverso tipo, incluso concatenando información mediante espacios. Podemos utilizar varios tipos de contenido

CONQUERBLOCKS

Pseudoselectores de formulario

::content

Valor	Descripción	Ejemplo
STRING "..."	Añade el contenido de texto indicado.	content: "Contenido:";
attr(atributo)	Añade el valor del atributo HTML indicado.	content: attr(href);
IMAGE url(...)	Añade la imagen indicada en la URL.	content: url(icon.png);
GRADIENT	Añade un gradiente del tamaño indicado.	content: linear-gradient(red, blue);
COUNTER	Define un contador CSS para mostrar.	content: counter(item);

CONQUERBLOCKS

Pseudoselectores de formulario

::content

Ten en cuenta que la propiedad content sólo funciona dentro de los pseudoelementos ::before y ::after que explicaremos a continuación. No puede utilizarse en otro lugar.

CONQUERBLOCKS

Pseudoselectores de formulario

::before / ::after

Los pseudoelementos ::before y ::after permiten hacer referencia a «justo antes del elemento» y «justo después del elemento», respectivamente.

Además de selectores básicos como clase o id, combinadores, atributos o pseudoclases, puedes añadir un pseudoelemento, precedido por doble ::

CONQUERBLOCKS

Pseudoselectores de formulario

```
q::before {  
    content: "<";  
    color: #888;  
}
```

```
q::after {  
    content: ">";  
    color: #888;  
}
```

CONQUERBLOCKS

Pseudoselectores de formulario

Función attr()

Es interesante recalcar la utilidad de la función CSS attr(A). Esta función se puede utilizar en la propiedad content para recuperar el valor del atributo HTML especificado en A. Veamos un ejemplo para clarificarlo, concatenándolo con un texto aplicado en la propiedad content

CONQUERBLOCKS

Pseudoselectores de formulario

Función attr()

```
a::after {  
    content: " (" attr(href) " )";  
}
```

CONQUERBLOCKS

Pseudoselectores de formulario

Este pequeño ejemplo de código CSS, realiza una búsqueda de elementos que sean enlaces `<a>`, y posteriormente (`::after`) muestra un contenido de texto posterior donde, envuelto entre paréntesis, escribe el contenido del atributo `href` del enlace tratado.

CONQUERBLOCKS

Pseudoelementos de resaltado

CONQUERBLOCKS

Pseudelementos de resaltado

Existe una serie de pseudelementos orientados a la selección o resaltado de texto en un documento HTML mostrado a través de un navegador

CONQUERBLOCKS

Pseudelementos de resaltado

Pseudelemento	Descripción
::selection	Aplica estilos al fragmento de texto seleccionado por el usuario.
::target-text	Aplica estilos al fragmento de texto enlazado tras el ancla de la URL.
::spelling-error	Aplica estilos al fragmento de texto resaltado por error ortográfico.
::grammar-error	Aplica estilos al fragmento de texto resaltado por error gramatical.

CONQUERBLOCKS

Pseudoselectores de resaltado

::selection

Cuando seleccionamos un fragmento de texto, el navegador suele aplicar un color de fondo que depende del sistema operativo, del tema, o similar.

CONQUERBLOCKS

Pseudoselectores de resaltado

```
::selection {  
    background: indigo;  
    color: white;  
}
```

CONQUERBLOCKS

Pseudelementos tipográficos

CONQUERBLOCKS

Pseudelementos tipográficos

Aunque ::before y ::after suelen ser los ejemplos de pseudelementos más frecuentes, existen muchos otros pseudelementos. Por ejemplo, dentro de una categoría de pseudelementos tipográficos, podemos encontrar los pseudelementos ::first-letter o ::first-line:

CONQUERBLOCKS

Pseudelementos tipográficos

Pseudelemento Descripción

::first-letter	Aplica los estilos en la primera letra del texto.
::first-line	Aplica los estilos en la primera línea del texto.

CONQUERBLOCKS

Pseudelementos tipográficos

```
p {  
    font-family: 'Open Sans', sans-serif;  
    font-size: 1rem;  
}  
  
p::first-letter {  
    font-family: 'Fancy Decorative', serif;  
    font-size: 4rem;  
}
```

CONQUERBLOCKS

Pseudelementos tipográficos

Esto es un minúsculo texto de ejemplo,
para mostrar las ventajas del CSS en la
maquetación de texto.

`::FIRST-Letter`

`::FIRST-LINE`

CONQUERBLOCKS

Ejercicios

CONQUERBLOCKS

Ejercicios Selectores II

1. Selecciona los elementos div que sean hijos de otro div
2. Selecciona los hijos de los elementos p
3. Selecciona los elementos p que sean hijos de article
4. Selecciona los p que esten dentro de un article
5. Selecciona los titulos h1 y los h3
6. Selecciona los ol que esten dentro de body
7. Cualquier elemento que tenga la clase "amarillo"
8. Los elementos que sean hijos de <body>
9. Los elementos <a> que sean hijos de <p>
10. Selecciona el elemento con id "platano"
11. Todos los elementos <div> que sean hermanos de un elemento <h2>
12. Todos los elementos <p> con clase = paisa, descendientes de un elemento
13. Selecciona los <h1>, <h2>, <h3> que tengan la clase "headertitle", hijos del header

Ejercicios Selectores III

1. El primer elemento <a> hijo de <p>
2. Selecciona los checkbox que estén marcados, hijos de un <form>
3. Selecciona el último <p> que sea descendiente de <article>
4. Seleccionar el segundo hijo de un elemento section
5. Seleccionar los hijos pares de un elemento ol
6. Seleccionar el antepenúltimo li de un ul de longitud indefinida
7. El segundo <a> de dentro de un <p>
8. Los button que sean hijos únicos
9. Los span que sean primeros hijos de un p
10. Penúltimo hijo de body

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 10



CONQUERBLOCKS

<Índice>

Clase 10

Clase 10: Texto en CSS

Tipografías

Font-family

Font-size, Font-style

font-weight

Line-height

text-indent

word spacing

CONQUERBLOCKS

Tipografías

CONQUERBLOCKS

Tipografías

- Las tipografías (también denominadas fuentes) son una parte muy importante del mundo de CSS.
- Son uno de los pilares del diseño web.
- La elección de una tipografía adecuada, su tamaño, color, espacio entre letras, interlineado y otras características pueden dotar de una parte fundamental en la percepción en la que una persona interpreta o accede a los contenidos de una página.

CONQUERBLOCKS

Tipografías

¿Qué es una tipografía?
Matemática pura

CONQUERBLOCKS

Tipografías

Serifas



CONQUERBLOCKS

Tipografías

Monoespacio



CONQUERBLOCKS

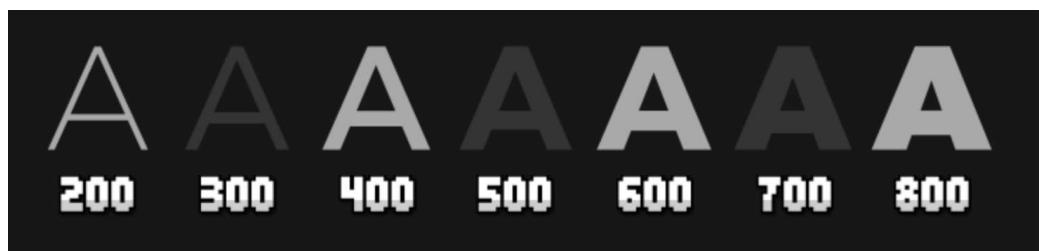
Tipografías

¿Cómo se carga una tipografía?

CONQUERBLOCKS

Tipografías

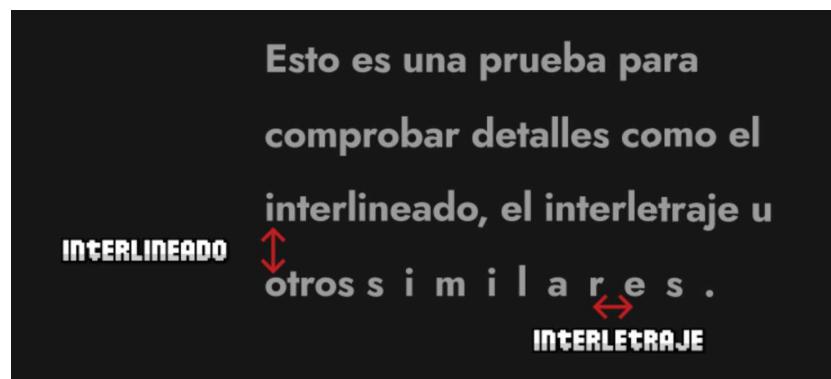
Grosor



CONQUERBLOCKS

Tipografías

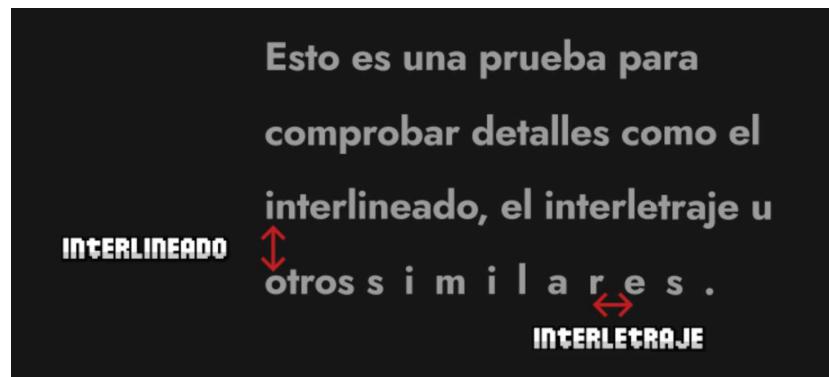
Interlineado e Interletrado



CONQUERBLOCKS

Tipografías

Interlineado e Interletrado



CONQUERBLOCKS

Tipografías

¿Misma tipo para todo?

Elección de parejas

CONQUERBLOCKS

font-family

CONQUERBLOCKS

font-family

La propiedad CSS para seleccionar una familia tipográfica concreta. Con esta propiedad, denominada font-family , podemos seleccionar una tipografía específica simplemente escribiendo su nombre

Cuidado con los espacios

CONQUERBLOCKS

Pseudoselectores de formulario

```
.container {  
    font-family: Verdana;  
    font-family: 'PT Sans';  
}
```

CONQUERBLOCKS

font-family

Los navegadores sólo son capaces de dibujar tipografías que reconoce el sistema o el navegador, es decir, que estén instaladas en el sistema o precargadas por el navegador. En el caso de no cumplir uno de estos casos, el texto se mostrará con otra tipografía «suplente» del sistema, pero que en muchos casos será considerablemente diferente y probablemente no quede igual.

CONQUERBLOCKS

font-family

Esto convierte una tarea a priori simple, en algo muy complejo, puesto que los sistemas operativos (Windows, Mac, GNU/Linux) tienen diferentes tipografías instaladas por defecto y no coinciden. Si además entramos en temas de licencias y tipografías propietarias, la cosa se vuelve aún más compleja.

CONQUERBLOCKS

font-family

Aunque en el ejemplo anterior hemos utilizado sólo una tipografía, lo recomendable es establecer una lista de tipografías. Un primer y sencillo paso para paliar (en parte) el problema anterior, es añadir varias tipografías separadas por comas, lo que además se considera una buena práctica de CSS

CONQUERBLOCKS

font-family

```
.container {  
    font-family: Vegur, 'PT Sans', Verdana, sans-serif;  
}
```

CONQUERBLOCKS

font-family

Fuentes seguras

Fuente	Significado	Fuentes de ejemplo
serif	Tipografía con serifa	Times New Roman, Georgia...
sans-serif	Tipografía sin serifa	Arial, Verdana, Tahoma...
cursive	Tipografía en cursiva	Sanvito, Corsiva...
fantasy	Tipografía decorativa	Critter, Cottonwood...
monospace	Tipografía monoespaciada	Courier, Courier New...

CONQUERBLOCKS

font-family

Demo

CONQUERBLOCKS

font-size

CONQUERBLOCKS

font-size

Usaremos font-size para establecer el tamaño base de la tipografía a la que aplique el selector

CONQUERBLOCKS

font-size

- **Medidas absolutas:** Son palabras clave como medium (tamaño medio), small (tamaño pequeño) o x-large (tamaño muy grande), entre otros.
- **Medidas relativas:** Palabras clave como smaller o larger, que representan un tamaño un poco más pequeño o grande que el actual.
- **Medida específica:** Simplemente, indicar un tamaño exacto con una unidad determinada como px (píxeles), rem (root em) u otras unidades disponibles en CSS.

CONQUERBLOCKS

font-size

Propiedad	font-size
Valores	tamaño_absoluto tamaño_relativo unidad de medida porcentaje inherit
Se aplica a	Todos los elementos
Valor inicial	medium
Descripción	Establece el tamaño de letra utilizado para el texto

CONQUERBLOCKS

font-size

Demo

CONQUERBLOCKS

font-style

CONQUERBLOCKS

font-style

Se usa para cambiar entre texto normal o cursiva

CONQUERBLOCKS

font-style

Propiedad	Valor	Significado
font-style	normal	Estilo normal. Sin cambios aparentes. Valor por defecto.
font-style	italic	Cursiva. Estilo caracterizado por una ligera inclinación de las letras hacia la derecha.
font-style	oblique	Oblíqua. Idem al anterior, salvo que esta inclinación se realiza de forma artificial.
font-style	oblique ANGLE	Idem al anterior, pero indicando los grados de inclinación de -90deg a 90deg. Por defecto, 14deg.

CONQUERBLOCKS

font-style

Demo

CONQUERBLOCKS

font-weight

CONQUERBLOCKS

font-weight

Lo usaremos para establecer el grosor de la tipografía.

La tipografía debe soportarlo

CONQUERBLOCKS

font-weight

Propiedad	Valor	Significado
font-weight	normal (400) bold (700)	Peso absoluto (predefinido). El valor por defecto es normal.
font-weight	bolder lighter	Peso relativo (más pequeño o más grande).
font-weight	<i>peso</i>	Peso específico (número con el peso concreto).



CONQUERBLOCKS

font-weight

Descripción	Valores posibles
Valores de peso ligero	100 - Thin 200 - Extra Light 300 - Light
Valores de peso medio	400 - Normal 500 - Medium
Valores de peso grueso	600 - Semi Bold 700 - Bold 800 - Extra Bold 900 - Heavy

CONQUERBLOCKS

font-weight

Demo

CONQUERBLOCKS

line-height

CONQUERBLOCKS

line-height

La propiedad line-height permite especificar la altura que tendrá cada línea de texto de un párrafo.

Esto es una característica que puede facilitar muchísimo la lectura, puesto que un interlineado excesivo puede desorientar al lector, mientras que uno insuficiente puede hacer perder al visitante el foco en el texto.

CONQUERBLOCKS

line-height

Demo

CONQUERBLOCKS

letter-spacing

CONQUERBLOCKS

letter-spacing

La propiedad letter-spacing , permite indicar el espacio de separación que hay entre cada letra de un texto, denominado comúnmente interletraje o tracking.

Si utilizamos valores negativos tendremos las letras más unidas, mientras que con valores positivos, las letras se encontrarán más separadas unas de otras.

CONQUERBLOCKS

letter-spacing

Demo

CONQUERBLOCKS

text-indent

CONQUERBLOCKS

text-indent

Por último, la propiedad text-indent establece un tamaño de indentación, o lo que es lo mismo, hace un sangrado, en la primera línea del texto, desplazándolo la longitud especificada hacia la derecha (o izquierda en cantidades negativas).

CONQUERBLOCKS

text-indent

Demo

CONQUERBLOCKS

word-spacing

CONQUERBLOCKS

word-spacing

La propiedad word-spacing permite establecer el espacio que hay entre palabras en un texto determinado. Esto puede facilitar la legibilidad de los textos de una página web y da flexibilidad y control sobre ciertas tipografías que pueden no tener suficiente espacio entre las palabras escritas, ya que sus espacios tienen un tamaño insuficiente.

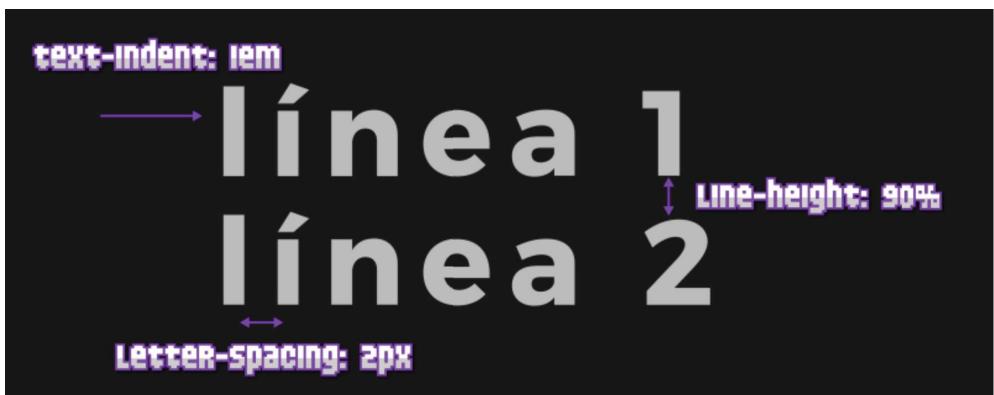
CONQUERBLOCKS

word-spacing

Demo

CONQUERBLOCKS

word-spacing



CONQUERBLOCKS

Ejercicios

CONQUERBLOCKS

Ejercicios Texto I

1. La fuente base de la página debe ser: color negro, tipo Arial, tamaño 0.9em, interlineado 1.4em
2. Los elementos <h2> de .articulo se muestran en color #CC6600, con un tamaño de letra de 1.6em, un interlineado de 1.2
3. Los enlaces del #menu deben ser de color blanco y tamaño de letra 1.3em. Si podéis, ponedle también el color azul al menú
4. El tamaño del texto de todos los contenidos de #lateral debe ser de 0.9em. La fecha de cada noticia debe mostrarse en color gris claro #999.
5. El elemento <h3> de #noticias debe mostrarse de color #003366.
6. El texto del elemento #publicidad es de color gris oscuro #555 y todos los enlaces de color #CC6600.
7. Los enlaces contenidos dentro de .articulo son de color #CC6600

Ejercicios Texto I

Logotipo

Buscar

Lorem Ipsum Dolor Sit Amet

Noticias

dd/mm/aaaa
[Lorem ipsum dolor sit amet](#)
dd/mm/aaaa
[Consectetuer adipiscing elit](#)
dd/mm/aaaa
[Donec molestie nunc eu sapien](#)
dd/mm/aaaa
[Maecenas aliquam dolor eget metus](#)
dd/mm/aaaa
[Fusce fringilla lorem id metus](#)
[Enlaces relacionados](#)
[Pion placera](#)
[Nulla in felis](#)
[Nam luctus](#)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit



Nullam est lacus. suscipit ut, dapibus quis, condimentum ac, risus. Vivamus vestibulum, ipsum sollicitudin faucibus pharetra, dolor metus fringilla dui, vel aliquet pede diam tempor tortor.

Vestibulum pulvinar urna et quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nullam vel turpis vitae dui imperdiet laoreet. Quisque eget ipsum.

Donec commodo, turpis vel venenatis sollicitudin, quam ante convallis justo, sed eleifend justo lectus quis sapien. Ut consequat libero eget est.

[Seguir leyendo...](#)

Phasellus blandit

Praesent sodales imperdiet augue. Mauris lorem felis, semper eu, tincidunt eu, sollicitudin eget, sem. Nulla facilisi. Morbi ut enim ut enim ultricies dapibus.

[Seguir leyendo...](#)

Nullam vel turpis

Donec commodo, turpis vel venenatis sollicitudin, quam ante convallis justo, sed eleifend justo lectus quis sapien. Ut consequat libero eget est.

[Seguir leyendo...](#)

Vivamus lobortis turpis ac ante fringilla faucibus



Quisque eget ipsum. Donec commodo, turpis vel venenatis sollicitudin, quam ante convallis justo, sed eleifend justo lectus quis sapien. Ut consequat libero eget est.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam dictum hendrerit neque. Mauris id ligula non elit mattis semper. Fusce arcu ipsum, tempus eget, tincidunt at, imperdiet in, mi.

Sed fermentum cursus dolor. Aenean a diam. Phasellus feugiat. Donec tempor dignissim sem.

[Seguir leyendo...](#)

[Noticias](#) | [Publicidad](#) | [Contacto](#) | [Sobre Nosotros](#) | [Politica de Privacidad](#)

© Copyright Lorem ipsum

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUER BLOCKS

{CSS}

Clase 11



CONQUERBLOCKS

Clase anterior

Clase anterior

Texto en CSS

Tipografías

Font-family

Font-size, Font-style

font-weight

Line-height

text-indent

word spacing

CONQUERBLOCKS

<Índice>

Clase 11

Clase 11: Texto en CSS 2

Corrección del ejercicio anterior

Espacios en blanco

Límites de línea y palabras

Alineación

Transformaciones

Decoraciones

Ejercicios

CONQUERBLOCKS

Corrección del ejercicio anterior

CONQUERBLOCKS

Ejercicios Texto I

1. La fuente base de la página debe ser: color negro, tipo Arial, tamaño 0.9em, interlineado 1.4em
2. Los elementos <h2> de .articulo se muestran en color #CC6600, con un tamaño de letra de 1.6em, un interlineado de 1.2
3. Los enlaces del #menu deben ser de color blanco y tamaño de letra 1.3em. Si podéis, ponedle también el color azul al menú
4. El tamaño del texto de todos los contenidos de #lateral debe ser de 0.9em. La fecha de cada noticia debe mostrarse en color gris claro #999.
5. El elemento <h3> de #noticias debe mostrarse de color #003366.
6. El texto del elemento #publicidad es de color gris oscuro #555 y todos los enlaces de color #CC6600.
7. Los enlaces contenidos dentro de .articulo son de color #CC6600

Espacios en blanco

Espacios en blanco

Ya hablamos anteriormente del tema de los espacios en blanco entre palabras y etiquetas

CONQUERBLOCKS

Espacios en blanco

```
<p>Hola amigos      ¿Qué tal?</p>
```

CONQUERBLOCKS

Espacios en blanco

Esto se puede modificar de varias formas. La primera y quizás la más antigua y poco adecuada, sería reemplazar cada espacio que se quiere visualizar en el navegador por la entidad HTML de no break space, es decir el código . Esto representa literalmente un espacio en blanco.

CONQUERBLOCKS

Espacios en blanco

Sin embargo, no es la mejor solución, ya que puede ensuciar bastante el código HTML y no es demasiado semántico.

Existe una propiedad CSS denominada white-space, que permite modificar el comportamiento de los espacios en blanco en el HTML a la hora de renderizarlos.

CONQUERBLOCKS

Espacios en blanco

Propiedad	Valor	Significado
white-space	normal nowrap pre pre-line pre-wrap	Comportamiento de los espacios
tab-size	NUMBER SIZE	Ancho de las tabulaciones (espacio o tamaño)

CONQUERBLOCKS

Espacios en blanco

Valor	Espacios en blanco consecutivos	Contenido
normal	Los espacios consecutivos se transforman en uno solo.	Se ajusta al contenedor.
nowrap	Los espacios consecutivos se transforman en uno solo.	Ignora saltos de línea.
pre	Respeta y muestra literalmente los espacios.	Ignora saltos de línea.
pre-wrap	Respeta y muestra literalmente los espacios.	Se ajusta al contenedor.
pre-line	Respeta literalmente los espacios y suprime los espacios del final.	Se ajusta al contenedor.

CONQUERBLOCKS

Espacios en blanco

Por otro lado, tenemos la propiedad **tab-size** que permite establecer el número de espacios que se mostrarán en el cliente o navegador al representar el carácter de un TAB (tabulador), que generalmente se convierten a espacios en blanco, sin embargo son visibles en elementos HTML como <textarea> o <pre>.

CONQUERBLOCKS

Espacios en blanco

Demo

CONQUERBLOCKS

Límites de línea y palabras

CONQUERBLOCKS

Límites de línea y palabras

Existen ciertas situaciones donde palabras o líneas de texto demasiado largas no caben en su contenedor, y se debe tomar una decisión de como se comportará el navegador al renderizar ese contenido: desbordar el contenedor, desplazar a un lugar donde haya más espacio o partir/dividir el contenido.

CONQUERBLOCKS

Límites de línea y palabras

Si te encuentras con la desagradable situación en la que un texto concreto (por ejemplo, un enlace demasiado largo) no cabe dentro de un contenedor, verás que el texto puede desbordarse y provocar efectos no deseados como salirse de su lugar

CONQUERBLOCKS

Límites de línea y palabras

Propiedad	Valor	Significado
word-break	normal keep-all break-all break-word	Indica si se pueden partir palabras de forma natural.
line-break	auto loose normal strict anywhere	Determina como dividir líneas.
hyphens	manual none auto	Indica si se debe dividir las palabras por guiones.
overflow-wrap	normal break-word anywhere	Indica si puede forzar partir palabras y evitar desbordamiento.

La propiedad overflow-wrap sólo funciona cuando white-space está establecida a valores que respeten los espacios. La propiedad word-wrap es la antigua forma de referirse a overflow-wrap.

CONQUERBLOCKS

Límites de línea y palabras

Sin embargo, no es la mejor solución, ya que puede ensuciar bastante el código HTML y no es demasiado semántico.

Existe una propiedad CSS denominada white-space, que permite modificar el comportamiento de los espacios en blanco en el HTML a la hora de renderizarlos.

CONQUERBLOCKS

Límites de línea y palabras

Demo

CONQUERBLOCKS

Alineación

CONQUERBLOCKS

Alineación

Cuando estamos escribiendo mucho texto en nuestros elementos, es posible que necesitemos alinear los textos, para colocarlos de una forma específica: texto a la izquierda, texto a la derecha, texto centrado, etc...

CONQUERBLOCKS

Alineación

Propiedad	Valor	Significado
text-align	<code>start end justify center match-parent justify-all</code>	Alineación del texto.
text-align-last	<code>auto start end justify center match-parent</code>	Alineación de última línea del texto.
text-justify	<code>auto inter-word inter-character none</code>	Método de justificación de textos en <code>justify</code> .
vertical-align	<code>baseline sub super top middle bottom</code> <code>text-top text-bottom</code>	Alineado de textos respecto a elementos.

CONQUERBLOCKS

Alineación text-align y text-align-last

Valores	Descripción
<code>start</code>	Alinea el texto al principio. También se puede usar <code>left</code> .
<code>end</code>	Alinea el texto al final. También se puede usar <code>right</code> .
<code>center</code>	Alinea el texto en el centro.
<code>justify</code>	Justifica el texto, es decir, procura que ocupe toda la línea.
<code>match-parent</code>	Utiliza la alineación establecida en el elemento padre.
<code>justify-all</code>	Usa <code>justify</code> en las propiedades <code>text-align</code> y en <code>text-align-last</code> .

CONQUERBLOCKS

Alineación

vertical-align

Valor	¿Cómo hace la alineación?
baseline	La base del elemento con la base del elemento padre.
sub	El elemento como un subíndice.
super	El elemento como un superíndice.
top	La parte superior del elemento con la parte superior del elemento más alto de la línea.
middle	El elemento en la mitad del elemento padre.
bottom	La parte inferior del elemento con la parte inferior del elemento más bajo de esa línea.
text-top	La parte superior del elemento con la parte superior del texto padre.
text-bottom	La parte inferior del elemento con la parte inferior del texto padre.
SIZE	Sube o baja un elemento el tamaño o porcentaje especificado.

CONQUERBLOCKS

Límites de línea y palabras

Demo

CONQUERBLOCKS

Transformaciones

CONQUERBLOCKS

Transformaciones

En CSS es posible realizar transformaciones a ciertos textos, donde a pesar de tener un texto original en el HTML, queremos mediante CSS realizar una transformación y ponerlo todo en mayúsculas, todo en minúsculas o transformaciones similares sin tener que modificar el contenido HTML.

CONQUERBLOCKS

Transformaciones

Propiedad	Significado
text-transform	Convierte un texto, transformándolo a una variación (minúsculas, mayúsculas, capitalizado, etc...)

Valor	Descripción
none	No realiza ninguna transformación. Valor por defecto.
capitalize	Convierte a mayúsculas la primera letra de cada palabra.
uppercase	Convierte a mayúsculas todas las letras.
lowercase	Convierte a minúsculas todas las letras.

CONQUERBLOCKS

Transformaciones

Demo

CONQUERBLOCKS

Decoraciones

CONQUERBLOCKS

Decoraciones

En CSS es posible realizar decoraciones a ciertos elementos. Por ejemplo, ¿has visto que los enlaces por defecto siempre tienen un borde azul por debajo?

CONQUERBLOCKS

Decoraciones

Primera propiedad shorthand

Propiedad	Valor	Significado
text-decoration-line	none underline overline line-through	Indica el tipo de decoración de texto.
text-decoration-style	solid double dotted dashed wavy	Trazo de la decoración.
text-decoration-color	currentcolor COLOR	Indica el color de la decoración.
text-decoration-thickness	auto from-font SIZE	Indica el grosor del trazo de la decoración.
text-underline-position	auto from-font under	Indica donde aparece el trazo del subrayado.
text-underline-offset	auto SIZE	Indica el desplazamiento del trazo del subrayado.
text-decoration	LINE THICKNESS STYLE COLOR	Propiedad de atajo de las anteriores.

CONQUERBLOCKS

Decoraciones

Demo

CONQUERBLOCKS

Ejercicios

CONQUERBLOCKS

CURRICULUM VITAE DE BRUCE WAYNE

DATOS PERSONALES

- Nombre completo: Bruce Wayne
- Fecha de nacimiento: 1/5/1939
- Lugar de nacimiento: Gotham City

FORMACIÓN ACADÉMICA

- 1956-1961: Universidad del Espantapájaros
- 1952-1956: Instituto de Dos Caras
- 1944-1952: Escuela Primaria del Joker

EXPERIENCIA LABORAL

- 1975-1985: En el paro
- 1965-1975: Cazavillanos y demás chusma
- 1962-1965: Aprendiz de superhéroe

Ejercicios Texto 3

A partir de la página web que se te proporciona, debes crear una página web que tenga el mismo aspecto que la siguiente imagen sin modificar el html:

- El encabezado de nivel 1 debe aparecer centrado y en mayúsculas.
- El encabezado de nivel 2 debe aparecer en mayúsculas.
- El contenido etiquetado como debe aparecer subrayado.

CURRICULUM VITAE DE BRUCE WAYNE

DATOS PERSONALES

- Nombre completo: Bruce Wayne
- Fecha de nacimiento: 15/5/1939
- Lugar de nacimiento: Gotham City

FORMACIÓN ACADÉMICA

- 1956-1961: Universidad del Espantapájaros
- 1952-1956: Instituto de Dos Caras
- 1944-1952: Escuela Primaria del Joker

EXPERIENCIA LABORAL

- 1975-1985: En el paro
- 1965-1975: Cazavillanos y demás chusma
- 1962-1965: Aprendiz de superhéroe

espaciado entre letras debe ser 0.5 em y el espaciado entre palabras 1.5 em.

Además, el encabezado de nivel 1 debe aparecer centrado y en mayúsculas.

- El tipo de letra del encabezado de nivel 2 es igual que el encabezado de nivel 1. El color del texto debe ser #632423. El tamaño del texto debe ser 18 px. El espaciado entre letras debe ser 0.5 em. Además, el encabezado de nivel 2 debe aparecer en mayúsculas.
- El contenido de las listas debe aparecer en cursiva y en negrita. Sin embargo, el contenido etiquetado como no debe aparecer en negrita, pero debe aparecer en versalitas.

C U R R I C U L U M V I T A E D E B R U C E W A Y N E

D A T O S P E R S O N A L E S

- Nombre completo: Bruce Wayne
- Fecha de nacimiento: 15/5/1939
- Lugar de nacimiento: Gotham City

F O R M A C I Ó N A C A D É M I C A

- 1956-1961: Universidad del Espantapájaros
- 1952-1956: Instituto de Dos Caras
- 1944-1952: Escuela Primaria del Joker

E X P E R I E N C I A L A B O R A L

- 1975-1985: En el paro
- 1965-1975: Cazavillanos y demás chusma
- 1962-1965: Aprendiz de superhéroe

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 11



CONQUERBLOCKS

Clase anterior

Clase anterior

Texto en CSS

Tipografías

Font-family

Font-size, Font-style

font-weight

Line-height

text-indent

word spacing

CONQUERBLOCKS

<Índice>

Clase 11

Clase 11: Tipografías en CSS

Tipografías de sistema

Uso de tipografías Custom

Conversor de tipografías

Tipografías externas: Google Fonts

Uso correcto de Google Fonts

CONQUERBLOCKS

Corrección de los ejercicios anteriores

CONQUERBLOCKS

Ejercicios Texto 2

A partir de la página web que se te proporciona, debes crear una página web que tenga el mismo aspecto que la siguiente imagen sin modificar el html:

- El tipo de letra de la página es Arial, con el tipo de respaldo Verdana.
- El encabezado de nivel 1 utiliza el tipo de letra Georgia, con un tamaño de letra de 32 px y en versalitas.
- El encabezado de nivel 2 utiliza el tipo de letra Georgia, con un tamaño de letra de 24 px, en cursiva y en versalitas.
- Los títulos de los elementos de las listas aparecen en negrita.
- El contenido de los elementos de las listas, aunque están etiquetados como , no se visualizan en negrita.

CURRICULUM VITAE DE BRUCE WAYNE

DATOS PERSONALES

- Nombre completo: Bruce Wayne
- Fecha de nacimiento: 1/5/1939
- Lugar de nacimiento: Gotham City

FORMACIÓN ACADÉMICA

- 1956-1961: Universidad del Espantapájaros
- 1952-1956: Instituto de Dos Caras
- 1944-1952: Escuela Primaria del Joker

EXPERIENCIA LABORAL

- 1975-1985: En el paro
- 1965-1975: Cazavillanos y demás chusma
- 1962-1965: Aprendiz de superhéroe

Ejercicios Texto 3

A partir de la página web que se te proporciona, debes crear una página web que tenga el mismo aspecto que la siguiente imagen sin modificar el html:

- El encabezado de nivel 1 debe aparecer centrado y en mayúsculas.
- El encabezado de nivel 2 debe aparecer en mayúsculas.
- El contenido etiquetado como debe aparecer subrayado.

CURRICULUM VITAE DE BRUCE WAYNE

DATOS PERSONALES

- Nombre completo: Bruce Wayne
- Fecha de nacimiento: 1/5/1939
- Lugar de nacimiento: Gotham City

FORMACIÓN ACADÉMICA

- 1956-1961: Universidad del Espantapájaros
- 1952-1956: Instituto de Dos Caras
- 1944-1952: Escuela Primaria del Joker

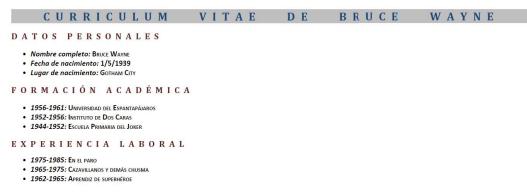
EXPERIENCIA LABORAL

- 1975-1985: En el paro
- 1965-1975: Cazavillanos y demás chusma
- 1962-1965: Aprendiz de superhéroe

Ejercicios Texto 4

A partir de la página web que se te proporciona, debes crear una página web que tenga el mismo aspecto que la siguiente imagen sin modificar el html:

- El tipo de letra principal del texto debe ser Calibri. Como este tipo de letra no está en todos los ordenadores, se debe definir como alternativa los tipos de letra Arial, Helvetica y el tipo genérico sans-serif.
- El tipo de letra del encabezado de nivel 1 debe ser Cambria. El tipo alternativo debe ser Times junto con el tipo genérico serif. El color del texto debe ser #1F497D y el color del fondo #BFBFBF. El tamaño del texto debe ser 24 px. El espacio entre letras debe ser 0.5 em y el espacio entre palabras 1.5 em. Además, el encabezado de nivel 1 debe aparecer centrado y en mayúsculas.
- El tipo de letra del encabezado de nivel 2 es igual que el encabezado de nivel 1. El color del texto debe ser #632423. El tamaño del texto debe ser 18 px. El espacio entre letras debe ser 0.5 em. Además, el encabezado de nivel 2 debe aparecer en mayúsculas.
- El contenido de las listas debe aparecer en cursiva y en negrita. Sin embargo, el contenido etiquetado como no debe aparecer en negrita, pero debe aparecer en versalitas.



Font

Font

Propiedad font (shorthand)

Todas las propiedades vistas en el tema anterior y algunas otras que aún no hemos visto se pueden resumir en la siguiente sentencia

CONQUERBLOCKS

Font

```
.container {  
    font: <style> <variant> <weight> <stretch> <size/line-height> <family>;  
}
```

Ventajas vs Inconvenientes

```
.container {  
    /* Opción 1 */  
    font: italic normal 400 16px Arial, Verdana, sans-serif;  
  
    /* Opción 2 */  
    font: italic normal 400 normal 16px/22px Arial, Verdana, sans-serif;  
}
```

CONQUERBLOCKS

Font



CONQUERBLOCKS

Tipografías de sistema

CONQUERBLOCKS

Tipografías de sistema

Hay que aclarar un detalle muy importante: los navegadores sólo son capaces de dibujar tipografías que reconoce el sistema o el navegador, es decir, que estén instaladas en el sistema o precargadas por el navegador

CONQUERBLOCKS

Tipografías de sistema

Fuente	Significado	Fuentes de ejemplo
serif	Tipografía con serifa	Times New Roman, Georgia...
sans-serif	Tipografía sin serifa	Arial, Verdana, Tahoma...
cursive	Tipografía en cursiva	Sanvito, Corsiva...
fantasy	Tipografía decorativa	Critter, Cottonwood...
monospace	Tipografía monoespaciada	Courier, Courier New...

CONQUERBLOCKS

Tipografías de sistema

Debemos saber que las tipografías se almacenan en unos ficheros con formatos especiales, como por ejemplo .ttf o .woff2 (hablaremos de ellos más adelante).

CONQUERBLOCKS

Tipografías de sistema

En una página web, salvo casos muy particulares, lo recomendable es que existan entre una y cuatro tipografías diferentes. Por ejemplo, una tipografía para encabezados o titulares, una para párrafos o textos extensos, una para código (si existe), y alguna que otra alternativa o similar.

CONQUERBLOCKS

Tipografías de sistema

Para ver las tipografías correctamente en un navegador, dichas tipografías deben estar instaladas en el sistema. Si no lo están, la tipografía no se verá y el sistema elegirá otra tipografía «suplente».

CONQUERBLOCKS

Uso de tipografías custom

CONQUERBLOCKS

Uso de tipografías custom

```
.container {  
    font-family: Vegur, Georgia, sans-serif;  
}
```

- Un usuario con la tipografía **Vegur** instalada, vería sin problemas el diseño ideal.
- Si no tiene la tipografía Vegur instalada en el sistema, la vería con la segunda tipografía: Georgia.
- Si tampoco tiene la tipografía Georgia instalada, buscará una tipografía de sistema «sin serifa».

CONQUERBLOCKS

Uso de tipografías custom

OJO

Muchas tipografías tienen derechos de autor y no se deberían utilizar en nuestra página web si no hemos pagado su licencia. Antes de utilizar una tipografía, investiga que tipo de licencia tiene o si puedes utilizarla. Si tienes dudas, busca tipografías de dominio público o CC0.

CONQUERBLOCKS

Uso de tipografías custom

@font-face

CONQUERBLOCKS

Uso de tipografías custom

Para solucionar el problema de utilizar una tipografía que el usuario no tiene instalada en su sistema, utilizaremos la regla @font-face de CSS.

Dicha regla nos permite descargar en el navegador una tipografía desde una página y utilizarla aunque no esté instalada en el sistema

CONQUERBLOCKS

Uso de tipografías custom

Observa el siguiente ejemplo de código donde precargamos la tipografía Open Sans, utilizando la regla CSS @font-face. Esta regla suele colocarse al principio del fichero CSS para avisar al navegador que vamos a utilizar una tipografía que es muy posible que no se encuentre instalada.

CONQUERBLOCKS

Uso de tipografías custom

```
@font-face {  
    font-family: 'Open Sans';  
    font-style: normal;  
    font-weight: 400;  
    src:  
        local("Open Sans"),  
        url("/fonts/opensans.woff2") format("woff2"),  
        url("/fonts/opensans.woff") format("woff"),  
        url("/fonts/opensans.ttf") format("truetype");  
}
```

CONQUERBLOCKS

Uso de tipografías custom

font-family
font-style
font-weight
src

Valor	Significado	Soporte
local("Nombre")	¿Está la fuente instalada? Si es así, no hace falta descargarla, la usa.	✓ Todos
url("file.woff2")	Formato Web Open Font Format 2 . Mejora de WOFF con Brotli .	✓ Bueno
url("file.woff")	Formato Web Open Font Format . Es un TTF comprimido, ideal para web.	✓ Bueno
url("file.ttf")	Formato True Type / Open Type . .ttf o .otf. Para soportar navegadores antiguos.	✓ Bueno
url("file.eot")	Formato Embedded OpenType . Mejora de OTF, propietaria de Microsoft.	✗ Sólo IE
url("file.svg")	Tipografías creadas como formas SVG. No usar , considerada obsoleta .	✗ No usar

CONQUERBLOCKS

Uso de tipografías custom

font-face vale para cualquier tipografía

CONQUERBLOCKS

Uso de tipografías custom

¿Cómo conseguimos todos esos formatos?

CONQUERBLOCKS

Conversor de tipografías

CONQUERBLOCKS

Conversor de tipografías

Vamos a buscar una tipografía gratuita

CONQUERBLOCKS

Conversor de tipografías

Vamos a generar todos los formatos necesarios

transfonter

CONQUERBLOCKS

Conversor de tipografías

¿Probamos?

CONQUERBLOCKS

Tipografías externas: Google Fonts

CONQUERBLOCKS

Tipografías externas: Google Fonts

En la actualidad, es muy común utilizar Google Fonts como repositorio proveedor de tipografías para utilizar en nuestros sitios web por varias razones

CONQUERBLOCKS

Tipografías externas: Google Fonts

- **Gratis:** Disponen de un amplio catálogo de fuentes y tipografías libres y/o gratuitas.
- **Cómodo:** Resulta muy sencillo su uso: Google nos proporciona un código y el resto lo hace él.
- **Rápido:** El servicio está muy extendido y utiliza sus propios servidores.

CONQUERBLOCKS

Tipografías externas: Google Fonts

Seleccionar Tipografía

CONQUERBLOCKS

Tipografías externas: Google Fonts

Seleccionar Pesos

CONQUERBLOCKS

Tipografías externas: Google Fonts

Preconexión a Google Fonts

La etiqueta `<link rel="preconnect" href="https://fonts.googleapis.com">` es un fragmento de código para que el navegador haga una preconexión a los subdominios de descarga de las tipografías. Esto hace que el tiempo de descarga sea un poco más corta:

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

CONQUERBLOCKS

Tipografías externas: Google Fonts

Dos formas de incluir las fuentes

- Link
- @import

CONQUERBLOCKS

Tipografías externas: Google Fonts

Existen dos formas de incluir la tipografía en nuestra página. Mediante un fragmento de código HTML, marcando la opción <link>, o mediante un fragmento de código CSS, marcando la opción @import.

Veamos el fragmento de código que nos aparece si marcamos la primera, <link> para añadirlo en nuestro código HTML, más concretamente, antes de cerrar la etiqueta </head>:

CONQUERBLOCKS

Tipografías externas: Google Fonts

Definir la tipografía a usar

```
.menu {  
    font-family: Lato, Roboto, sans-serif;  
    font-weight: 400;  
}
```

CONQUERBLOCKS

Tipografías externas: Google Fonts

Demo

CONQUERBLOCKS

Uso correcto de Google Fonts

CONQUERBLOCKS

Uso correcto de Google Fonts

Ojo con la GDPR
(uso de las IP's)

CONQUERBLOCKS

Uso correcto de Google Fonts

1. Descarga la familia con los pesos seleccionados
2. Usa font-face

CONQUERBLOCKS

Uso correcto de Google Fonts

Demo

CONQUERBLOCKS

Ejercicios

CONQUERBLOCKS

Ejercicios Texto I

Crear un fichero con 6 párrafos y aplicarle los siguientes estilos

- El primer párrafo tiene una tipografía segura Verdana (sans-serif).
- El segundo párrafo tiene una tipografía segura Didot (serif).
- El tercer párrafo tiene una tipografía segura FreeMono (monospace).
- El cuarto párrafo tiene una tipografía importada desde Google Fonts [Roboto](#) [400](#) dentro de la etiqueta style.
- El quinto párrafo tiene una tipografía descargada de Google Fonts Akronim y llamada con la instrucción @font-face. El tamaño de fuente 24px. Podéis descargarla directamente desde Google Fonts.
- El sexto párrafo tiene una tipografía importada desde Google Fonts [Montserrat](#) [400](#) mediante un <link>.



<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

{CSS}

Clase 13



CONQUERBLOCKS

Clase 12: Tipografías en CSS

Tipografías de sistema

Uso de tipografías Custom

Conversor de tipografías

Tipografías externas: Google Fonts

Uso correcto de Google Fonts

Clase anterior

Clase 12

CONQUERBLOCKS

<Índice>

Clase 13

Clase 13: Iconos tipográficos

Corrección de ejercicios anteriores

¿Qué son?

Tipografías de iconos

Propiedades y mejoras VS imágenes

Ejemplos de uso: fontawesome, bootstrap icons...

Icomoon

CONQUERBLOCKS

Corrección de los ejercicios anteriores

CONQUERBLOCKS

Ejercicios Texto 5

Crear un fichero con 6 párrafos y aplicarle los siguientes estilos

- El primer párrafo tiene una tipografía segura Verdana (sans-serif).
- El segundo párrafo tiene una tipografía segura Didot (serif).
- El tercer párrafo tiene una tipografía segura FreeMono (monospace).
- El cuarto párrafo tiene una tipografía importada desde Google Fonts `Roboto`
`400` dentro de la etiqueta style.
- El quinto párrafo tiene una tipografía descargada de Google Fonts Akronim y llamada con la instrucción @font-face. El tamaño de fuente 24px. Podéis descargarla directamente desde Google Fonts.
- El sexto párrafo tiene una tipografía importada desde Google Fonts `Montserrat`
`400` mediante un <link>.

¿Qué son los iconos tipográficos?

¿Qué son los iconos tipográficos?

Uso de imágenes para decoración

CONQUERBLOCKS

Tipografías de iconos

CONQUERBLOCKS

Tipografías de sistema

Recuerda que las tipografías no sólo tienen letras

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

Escalabilidad y Resolución

Las tipografías de iconos son vectoriales, lo que significa que se pueden escalar a cualquier tamaño sin perder calidad. Las imágenes, en cambio, están basadas en píxeles y pueden perder claridad cuando se escalan a tamaños más grandes.

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

Tamaño de archivo

Las tipografías de iconos suelen ser más livianas que las imágenes, ya que solo se trata de fuentes de texto en lugar de archivos de imagen completos. Esto puede mejorar el rendimiento de tu sitio web, ya que las páginas cargarán más rápido.

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

Facilidad de personalización

Las tipografías de iconos a menudo se pueden personalizar utilizando CSS. Puedes cambiar el color, tamaño y otros atributos utilizando propiedades CSS estándar. En cambio, para las imágenes, normalmente tendrías que crear varias versiones con diferentes tamaños y colores.

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

Accesibilidad

Las tipografías de iconos se pueden ajustar fácilmente para cumplir con requisitos de accesibilidad, como el contraste y el tamaño del texto. Además, las fuentes de texto son legibles por lectores de pantalla, lo que mejora la experiencia para usuarios con discapacidades visuales.

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

Facilidad de implementación

Agregar iconos a través de tipografías es bastante sencillo. Solo necesitas cargar la fuente una vez y luego puedes utilizar las clases correspondientes para mostrar los iconos en tu sitio. Por otro lado, trabajar con imágenes puede requerir más pasos y consideraciones, como la optimización para la web.

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

Consistencia

Las tipografías de iconos suelen ofrecer un conjunto coherente de iconos que siguen un mismo estilo visual. Esto ayuda a mantener la coherencia en el diseño de tu sitio.

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

Sin embargo, también hay algunos casos en los que las imágenes pueden ser más apropiadas. Por ejemplo, si necesitas detalles muy precisos en tus iconos o si estás trabajando con ilustraciones complejas, las imágenes pueden ser la mejor opción.

CONQUERBLOCKS

Propiedades y mejoras VS imágenes

En general, usar tipografías de iconos en CSS es una práctica común en el diseño web moderno debido a sus beneficios en términos de rendimiento, personalización y accesibilidad.

CONQUERBLOCKS

Ejemplos de uso

CONQUERBLOCKS

Ejemplos de uso

Flaticon

CONQUERBLOCKS

Ejemplos de uso

Google Material Icon Font

CONQUERBLOCKS

Ejemplos de uso

Fontawesome

CONQUERBLOCKS

Ejemplos de uso

Bootstrap icons

CONQUERBLOCKS

Icomoon

CONQUERBLOCKS

Icomoon

Llevo casi 10 años usándola

CONQUERBLOCKS

Demo

CONQUERBLOCKS

Ejercicios Texto I

Vamos a crear un menú de iconos con cada uno de los tipos de iconos tipográficos que hemos visto en el tema.

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

[@bienvenidosaez](https://www.instagram.com/bienvenidosaez)

Youtube

[youtube.com/bienvenidosaez](https://www.youtube.com/bienvenidosaez)

CONQUER BLOCKS

{CSS}

Clase 14



CONQUERBLOCKS

Clase anterior

Clase 13

Clase 13: Iconos tipográficos

Corrección de ejercicios anteriores

¿Qué son?

Tipografías de iconos

Propiedades y mejoras VS imágenes

Ejemplos de uso: fontawesome, bootstrap icons...

Icomoon

CONQUERBLOCKS

<Índice>

Clase 14

Clase 14: El modelo de caja

¿Qué es el modelo de caja?

Width y Height

Margin, Padding y Border

Box-sizing, Opacity y Overflow

CONQUERBLOCKS

¿Qué es el modelo de caja?

CONQUERBLOCKS

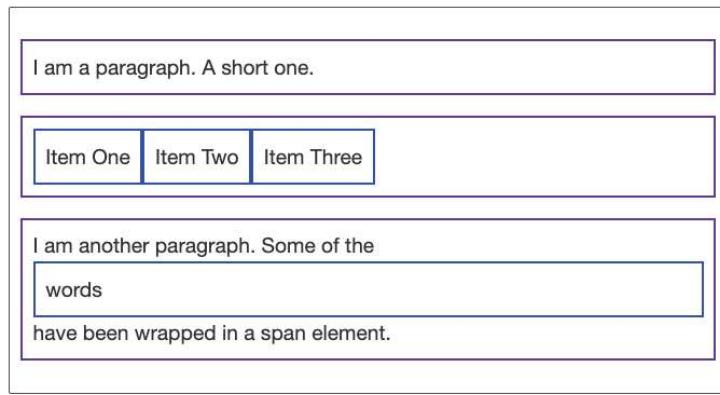
¿Qué es el modelo de caja?

Cuando hablamos del modelo de cajas en CSS, estamos haciendo referencia a un sistema que tiene el navegador de interpretar las diferentes partes de lo que solemos denominar «caja»: un elemento HTML con unas ciertas dimensiones.

CONQUERBLOCKS

¿Qué es el modelo de caja?

Todo en CSS tiene una caja alrededor, y comprender estas cajas es clave para poder crear diseños con CSS o para alinear elementos con otros elementos.



CONQUERBLOCKS

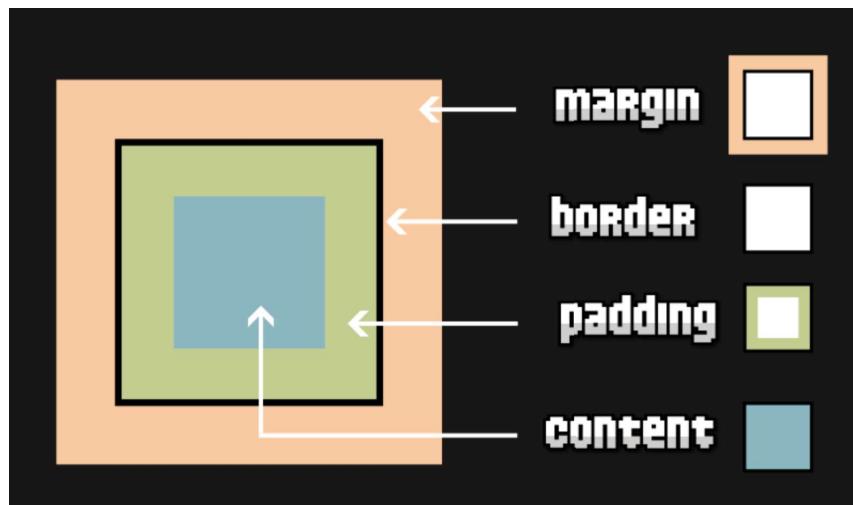
¿Qué es el modelo de caja?

La representación básica del modelo de cajas se basa en varios conceptos importantes, como veremos a continuación:

- El **borde** (border) es el límite que separa el interior del exterior del elemento.
- El **margin** (margin) es la parte exterior del elemento, por fuera del borde.
- El **relleno** (padding) es la parte interior del elemento, entre el contenido y el borde.
- El **contenido**, en azul, es la parte interior del elemento, excluyendo el relleno.

CONQUERBLOCKS

¿Qué es el modelo de caja?



CONQUERBLOCKS

Width

CONQUERBLOCKS

Width

- El ancho de un elemento lo determinará su contenido a no se que se especifique
- Lo usaremos para establecer el ancho de un elemento (o de su caja contenedora)
- No incluye padding, border ni margin
- Admite varios tipos de unidades
- **Sólo funciona con los elementos de bloque**

CONQUERBLOCKS

Width

CSS Syntax

```
width: auto|value|initial|inherit;
```

Property Values

Value	Description	Demo
auto	Default value. The browser calculates the width	Demo >
length	Defines the width in px, cm, etc. Read about length units	Demo >
%	Defines the width in percent of the containing block	Demo >
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

CONQUERBLOCKS

max-width y min-width

CONQUERBLOCKS

max-width y min-width

max-width

none | SIZE

Ancho máximo que puede ocupar un elemento.

min-width

0 | SIZE

Ancho mínimo que puede ocupar un elemento.

CONQUERBLOCKS

Height

CONQUERBLOCKS

Height

- Lo usaremos para establecer el alto de un elemento (o de su caja contenedora)
- No incluye padding, border ni margin
- Admite varios tipos de unidades
- Si no establece, lo establece el contenido
- **Sólo funciona con los elementos de bloque**

CONQUERBLOCKS

Height

CSS Syntax

```
height: auto|length|initial|inherit;
```

Property Values

Value	Description	Demo
auto	The browser calculates the height. This is default	Demo >
length	Defines the height in px, cm, etc. Read about length units	Demo >
%	Defines the height in percent of the containing block	Demo >
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

CONQUERBLOCKS

max-height y min-height

CONQUERBLOCKS

max-height y min-height

max-height	none SIZE	Alto máximo que puede ocupar un elemento.
min-height	0 SIZE	Alto mínimo que puede ocupar un elemento.

CONQUERBLOCKS

DEMO

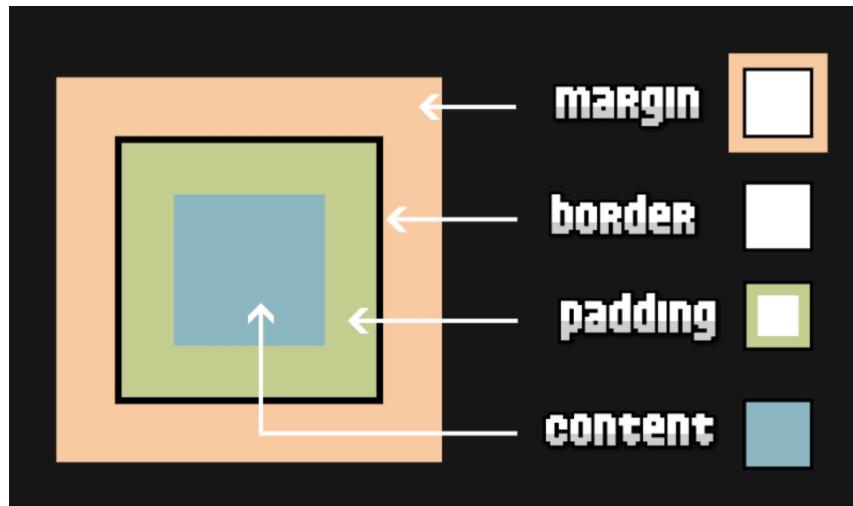
W3c

CONQUERBLOCKS

Margin

CONQUERBLOCKS

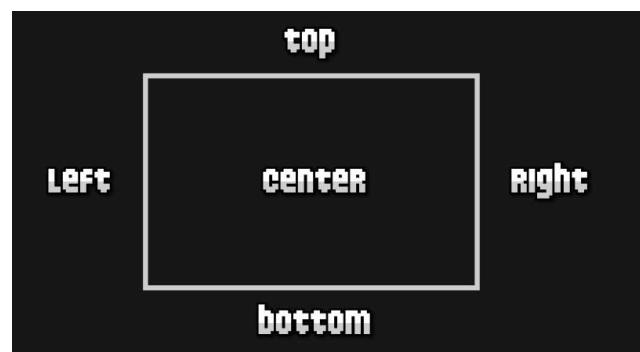
Margin



CONQUERBLOCKS

Margin

Zonas de un elemento



CONQUERBLOCKS

Margin

All CSS Margin Properties

Property	Description
<code>margin</code>	A shorthand property for setting all the margin properties in one declaration
<code>margin-bottom</code>	Sets the bottom margin of an element
<code>margin-left</code>	Sets the left margin of an element
<code>margin-right</code>	Sets the right margin of an element
<code>margin-top</code>	Sets the top margin of an element

CONQUERBLOCKS

Margin

Margin

CONQUERBLOCKS

Margin

DEMO

CONQUERBLOCKS

Margin

margin-collapse

Otro detalle que merece la pena destacar, y que sorprende mucho en el aprendizaje del modelo de cajas de CSS, es el llamado «margin collapse» o colapso de márgenes

CONQUERBLOCKS

Margin

Como veremos más adelante, CSS tiene varios mecanismos de maquetación. Los más básicos son inline y block, y en ellos, ocurre la siguiente situación. Imaginemos que tenemos dos elementos adyacentes, y cada uno de ellos, tiene un margen de 20px definido.

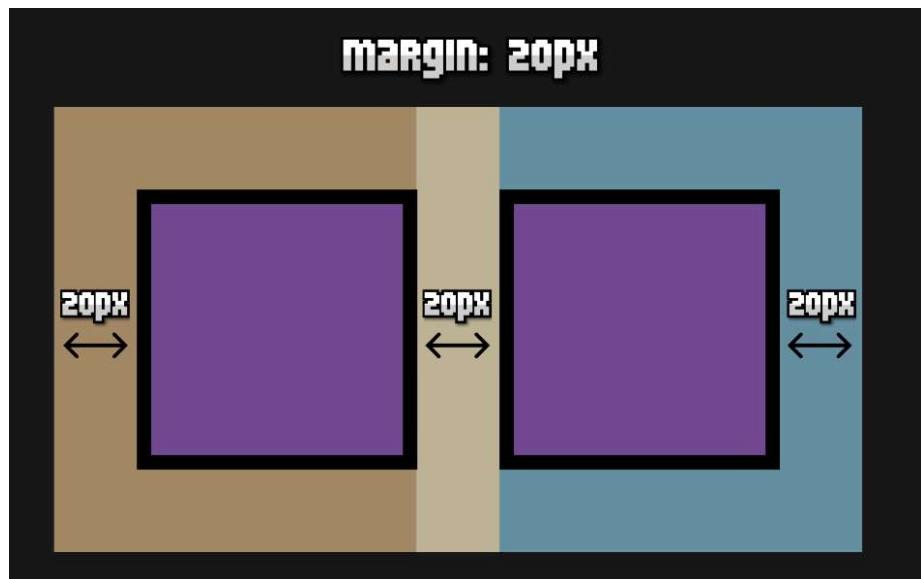
CONQUERBLOCKS

Margin

Quizás, lo más intuitivo es pensar que el elemento de la izquierda tiene 20px de margen derecho, mientras que el elemento de la derecha tiene 20px de margen izquierdo, que al sumarse, predecimos que resultará un tamaño de 40px. Sin embargo, la realidad es distinta:

CONQUERBLOCKS

Margin



CONQUERBLOCKS

Margin

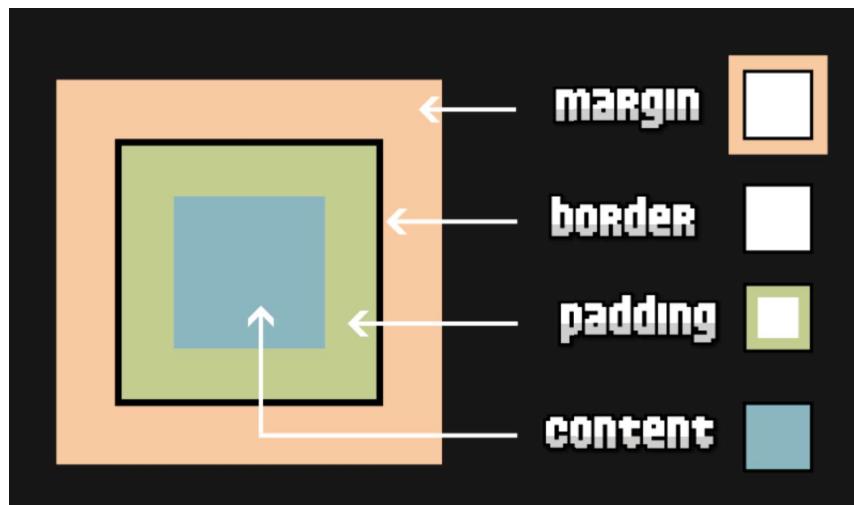
DEMO

CONQUERBLOCKS

Padding

CONQUERBLOCKS

Padding



CONQUERBLOCKS

Padding

Zonas de un elemento



CONQUERBLOCKS

Padding

All CSS Padding Properties

Property	Description
<code>padding</code>	A shorthand property for setting all the padding properties in one declaration
<code>padding-bottom</code>	Sets the bottom padding of an element
<code>padding-left</code>	Sets the left padding of an element
<code>padding-right</code>	Sets the right padding of an element
<code>padding-top</code>	Sets the top padding of an element

CONQUERBLOCKS

Border

Por separado y shorthand

CONQUERBLOCKS

Border

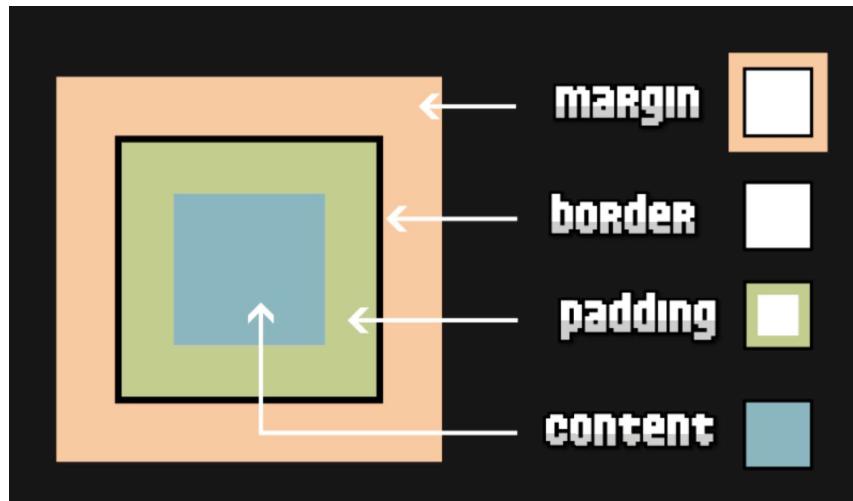
DEMO

CONQUERBLOCKS

Border

CONQUERBLOCKS

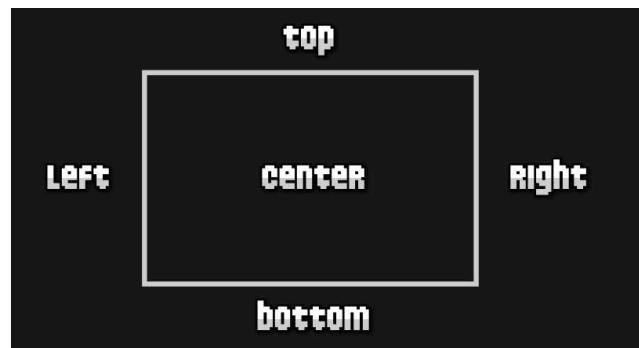
Border



CONQUERBLOCKS

Border

Zonas de un elemento



CONQUERBLOCKS

Border

Tres factores a tener en cuenta con los bordes

- **Tamaño:** `border-width`
- **Estilo:** `border-style`
- **Color:** `border-color`

CONQUERBLOCKS

Border

También se pueden establecer por separado

CONQUERBLOCKS

Border

Propiedad shorthand

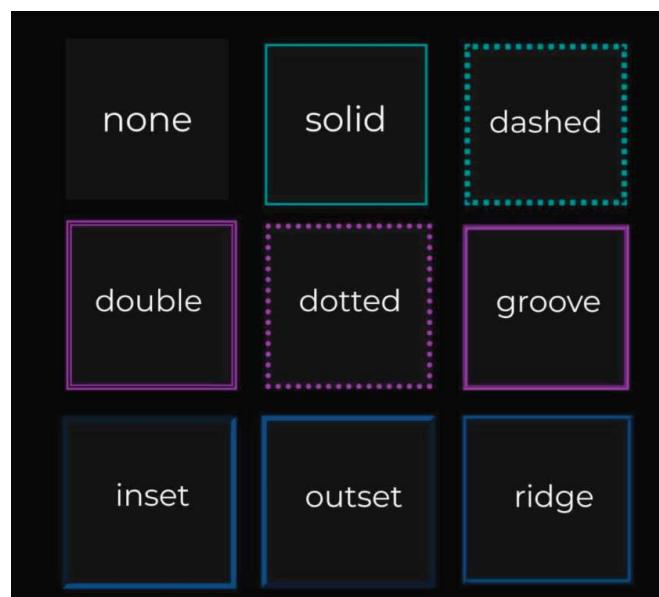
CONQUERBLOCKS

Border

Bordes redondeados

CONQUERBLOCKS

Border



CONQUERBLOCKS

Border

DEMO

CONQUERBLOCKS

Box-sizing

CONQUERBLOCKS

Box-sizing

Por defecto, el modelo de cajas de CSS sigue un esquema donde al dar un tamaño a un elemento, concretamente se le da tamaño a su contenido, de modo que si además le añadimos un borde (border) y/o un relleno (padding), se sumará al tamaño indicado previamente.

CONQUERBLOCKS

Box-sizing

Esto puede resultar algo poco intuitivo, ya que un elemento de tamaño width: 200px, con un border: 15px y un padding: 20px, tendría en realidad un tamaño total de 270px (15px + 15px de borde izquierdo y derecho, 20px + 20px de relleno izquierdo y derecho y los 200px del contenido).

CONQUERBLOCKS

Box-sizing

Tener que realizar estos cálculos, puede resultar complejo y poco predecible, especialmente en situaciones donde estamos muy justos de espacio.

CONQUERBLOCKS

Box-sizing

box-sizing

En CSS, existe una propiedad denominada box-sizing que permite alterar esta versión del modelo de cajas. Lo explicado en el ejemplo anterior es el funcionamiento por defecto del modelo de cajas de CSS, y el valor content-box de la propiedad box-sizing, que obviamente, es el valor por defecto

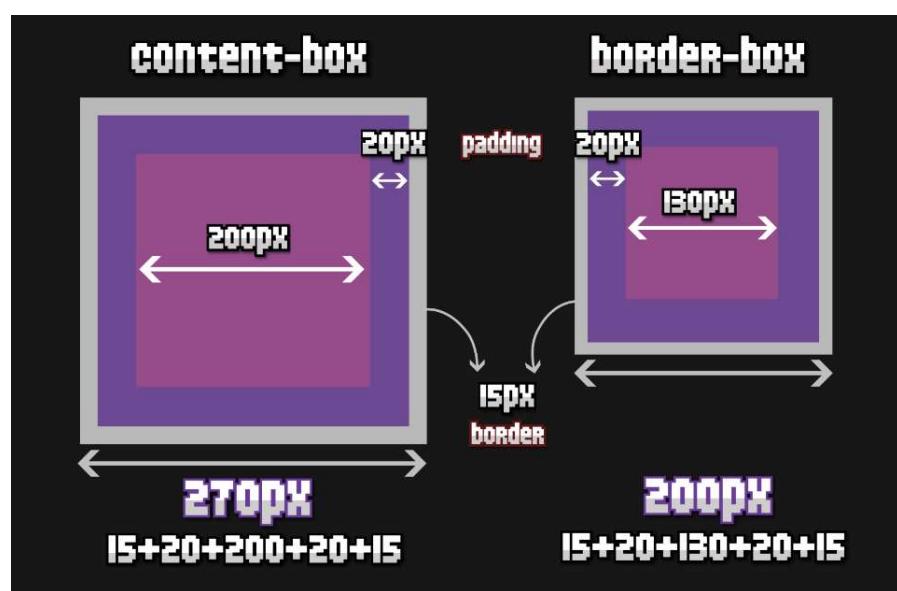
CONQUERBLOCKS

Box-sizing

Valor	Descripción
content-box	Las propiedades <code>width</code> y <code>height</code> no incluyen borde, relleno y/o margen. Valor por defecto.
border-box	Las propiedades <code>width</code> y <code>height</code> incluyen borde y relleno.

CONQUERBLOCKS

Box-sizing



CONQUERBLOCKS

Opacity

CONQUERBLOCKS

Opacity

Por defecto, el modelo de cajas de CSS establece una opacidad para cada caja de valor 1, es decir, totalmente visible.

Este aspecto podrá ser modificado con la propiedad opacity

CONQUERBLOCKS

Opacity

CSS Syntax

```
opacity: number|initial|inherit;
```

Property Values

Value	Description	Demo
<i>number</i>	Specifies the opacity. From 0.0 (fully transparent) to 1.0 (fully opaque)	Demo >
initial	Sets this property to its default value. Read about <i>initial</i>	
inherit	Inherits this property from its parent element. Read about inherit	

CONQUERBLOCKS

Opacity

DEMO

CONQUERBLOCKS

Overflow

CONQUERBLOCKS

Overflow

CSS
IS
AWESOME

CONQUERBLOCKS

Overflow

Volvamos a pensar en la situación de la imagen anterior: Damos un tamaño de ancho y alto a un elemento HTML, pero su contenido de texto es tan grande que no cabe dentro de ese elemento. ¿Qué ocurriría? Probablemente lo que vimos en la imagen: el contenido se desbordaría.

CONQUERBLOCKS

Overflow

Propiedad	Valor	Significado
overflow-x	visible hidden scroll auto	Establece el desbordamiento del eje X (horizontal).
overflow-y	visible hidden scroll auto	Establece el desbordamiento del eje Y (vertical).
overflow	overflow-x overflow-y	Propiedad de atajo que establece desbordamiento de ambos ejes.

CONQUERBLOCKS

Overflow

Valor	¿Qué ocurre si se desborda el contenedor?	¿Desbordamiento?
visible	Se muestra el contenido que sobresale (comportamiento por defecto)	Sí
hidden	Se oculta el contenido que sobresale.	No
scroll	Se colocan barras de desplazamiento (horizontales y verticales).	No
auto	Se colocan barras de desplazamiento (sólo las necesarias).	No

CONQUERBLOCKS

Overflow

DEMO

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 15



CONQUERBLOCKS

Clase 14: Modelo de caja

¿Qué es el modelo de caja?

Width y Height

Margin, Padding y Border

Box-sizing, Opacity y Overflow

Clase anterior

Clase 14

CONQUERBLOCKS

<Índice>

Clase 15

Clase 15: Manejo de imágenes en CSS

Background-image

Background-position, clip y size y demás...

Background

Object-fit

Aspect-ratio

Gradient

Manejo de img

CONQUERBLOCKS

Introducción

CONQUERBLOCKS

Introducción

Diferenciemos entre los dos tipos de imágenes que podemos tener en una web

Imágenes de contenido

Imágenes de decoración

CONQUERBLOCKS

Introducción

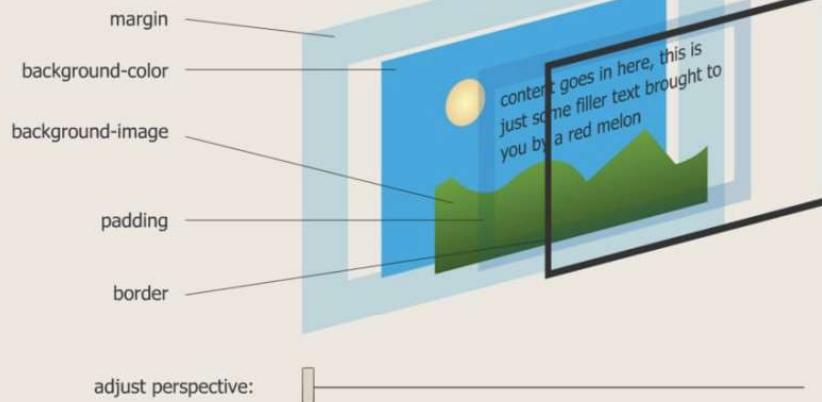
Recordemos también el modelo de caja

CONQUERBLOCKS

Introducción

Basic CSS Box Model Demo

made by redmelon.net

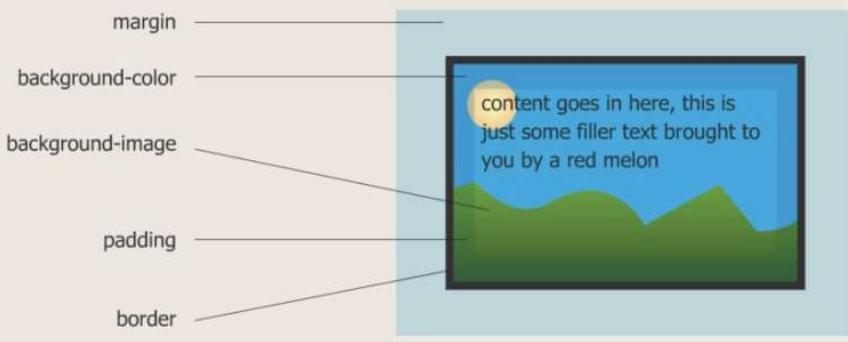


CONQUERBLOCKS

Introducción

Basic CSS Box Model Demo

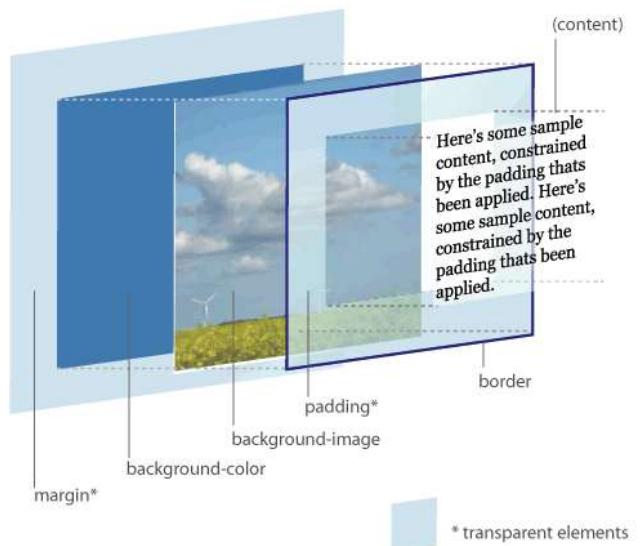
made by redmelon.net



CONQUERBLOCKS

Introducción

THE CSS BOX MODEL HIERARCHY



CONQUERBLOCKS

Background-image

CONQUERBLOCKS

Background-image

Lo usaremos para poder establecer imágenes de fondo en nuestros contenedores

CONQUERBLOCKS

Background-image

Podemos usar 1 o varias imágenes de fondo

También podremos establecer diferentes degradados como fondo

CONQUERBLOCKS

Background-image

Propiedad	Valor	Significado
background-image	none	No utiliza ninguna imagen de fondo.
background-image	<code>url("imagen.jpg")</code>	Usa la imagen indicada como fondo.
background-image	GRADIENT	Utiliza un gradiente de tipo lineal, radial o cónico.

CONQUERBLOCKS

Background-repeat

CONQUERBLOCKS

Background-repeat

CSS Syntax

```
background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit;
```

Property Values

Value	Description	Demo
repeat	The background image is repeated both vertically and horizontally. The last image will be clipped if it does not fit. This is default	Demo >
repeat-x	The background image is repeated only horizontally	Demo >
repeat-y	The background image is repeated only vertically	Demo >
no-repeat	The background-image is not repeated. The image will only be shown once	Demo >
space	The background-image is repeated as much as possible without clipping. The first and last image is pinned to either side of the element, and whitespace is distributed evenly between the images	Demo >
round	The background-image is repeated and squished or stretched to fill the space (no gaps)	Demo >
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

CONQUERBLOCKS

Background-image

W3c y demo

CONQUERBLOCKS

Background-position

CONQUERBLOCKS

Background-position

Nos permitirá posicionar una imagen dentro de un contenedor (como fondo)

CONQUERBLOCKS

Background-position

Property Values

Value	Description	Demo
left top	If you only specify one keyword, the other value will be "center"	Demo ➔
left center		
left bottom		
right top		
right center		
right bottom		
center top		
center center		
center bottom		
$x\% \ y\%$	The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%. Default value is: 0% 0%	Demo ➔
$xpos \ ypos$	The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other CSS units . If you only specify one value, the other value will be 50%. You can mix % and positions	Demo ➔
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

CONQUERBLOCKS

W3c y demo

CONQUERBLOCKS

Background-size

CONQUERBLOCKS

Background-size

Nos permite establecer el tamaño de la imagen cuando actúa como fondo

CONQUERBLOCKS

Background-size

CSS Syntax

```
background-size: auto|length|cover|contain|initial|inherit;
```

Property Values

Value	Description	Demo
auto	Default value. The background image is displayed in its original size	Demo ➔
length	Sets the width and height of the background image. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto". Read about length units	Demo ➔
percentage	Sets the width and height of the background image in percent of the parent element. The first value sets the width, the second value sets the height. If only one value is given, the second is set to "auto"	Demo ➔
cover	Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges	Demo ➔
contain	Resize the background image to make sure the image is fully visible	Demo ➔
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

CONQUERBLOCKS

Background-size

W3c y demo

CONQUERBLOCKS

Background-attachment

CONQUERBLOCKS

Background-attachment

Nor permite fijar la imagen de fondo
aun estando de fondo del contenedor

CONQUERBLOCKS

Background-attachment

CSS Syntax

```
background-attachment: scroll|fixed|local|initial|inherit;
```

Property Values

Value	Description
scroll	The background image will scroll with the page. This is default
fixed	The background image will not scroll with the page
local	The background image will scroll with the element's contents
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

CONQUERBLOCKS

Background-attachment

[W3c y demo](#)

CONQUERBLOCKS

Background-clip

CONQUERBLOCKS

Background-clip

Con esta propiedad podemos establecer qué sección de la caja ocupará el fondo: con borde, con padding o solo en contenido

CONQUERBLOCKS

Background-clip

CSS Syntax

```
background-clip: border-box|padding-box|content-box|initial|inherit;
```

Property Values

Value	Description	Demo
border-box	Default value. The background extends behind the border	Demo >
padding-box	The background extends to the inside edge of the border	Demo >
content-box	The background extends to the edge of the content box	Demo >
initial	Sets this property to its default value. Read about initial	
inherit	Inherits this property from its parent element. Read about inherit	

CONQUERBLOCKS

Background-clip

W3c y demo

CONQUERBLOCKS

Background-blend-mode

CONQUERBLOCKS

Background-blend-mode

Nos ayuda a establecer el nivel de fusión entre la imagen de fondo con el color de fondo de detrás. Es una utilidad muy usada en los programas de edición gráfica.

CONQUERBLOCKS

Background-blend-mode

CSS Syntax

```
background-blend-mode: normal|multiply|screen|overlay|darken|lighten|color-dodge|saturation|color|luminosity;
```

Property Values

Value	Description	Demo
normal	This is default. Sets the blending mode to normal	Demo
multiply	Sets the blending mode to multiply	Demo
screen	Sets the blending mode to screen	Demo
overlay	Sets the blending mode to overlay	Demo
darken	Sets the blending mode to darken	Demo
lighten	Sets the blending mode to lighten	Demo
color-dodge	Sets the blending mode to color-dodge	Demo
saturation	Sets the blending mode to saturation	Demo
color	Sets the blending mode to color	Demo
luminosity	Sets the blending mode to luminosity	Demo

CONQUERBLOCKS

Background-blend-mode

W3c y demo

CONQUERBLOCKS

Background

CONQUERBLOCKS

Background

Con esta propiedad shorthand estableceremos todas las propiedades de una vez.

CONQUERBLOCKS

Background

CSS Syntax

```
background: bg-color bg-image position/bg-size bg-repeat bg-origin bg-clip bg-attachment initial|inherit;
```

Property Values

Value	Description	CSS	Demo
<i>background-color</i>	Specifies the background color to be used	1	Demo >
<i>background-image</i>	Specifies ONE or MORE background images to be used	1	Demo >
<i>background-position</i>	Specifies the position of the background images	1	Demo >
<i>background-size</i>	Specifies the size of the background images	3	Demo >
<i>background-repeat</i>	Specifies how to repeat the background images	1	Demo >
<i>background-origin</i>	Specifies the positioning area of the background images	3	Demo >
<i>background-clip</i>	Specifies the painting area of the background images	3	Demo >
<i>background-attachment</i>	Specifies whether the background images are fixed or scrolls with the rest of the page	1	
initial	Sets this property to its default value. Read about initial	3	
inherit	Inherits this property from its parent element. Read about inherit	2	

CONQUERBLOCKS

Background

W3c y demo

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 16



CONQUERBLOCKS

<Índice>

Clase 16

Clase 16: Manejo de imágenes en CSS II

Background-image

Background-position, clip y size

Background

Object-fit

Aspect-ratio

Gradient

Manejo de img

CONQUERBLOCKS

Object-fit

CONQUERBLOCKS

Object-fit

Es probable que nos encontremos con un escenario en el que desee conservar la relación de aspecto original cuando trabaje con imágenes.

CONQUERBLOCKS

Object-fit

Preservar la relación de aspecto evitará que las imágenes aparezcan distorsionadas al estirarse o aplastarse.

CONQUERBLOCKS

Object-fit

La propiedad **object-fit** nos va a permitir cambiar el modo en el que se rellena una imagen en su contenedor padre.

CONQUERBLOCKS

Object-fit

Valor	Descripción
fill	Rellena la imagen en el espacio del contenedor padre. Habitualmente, estirando la imagen.
contain	Mantiene el aspecto natural conteniendo el máximo posible de la imagen.
cover	Mantiene el aspecto natural cubriendo el contenedor padre, de modo que no hayan huecos sin cubrir.
none	Mantiene el aspecto natural.

CONQUERBLOCKS

Object-fit



CONQUERBLOCKS

Object-fit



CONQUERBLOCKS

Object-fit

DEMO

CONQUERBLOCKS

Object-position

CONQUERBLOCKS

Object-position

Además, tenemos la propiedad object-position que nos sirve para utilizar junto a la propiedad object-fit y cambiar la posición donde aparece la imagen, especialmente cuando está recortada y sólo aparece un fragmento o parte de la imagen

CONQUERBLOCKS

Object-position

La propiedad funciona de forma muy parecida a como lo hace la propiedad background-position

CONQUERBLOCKS

Object-position

Valor	Descripción
50% 50%	Por defecto, la imagen está centrada tanto en X como en Y.
PERCENT PERCENT	Se puede indicar un porcentaje para colocarlo en el eje correspondiente.

CONQUERBLOCKS

Object-position

DEMO

CONQUERBLOCKS

Aspect-ratio

CONQUERBLOCKS

Aspect-ratio

La propiedad aspect-ratio permite cambiar la proporción de aspecto entre los valores width y height de una imagen u objeto desde CSS.

CONQUERBLOCKS

Aspect-ratio

De esta forma, podemos asegurarnos de que las imágenes no se deformarán o tendrán una relación de aspecto no adecuada en nuestras páginas, sin necesidad de darle un tamaño concreto de ancho y alto al elemento.

CONQUERBLOCKS

Aspect-ratio

Valor	Descripción
<code>auto</code>	El navegador calcula automáticamente la proporción adecuada.
<code>NUMBER / NUMBER</code>	Se utilizará la proporción <code>width / height</code> indicada.
<code>auto NUMBER / NUMBER</code>	Utiliza la proporción indicada, salvo que tenga un tamaño definido.
<code>NUMBER</code>	El valor de <code>aspect-ratio</code> también puede darse como un número decimal en lugar de <code>a / b</code> .

CONQUERBLOCKS

Aspect-ratio

DEMO

CONQUERBLOCKS

Gradient

CONQUERBLOCKS

Gradient

Además de poder establecer colores e imágenes de fondo, en CSS cabe la posibilidad de establecer degradados de fondo.



Gradient Backgrounds

CONQUERBLOCKS

Gradient

El primer tipo de gradiente es el **gradiente lineal**, que permite crear fondos degradados que van en una dirección y cambian de un color a otro, dependiendo de la lista de colores indicada por parámetro.

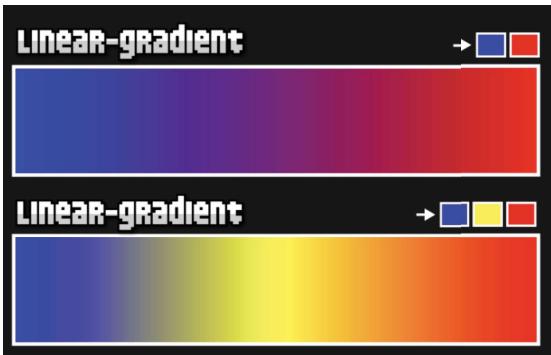
CONQUERBLOCKS

Gradient

Función y modalidad	Significado
<code>linear-gradient(COLOR , COLOR , ...)</code>	Gradiente de colores (hacia abajo).
<code>linear-gradient(DIRECTION , COLOR , COLOR , ...)</code>	Gradiente con dirección específica.
<code>linear-gradient(DIRECTION , COLOR SIZE , COLOR SIZE , ...)</code>	Gradiente indicando donde comienza a cambiar el color.
<code>linear-gradient(DIRECTION , COLOR SIZE SIZE , COLOR SIZE SIZE , ...)</code>	Gradiente indicando inicio y fin de cada color.

CONQUERBLOCKS

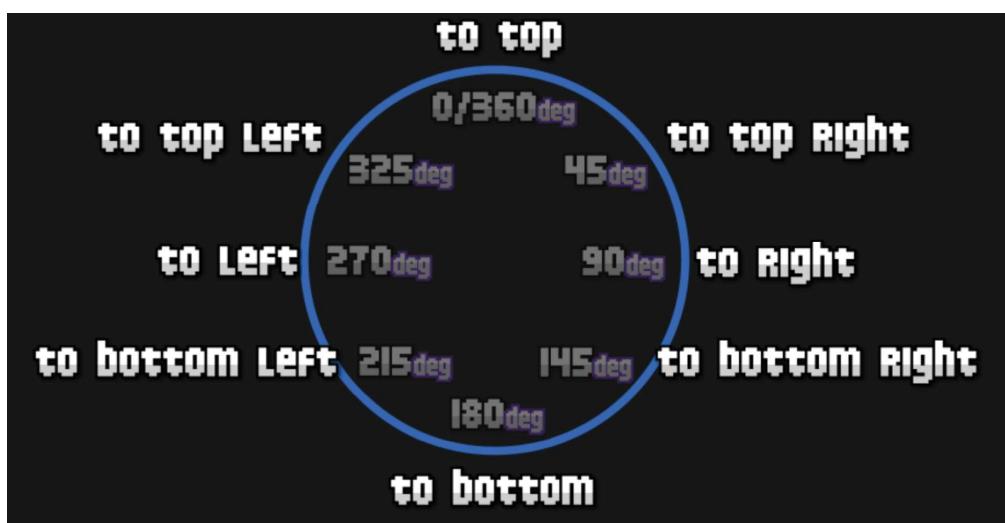
Gradient



```
.element {  
  background-image: linear-gradient(blue, red);  
}
```

CONQUERBLOCKS

Gradient



CONQUERBLOCKS

Gradient

Por defecto, las distancias entre colores se ajustan automáticamente con proporciones equilibradas. Sin embargo, si especificamos un tamaño (porcentaje o píxeles, por ejemplo) justo después del color en cuestión, podremos variar la posición en la que aparece ese color.

CONQUERBLOCKS

Gradient

```
.element {  
    /* Básico, con colores con distancias equilibradas */  
    background: linear-gradient(to right, black, blue, gold, green, red);  
  
    /* Equivalente al anterior */  
    background:  
        linear-gradient(  
            to right,  
            black 0%,      /* Color 100% negro en el punto 0% y comienza a cambiar */  
            blue 25%,     /* Color 100% azul en el punto 25% y comienza a cambiar */  
            gold 50%,     /* Color 100% dorado en el punto 50% y comienza a cambiar */  
            green 75%,    /* Color 100% verde en el punto 75% y comienza a cambiar */  
            red 100%      /* Color 100% rojo en el punto 100% */  
        );  
}
```

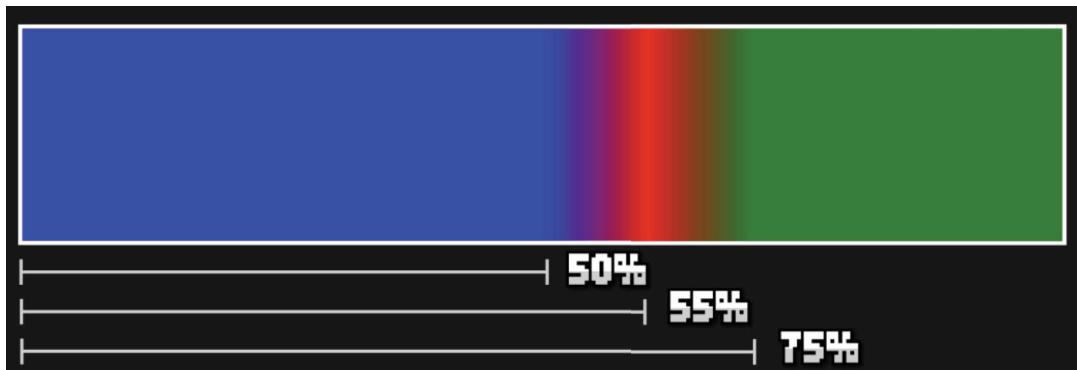
CONQUERBLOCKS

Gradient

Este porcentaje o representa el punto central del color que lo precede, respecto al tamaño del gradiente completo. Se muestran tres colores: el azul comenzando a cambiar en la posición 50% del elemento, el rojo comenzando a cambiar en la posición 55%, y el verde comenzando a cambiar en la posición 75%. Se vería algo así:

CONQUERBLOCKS

Gradient



CONQUERBLOCKS

Gradient

DEMO

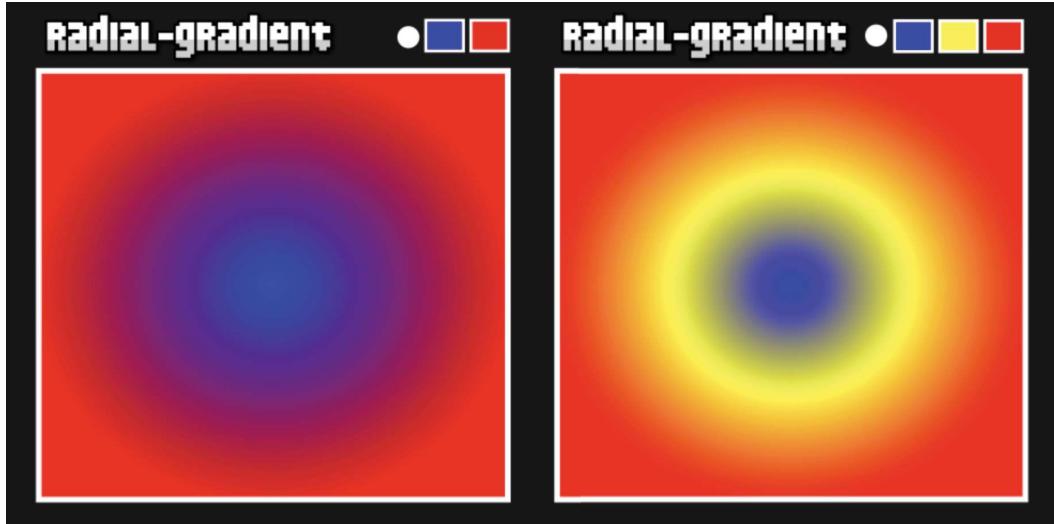
CONQUERBLOCKS

Gradient

Dentro de la categoría de gradientes tenemos una función que permite crear los llamados gradientes radiales. Estos son un tipo de degradado con formas circulares o elípticas. Para crearlos, sólo tenemos que utilizar la función radial-gradient() en lugar de la más conocida y frecuente función linear-gradient().

CONQUERBLOCKS

Gradient



CONQUERBLOCKS

Gradient

Función y modalidad	Significado
radial-gradient(COLOR , COLOR , ...)	Gradiente básico entre colores.
radial-gradient(<u>forma</u> , COLOR , COLOR , ...)	Gradiente con forma circular o elíptica.
radial-gradient(<u>forma</u> SIZE , COLOR , COLOR , ...)	Gradiente con tamaño o dimensión.
radial-gradient(<u>forma</u> SIZE at <u>ubicación</u> , COLOR , COLOR , ...)	Gradiente con colocación inicial.
radial-gradient(<u>forma</u> SIZE at <u>ubicación</u> , COLOR SIZE , COLOR SIZE , ...)	Gradiente con tamaños entre colores.
radial-gradient(<u>forma</u> SIZE at <u>ubicación</u> , COLOR SIZE SIZE , COLOR SIZE , ...)	Gradiente con tamaños de inicio y fin.

CONQUERBLOCKS

Gradient

Demo

CONQUERBLOCKS

Gradient

Generador online

CONQUERBLOCKS

Gradient

Existe un tercer tipo de gradiente en CSS, visualmente muy similar al gradiente radial, pero haciendo referencia a una figura cónica vista desde un plano superior, incluyendo sus sombras y brillos.

CONQUERBLOCKS

Gradient

Demo

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 17



CONQUERBLOCKS

<Índice>

Clase 17

Clase 17: Posicionamiento I

Introducción

Propiedad Display

Propiedad Position

Posicionamiento estático o normal

Propiedad Visibility

Display vs Opacity vs Visibility

CONQUERBLOCKS

Introducción

CONQUERBLOCKS

Introducción

Una de las partes más complejas de CSS sea la creación de layouts, colocación y distribución de los elementos de una página. Todo ha evolucionado en muy poco tiempo

CONQUERBLOCKS

Introducción

Sin embargo, es una parte fundamental dentro de CSS, y es necesario entenderla correctamente para que resulte más fácil de trabajar y crear nuestros diseños.

CONQUERBLOCKS

Introducción

Antes de comenzar, debemos saber que cada etiqueta HTML tiene un «tipo de representación visual» en el navegador, lo que habitualmente se suele denominar el tipo de caja.

CONQUERBLOCKS

Introducción

todos los elementos de una página web deben tener un tipo de representación, que es la forma en la que se va a comportar. En principio, partimos de dos tipos básicos y fundamentales llamados inline y block.

CONQUERBLOCKS

Introducción

¿Qué es inline y block?

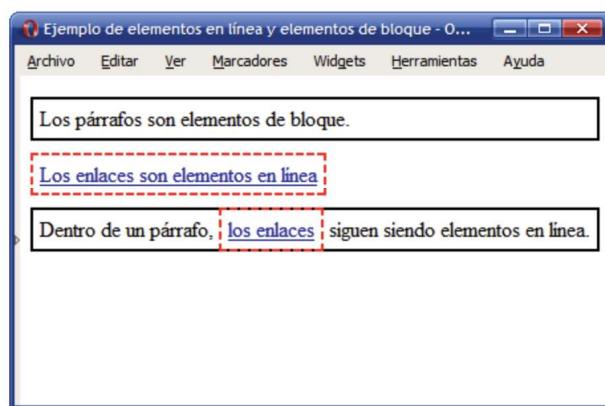


Figura 5.1 Cajas creadas por los elementos de línea y los elementos de bloque

CONQUERBLOCKS

Introducción

Cuando los navegadores descargan el contenido HTML y CSS de las páginas web, aplican un procesamiento muy complejo antes de mostrar las páginas en la pantalla del usuario.

CONQUERBLOCKS

Introducción

Para cumplir con el modelo de cajas presentado en el capítulo anterior, los navegadores crean una caja para representar a cada elemento de la página HTML

CONQUERBLOCKS

Introducción

Los factores que se tienen en cuenta para generar cada caja son:

- Las propiedades width y height de la caja
- El tipo de cada elemento HTML
- Posicionamiento de la caja (normal, relativo, absoluto, fijo o flotante)
- Las relaciones entre elementos (dónde se encuentra cada elemento, elementos descendientes, etc.)
- Otro tipo de información, como por ejemplo el tamaño de las imágenes y el tamaño de la ventana del navegador.

CONQUERBLOCKS

display

CONQUERBLOCKS

display

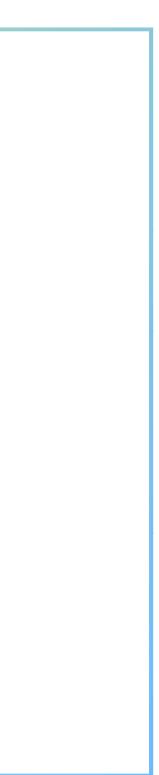
La propiedad display de CSS permite modificar el comportamiento de un elemento HTML, cambiándolo al que le indiquemos, como por ejemplo inline o block (u otro de los que veremos más adelante)

CONQUERBLOCKS

display

Tipo de caja	Características
block	Se apila en vertical. Ocupa todo el ancho disponible de su etiqueta contenedora.
Versión en línea	
inline	Se coloca en horizontal. Se adapta al ancho de su contenido. Ignora width o height.
inline-block	Combinación de los dos anteriores. Se comporta como inline pero no ignora width o height.

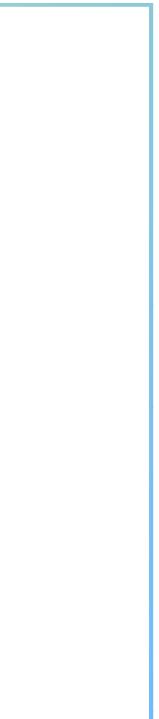
CONQUERBLOCKS



display

block

CONQUERBLOCKS



display

inline

CONQUERBLOCKS



display

inline-block

Si utilizamos la propiedad display con el valor inline-block, conseguiremos un elemento que funcionará como si fuera un elemento inline, pero haciendo caso a las propiedades width y height, las cuales se ignoran si estamos utilizando un display: inline.

CONQUERBLOCKS

display

Demo

CONQUERBLOCKS

display

vertical-align

nos ayudará a establecer el alineamiento vertical de los elementos de línea

Demo

CONQUERBLOCKS

display

Mediante la propiedad display también podemos utilizar flex, grid, list-item o table.

Los veremos por separado.

CONQUERBLOCKS



display

none

CONQUERBLOCKS



position

CONQUERBLOCKS

position

Los navegadores crean y posicionan de forma automática todas las cajas que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestra cada caja.

CONQUERBLOCKS

position

Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y diseñar estructuras de páginas que de otra forma no serían posibles.

CONQUERBLOCKS

position

El estándar de CSS define cinco modelos diferentes para posicionar una caja

Valor	Significado
static	Posicionamiento estático. Utiliza el orden natural de los elementos HTML.
relative	Posicionamiento relativo. Los elementos se mueven ligeramente en base a su posición estática.
absolute	Posicionamiento absoluto. Los elementos se colocan en base al contenedor padre.
fixed	Posicionamiento fijo. Idem al absoluto, pero aunque hagamos scroll no se mueve.
sticky	Posicionamiento «pegado». Similar al relativo, usado para pegar menús a la parte superior.

CONQUERBLOCKS

position

Posicionamiento normal o estático

Se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario.

CONQUERBLOCKS

position

Posicionamiento relativo

variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.

CONQUERBLOCKS

position

Posicionamiento absoluto

la posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.

CONQUERBLOCKS

position

Posicionamiento fijo

variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.

CONQUERBLOCKS

position

Posicionamiento flotante

se trata del modelo más especial de posicionamiento, ya que desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran

CONQUERBLOCKS

position

Posicionamiento sticky

Hará que en un contenedor con scroll el contenido se quede pegado en la parte de arriba

CONQUERBLOCKS

position

la propiedad position sólo indica cómo se posiciona una caja, pero no la desplaza.

CONQUERBLOCKS

position

Normalmente, cuando se posiciona una caja también es necesario desplazarla respecto de su posición original o respecto de otro origen de coordenadas. CSS define cuatro propiedades llamadas top, right, bottom y left para controlar el desplazamiento de las cajas posicionadas

CONQUERBLOCKS

position

Propiedad	Valor	Significado
top:	auto SIZE	Empuja el elemento una distancia desde la parte superior hacia el inferior.
bottom:	auto SIZE	Empuja el elemento una distancia desde la parte inferior hacia la superior.
left:	auto SIZE	Empuja el elemento una distancia desde la parte izquierda hacia la derecha.
right:	auto SIZE	Empuja el elemento una distancia desde la parte derecha hacia la izquierda.

CONQUERBLOCKS

Posicionamiento normal o estático

CONQUERBLOCKS

Posicionamiento normal

El posicionamiento normal o estático es el modelo que utilizan por defecto los navegadores para mostrar los elementos de las páginas.

CONQUERBLOCKS

Posicionamiento normal

En este modelo, sólo se tiene en cuenta si el elemento es de bloque o en línea, sus propiedades width y height y su contenido.

CONQUERBLOCKS

Posicionamiento normal

Los elementos de bloque forman lo que CSS denomina "contextos de formato de bloque". En este tipo de contextos, las cajas se muestran una debajo de otra comenzando desde el principio del elemento contenedor. La distancia entre las cajas se controla mediante los márgenes verticales.

CONQUERBLOCKS

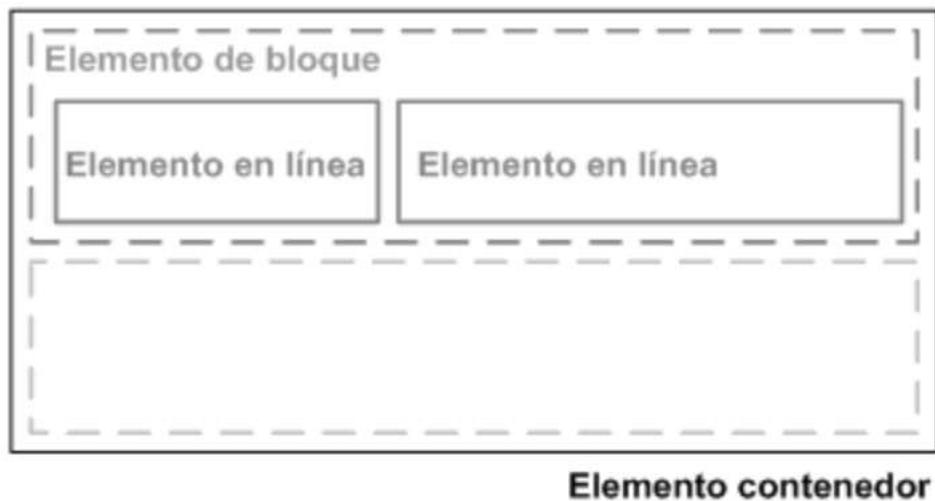
Posicionamiento normal



Figura 5.2 Posicionamiento normal de los elementos de bloque

CONQUERBLOCKS

Posicionamiento normal



CONQUERBLOCKS

Posicionamiento normal

Si un elemento se encuentra dentro de otro, el elemento padre se llama "elemento contenedor" y determina tanto la posición como el tamaño de todas sus cajas interiores.

CONQUERBLOCKS

Posicionamiento normal

Si un elemento no se encuentra dentro de un elemento contenedor, entonces su elemento contenedor es el elemento `<body>` de la página.

Normalmente, la anchura de los elementos de bloque está limitada a la anchura de su elemento contenedor, aunque en algunos casos sus contenidos pueden desbordar el espacio disponible.

CONQUERBLOCKS

Posicionamiento normal

¿Os suena verdad? Es el comportamiento que llevamos viendo desde el principio, pero ahora sabéis exactamente por qué se posicionaban los elementos así

CONQUERBLOCKS

visibility

CONQUERBLOCKS

visibility

Las propiedades display y visibility controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página

CONQUERBLOCKS

visibility

Habitualmente se utilizan junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes cuando el usuario pincha sobre ellos.

CONQUERBLOCKS

visibility

La propiedad `visibility` permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento pero no la muestra.

CONQUERBLOCKS

visibility

La propiedad `visibility` permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento pero no la muestra.

CONQUERBLOCKS

visibility

La propiedad `visibility` permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento pero no la muestra.

CONQUERBLOCKS

visibility

Value	Description
<code>visible</code>	Default value. The element is visible
<code>hidden</code>	The element is hidden (but still takes up space)

CONQUERBLOCKS

visibility

Inicialmente todas las cajas que componen la página son visibles. Empleando el valor hidden es posible convertir una caja en invisible para que no muestre sus contenidos.

CONQUERBLOCKS

visibility

El resto de elementos de la página se muestran **como si la caja todavía fuera visible**, por lo que en el lugar donde originalmente se mostraba la caja invisible, ahora se muestra un hueco vacío.

CONQUERBLOCKS

Display vs Opacity vs Visibility

CONQUERBLOCKS

Display vs Opacity vs Visibility

Siempre existen diferentes formas de
hacer la misma cosa

CONQUERBLOCKS

Display vs Opacity vs Visibility

No siempre tienen el mismo efecto

CONQUERBLOCKS

Display vs Opacity vs Visibility

```
visibility: hidden;  
opacity: 0;  
display: none;
```

CONQUERBLOCKS

Ejemplos

CONQUERBLOCKS

Ejemplos

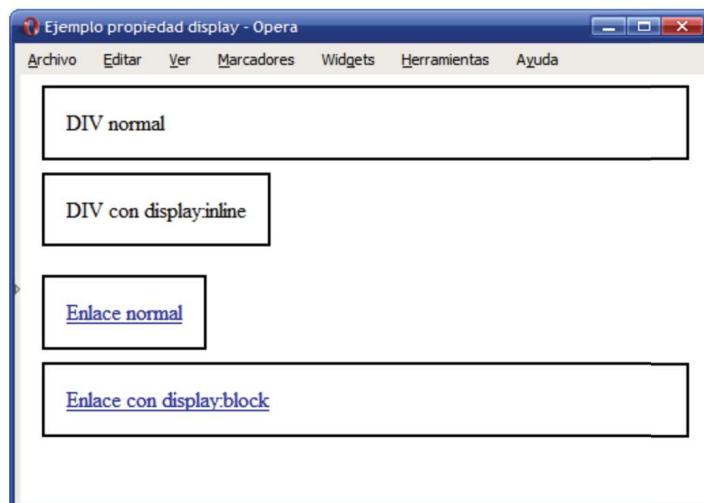


Figura 5.22 Ejemplo de propiedad display

CONQUERBLOCKS

Ejemplos

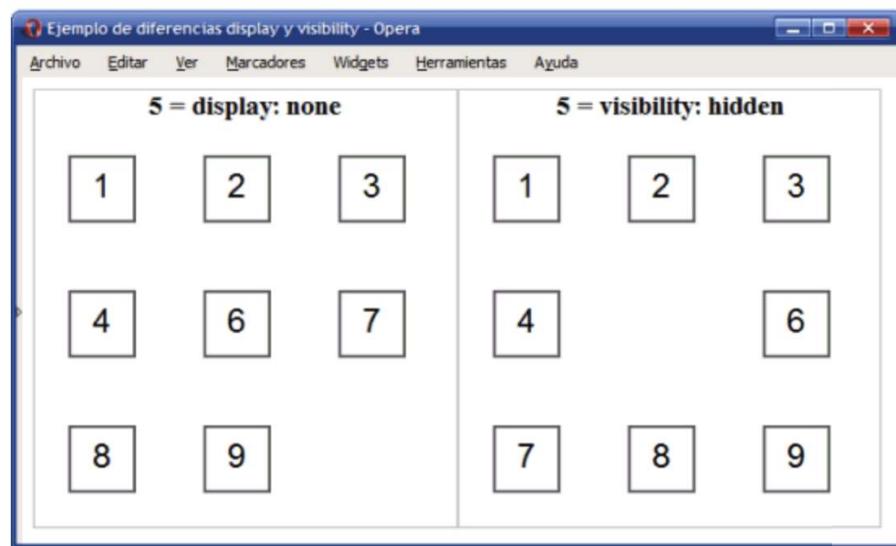


Figura 5.21 Diferencias visuales entre las propiedades display y visibility

CONQUERBLOCKS

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

[@bienvenidosaez](#)

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 18



CONQUERBLOCKS

<Índice>

Clase 18

Clase 18: Posicionamiento II

Posicionamiento relativo

Posicionamiento absoluto

Posicionamiento fijo

Float

Sticky

CONQUERBLOCKS

Posicionamiento relativo

CONQUERBLOCKS

Posicionamiento relativo

Si utilizamos la palabra clave relative activaremos el modo de posicionamiento relativo, que es el más sencillo de todos, quitando el por defecto

CONQUERBLOCKS

Posicionamiento relativo

En este modo, los elementos se colocan exactamente igual que en el posicionamiento estático (permanecen en la misma posición), pero dependiendo del valor de las propiedades top, bottom, left o right variaremos ligeramente la posición del elemento

CONQUERBLOCKS

Posicionamiento relativo

El valor de la propiedad top se interpreta como el desplazamiento entre el borde superior de la caja en su posición final y el borde superior de la misma caja en su posición original

CONQUERBLOCKS

Posicionamiento relativo

De la misma forma, el valor de las propiedades left, right y bottom indica respectivamente el desplazamiento entre el borde izquierdo/derecho/inferior de la caja en su posición final y el borde izquierdo/derecho/inferior de la caja original.

CONQUERBLOCKS

Posicionamiento relativo

En el posicionamiento relativo se puede decir que “empujamos” las cajas

CONQUERBLOCKS

Posicionamiento relativo

- La propiedad top se emplea para mover las cajas de forma descendente
- La propiedad bottom mueve las cajas de forma ascendente
- La propiedad left se utiliza para desplazar las cajas hacia la derecha
- La propiedad right mueve las cajas hacia la izquierda.

CONQUERBLOCKS

Posicionamiento relativo

Este comportamiento parece poco intuitivo y es causa de errores cuando se empiezan a diseñar páginas con CSS

CONQUERBLOCKS

Posicionamiento relativo

Si se utilizan valores negativos en las propiedades top, right, bottom y left, su efecto es justamente el inverso.

`left: 100px = right: -100px`

CONQUERBLOCKS

Posicionamiento relativo

El desplazamiento relativo de una caja **no afecta al resto de cajas adyacentes**, que se muestran en la misma posición que si la caja desplazada no se hubiera movido de su posición original

CONQUERBLOCKS

Posicionamiento relativo



Figura 5.5 Elementos posicionados de forma normal

CONQUERBLOCKS

Posicionamiento relativo

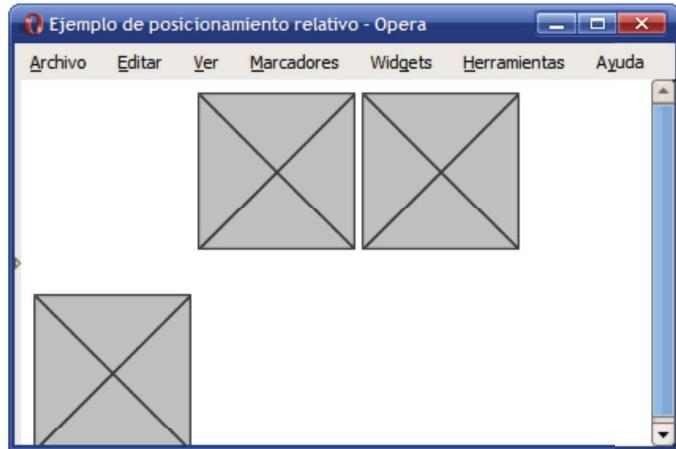
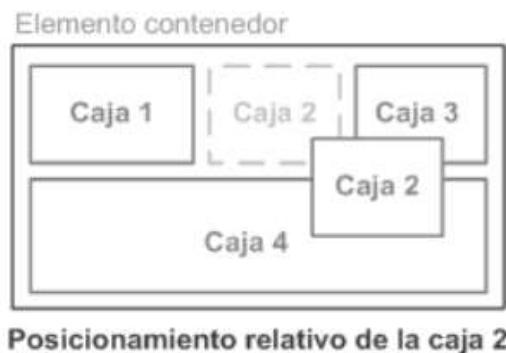
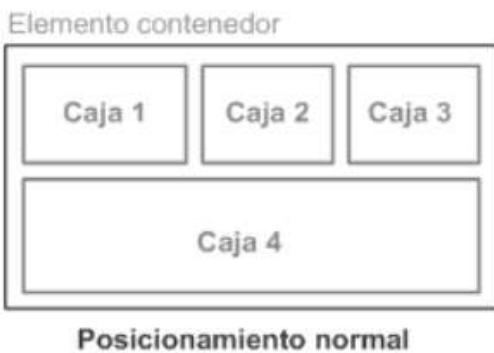


Figura 5.6 Elemento posicionado de forma relativa

CONQUERBLOCKS

Posicionamiento relativo



CONQUERBLOCKS

Posicionamiento relativo

Demo

CONQUERBLOCKS

Posicionamiento relativo

Position static con margin VS Relative

CONQUERBLOCKS

Posicionamiento absoluto

CONQUERBLOCKS

Posicionamiento absoluto

El posicionamiento absoluto se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento

Posicionamiento potentísimo

CONQUERBLOCKS

Posicionamiento absoluto

Este tipo de posicionamiento cambia totalmente el posicionamiento normal

CONQUERBLOCKS

Posicionamiento absoluto

Atención a la frase

CONQUERBLOCKS

Posicionamiento absoluto

Cuando una caja está posicionada de forma absoluta, su origen de coordenadas, el primer ancestro que esté posicionado de forma diferente al normal o estático

CONQUERBLOCKS

Posicionamiento absoluto

Cuando una caja está posicionada de forma absoluta sale del flujo normal del posicionamiento, por lo que las demás cajas se comportan como si no existiera

CONQUERBLOCKS

Posicionamiento absoluto

Usaremos también las propiedades:
top, bottom, left, right
pero funcionarán totalmente diferente al
posicionamiento relativo.

Marcarán la posición donde se empezará a pintar
la caja que está posicionada de forma absoluta.

CONQUERBLOCKS

Posicionamiento absoluto

Determinar la referencia utilizada para interpretar los valores de top, right, bottom y left de una caja posicionada de forma absoluta es un proceso complejo que se compone de los siguientes pasos:

- Se buscan todos los elementos contenedores de la caja hasta llegar al elemento <body> de la página.
- Se recorren todos los elementos contenedores empezando por el más cercano a la caja y llegando hasta el <body>

CONQUERBLOCKS

Posicionamiento absoluto

- El primer elemento contenedor que esté posicionado de cualquier forma diferente a position: static se convierte en la referencia que determina la posición de la caja posicionada de forma absoluta.
- Si ningún elemento contenedor está posicionado, la referencia es la ventana del navegador, que no debe confundirse con el elemento <body> de la página.

CONQUERBLOCKS

Posicionamiento absoluto

Una vez determinada la referencia del posicionamiento absoluto, la interpretación de los valores de las propiedades top, right, bottom y left se realiza como sigue:

- El valor de la propiedad top indica el desplazamiento desde el borde superior de la caja hasta el borde superior del elemento contenedor que se utiliza como referencia.
- El valor de la propiedad right indica el desplazamiento desde el borde derecho de la caja hasta el borde derecho del elemento contenedor que se utiliza como referencia.

CONQUERBLOCKS

Posicionamiento absoluto

- El valor de la propiedad bottom indica el desplazamiento desde el borde inferior de la caja hasta el borde inferior del elemento contenedor que se utiliza como referencia.
- El valor de la propiedad left indica el desplazamiento desde el borde izquierdo de la caja hasta el borde izquierdo del elemento contenedor que se utiliza como referencia.

CONQUERBLOCKS

Posicionamiento absoluto

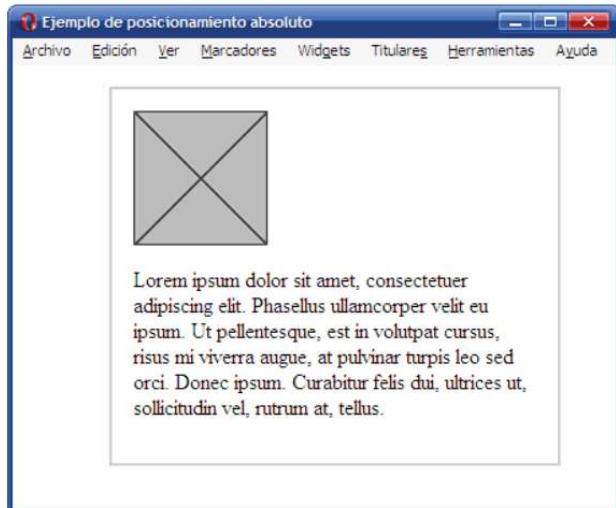


Figura 5.8 Situación original antes de modificar el posicionamiento

CONQUERBLOCKS

Posicionamiento absoluto

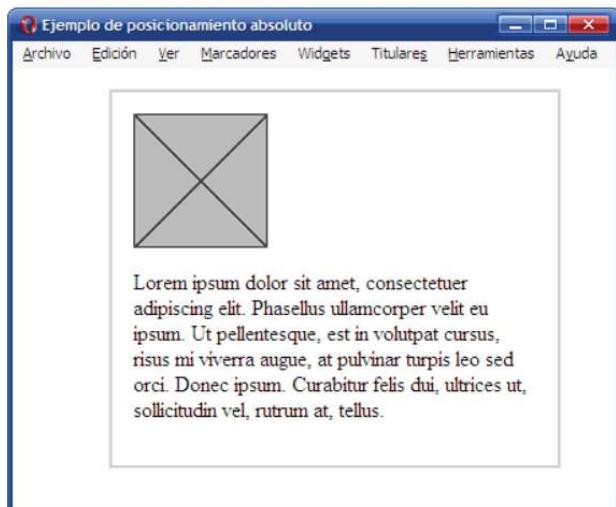


Figura 5.8 Situación original antes de modificar el posicionamiento

CONQUERBLOCKS

Posicionamiento absoluto

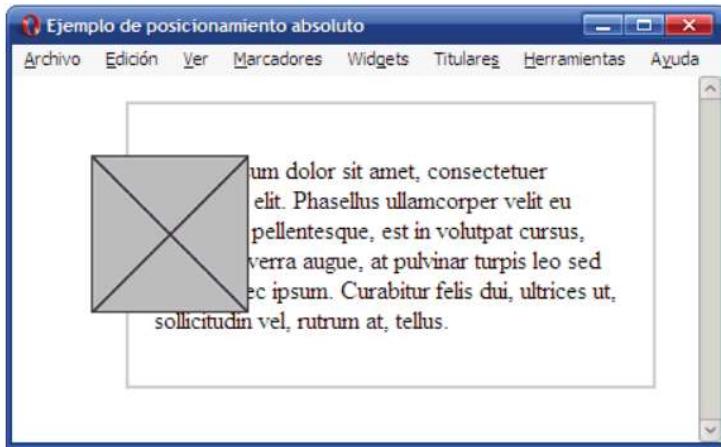


Figura 5.9 Imagen posicionada de forma absoluta

CONQUERBLOCKS

Posicionamiento absoluto

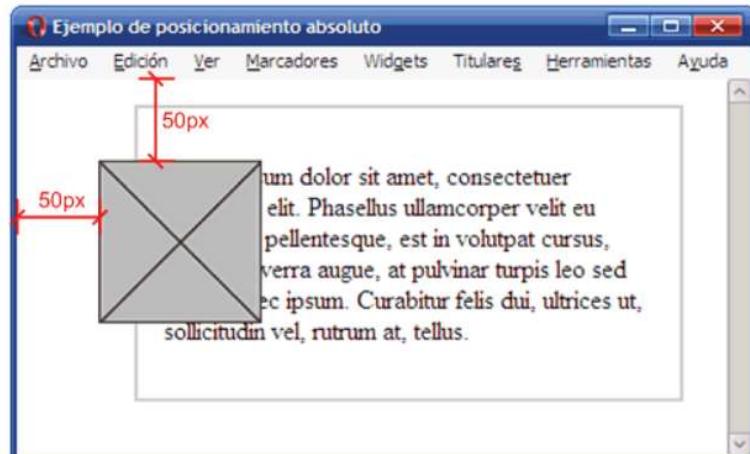


Figura 5.10 La referencia del posicionamiento absoluto es la ventana del navegador

CONQUERBLOCKS

Posicionamiento absoluto

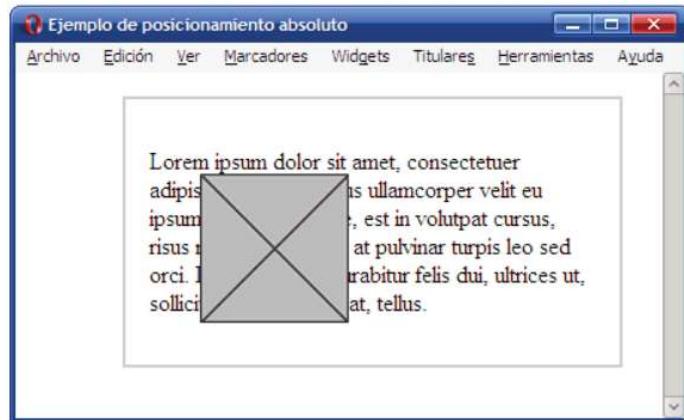


Figura 5.11 Imagen posicionada de forma absoluta

CONQUERBLOCKS

Posicionamiento absoluto

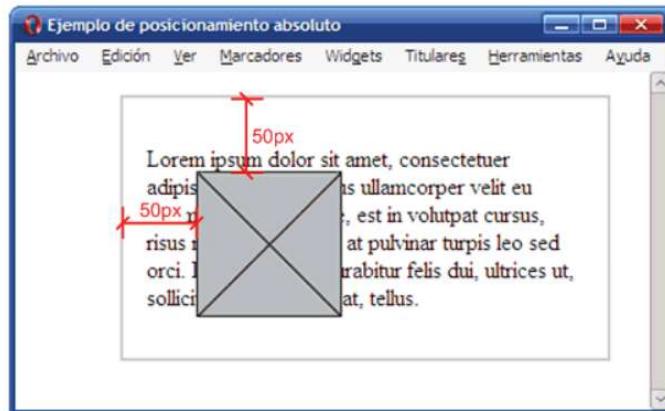
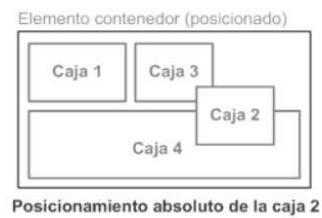
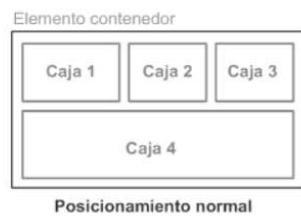
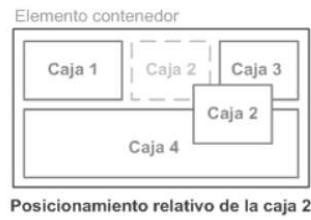
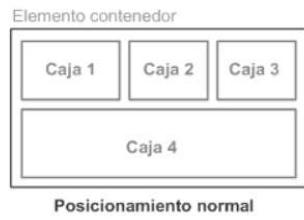


Figura 5.12 La referencia del posicionamiento absoluto es el elemento contenedor de la imagen

CONQUERBLOCKS

Posicionamiento absoluto



CONQUERBLOCKS

Posicionamiento absoluto

Demo

CONQUERBLOCKS

Posicionamiento fijo

CONQUERBLOCKS

Posicionamiento fijo

El estándar CSS considera que el posicionamiento fijo es un caso particular del posicionamiento absoluto, ya que sólo se diferencian en el comportamiento de las cajas posicionadas.

CONQUERBLOCKS

Posicionamiento fijo

Cuando una caja se posiciona de forma fija, la forma de obtener el origen de coordenadas para interpretar su desplazamiento es idéntica al posicionamiento absoluto. De hecho, si el usuario no mueve la página HTML en la ventana del navegador, no existe ninguna diferencia entre estos dos modelos de posicionamiento.

CONQUERBLOCKS

Posicionamiento fijo

La principal característica de una caja posicionada de forma fija es que su posición es inamovible dentro de la ventana del navegador. El posicionamiento fijo hace que las cajas no modifiquen su posición ni aunque el usuario suba o baje la página en la ventana de su navegador.

CONQUERBLOCKS

Posicionamiento fijo

Si la página se visualiza en un medio paginado (por ejemplo en una impresora) las cajas posicionadas de forma fija se repiten en todas las páginas. Esta característica puede ser útil para crear encabezados o pies de página en páginas HTML preparadas para imprimir.

CONQUERBLOCKS

Posicionamiento fijo

Demo

CONQUERBLOCKS

Float

CONQUERBLOCKS

Float

Existe una propiedad denominada float que tiene un funcionamiento peculiar con el que cambiamos el flujo de ordenación de elementos.

CONQUERBLOCKS

Float

La propiedad float es una propiedad que podría ser interesante en determinadas condiciones, sin embargo, el código resultante suele ser más sucio y complejo de lo que sería mediante otros métodos actuales, por lo que se recomienda sólo utilizarlo por razones de retrocompatibilidad y darle preferencia a métodos como Flexbox o Grid CSS.

CONQUERBLOCKS

Float

Con la propiedad float puedes conseguir que los elementos que quieras, alteren su comportamiento y floten a la izquierda (left) o a la derecha (right). Con el valor none (valor por defecto) eliminas esta característica de desplazamiento.

CONQUERBLOCKS

Float

Demo

CONQUERBLOCKS

Sticky

CONQUERBLOCKS

Sticky

El posicionamiento sticky se suele utilizar cuando queremos que un elemento se posicione en un lugar específico de forma fija («sticky», pegajoso), como por ejemplo, cuando al hacer scroll llegamos a un elemento y queremos que ese elemento se quede fijo en la parte superior mientras continuamos haciendo scroll.

CONQUERBLOCKS

Sticky

Este comportamiento es muy habitual con los menús superiores de las páginas o las cabeceras de secciones.

CONQUERBLOCKS

Sticky

Es muy importante, para que funcione position: sticky; que el elemento contenedor del “elemento pegajoso” tenga scroll, es decir, que sea más alto de lo que se ve en pantalla. De lo contrario no funcionará.

CONQUERBLOCKS

Sticky

Demo

CONQUERBLOCKS

Apilamiento con z-index

CONQUERBLOCKS

z-index

Cuando modificamos la posición de las cajas, puede ocurrir apilamiento entre los diferentes elementos

CONQUERBLOCKS

z-index

z-index

establece el nivel de profundidad en el que está un elemento sobre los demás. De esta forma, podemos hacer que un elemento se coloque encima o debajo de otro.

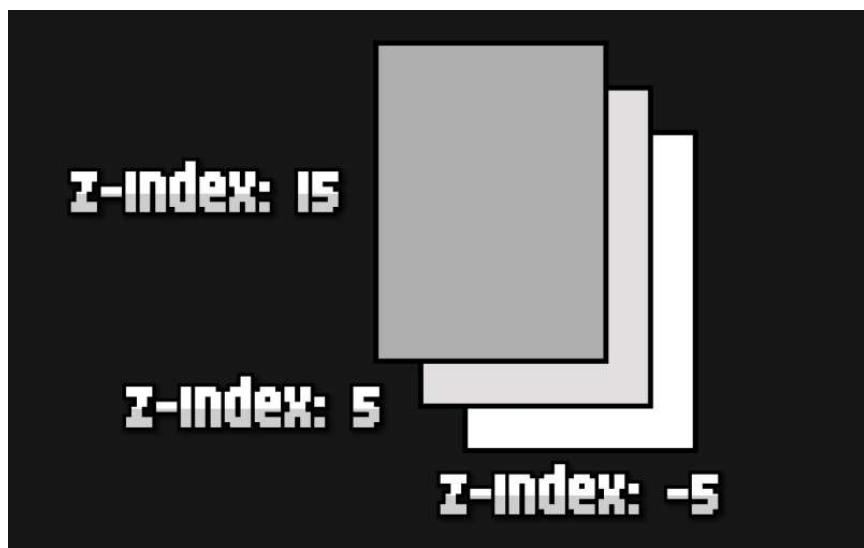
CONQUERBLOCKS

z-index

Su funcionamiento es muy sencillo, sólo hay que indicar un número que representará el nivel de profundidad del elemento. Los elementos un número más alto estarán por encima de otros con un número más bajo, que permanecerán ocultos detrás de los primeros.

CONQUERBLOCKS

z-index



CONQUERBLOCKS

z-index

Nota: Los niveles z-index, así como las propiedades top, left, bottom y right no funcionan con elementos que estén utilizando posicionamiento estático. **Deben tener un tipo de posicionamiento diferente a estático.**

CONQUERBLOCKS

Ejemplos

CONQUERBLOCKS

Ejemplos

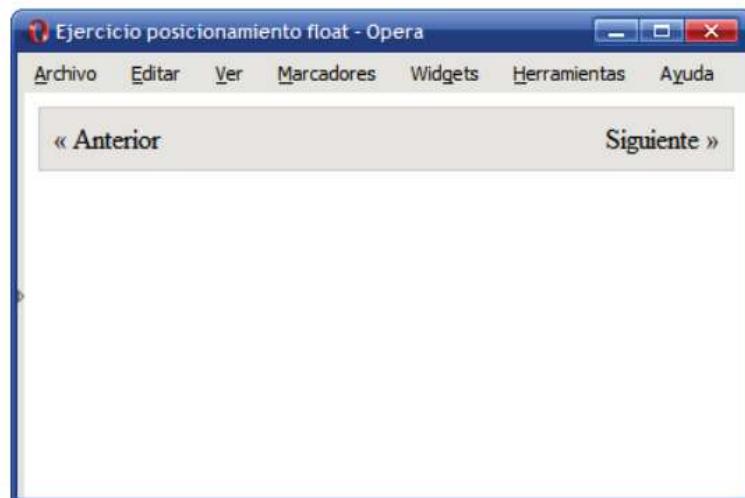
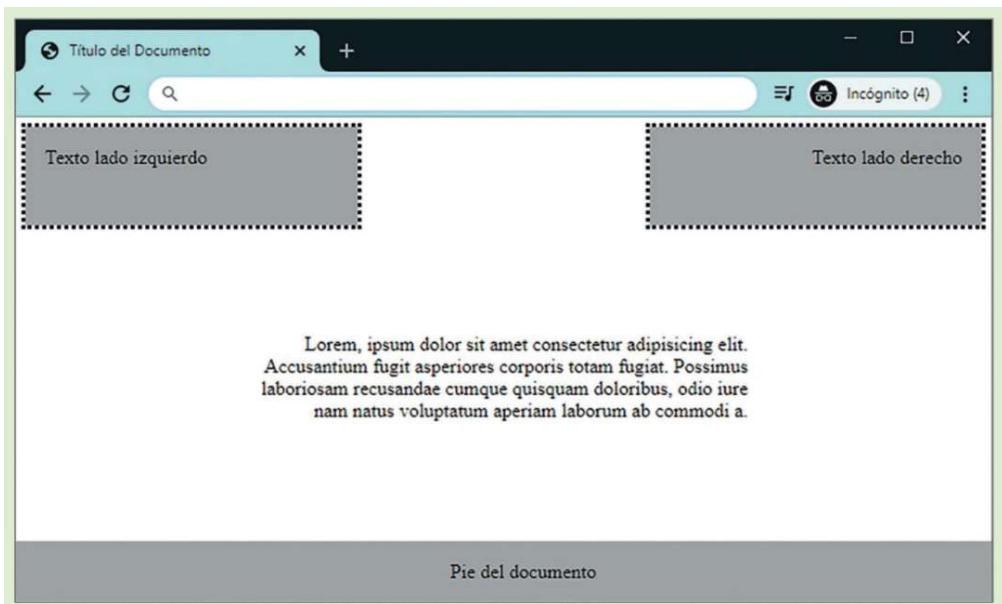


Figura 15.8 Elementos posicionados mediante float

CONQUERBLOCKS

Ejemplos



CONQUERBLOCKS

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 20



CONQUERBLOCKS

<Índice>

Clase 20

Clase 20: Flexbox

Flex direction

Align items

Justify content

Flex wrap

Gap

Align self

Order y Flexibilidad

CONQUERBLOCKS

inline vs block vs inline-block

CONQUERBLOCKS

Flexbox

La propiedad display de CSS permite modificar el comportamiento de un elemento HTML, cambiándolo al que le indiquemos, como por ejemplo inline o block (u otro de los que veremos más adelante)

CONQUERBLOCKS

Flexbox

Tipo de caja	Características
block	Se apila en vertical. Ocupa todo el ancho disponible de su etiqueta contenedora.
Versión en línea	
inline	Se coloca en horizontal. Se adapta al ancho de su contenido. Ignora width o height.
inline-block	Combinación de los dos anteriores. Se comporta como inline pero no ignora width o height.

CONQUERBLOCKS

display



CONQUERBLOCKS

display

Block and inline

El comportamiento de los elementos en bloque y en línea es fundamental para CSS y el hecho de que un documento html marcado correctamente será legible por defecto

CONQUERBLOCKS

display

Flujo normal

Este diseño se conoce como “Diseño de bloque y en línea” o Flujo normal” porque esta es la forma en la que los elementos se presentan si no les hacemos nada más.

CONQUERBLOCKS

display

Display con dos valores

inner-outter

Definen dos valores: externo de visualización e interno de visualización.

CONQUERBLOCKS

display

Un elemento que tengan los atributos de bloque podremos establecerle: ancho, alto, padding, margin etc...

CONQUERBLOCKS

Flexbox

CONQUERBLOCKS

Flexbox

En CSS, inicialmente se utilizaba el posicionamiento (static, relative, absolute...), los elementos en línea o en bloque (y derivados) o la propiedad float para realizar maquetaciones

CONQUERBLOCKS

Flexbox

Flex (también llamado flexbox) es un sistema de elementos flexibles que llega con la idea de olvidar estos mecanismos y acostumbrarnos a una mecánica más potente, limpia y personalizable, en la que los elementos HTML se adaptan y colocan automáticamente y es más fácil personalizar los diseños de una página web.

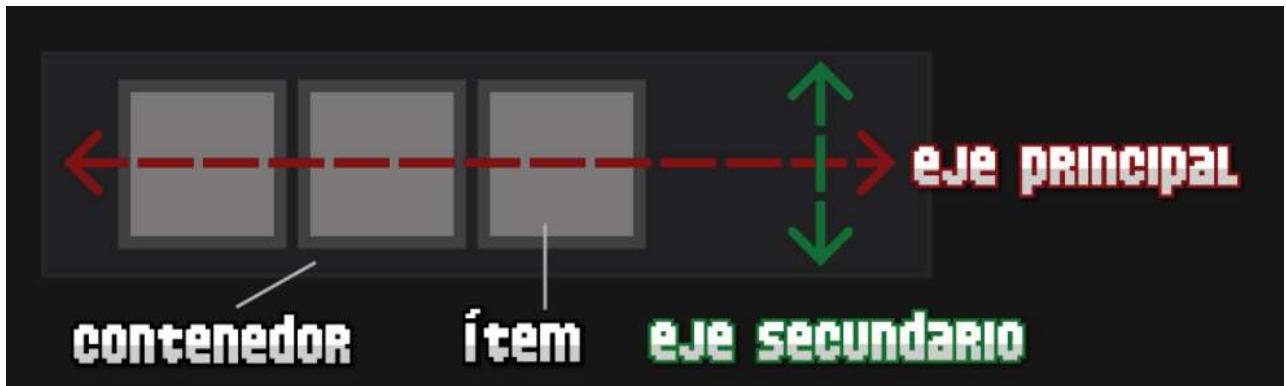
CONQUERBLOCKS

Flexbox

Para empezar a utilizar flex lo primero que debemos hacer es conocer algunos de los elementos básicos de este nuevo esquema, que son los siguientes

CONQUERBLOCKS

Flexbox



CONQUERBLOCKS

Flexbox

- **Contenedor:** Es el elemento padre que tendrá en su interior cada uno de los ítems flexibles. Observa que al contrario que muchas otras estructuras CSS, por norma general, en Flex establecemos las propiedades al elemento padre.
- **Ítem:** Cada uno de los hijos que tendrá el contenedor en su interior.

CONQUERBLOCKS

Flexbox

- **Eje principal:** Los contenedores flexibles tendrán una orientación principal específica. Por defecto, el eje principal del contenedor flex es en horizontal (en fila).
- **Eje secundario:** De la misma forma, los contenedores flexibles tendrán una orientación secundaria, perpendicular a la principal. Si la principal es en horizontal, la secundaria será en vertical (y viceversa).

CONQUERBLOCKS

Flexbox

```
<div class="container"> ← Flex container →  
  <div class="item item-1">1</div> ← Flex items →  
  <div class="item item-2">2</div>  
  <div class="item item-3">3</div>  
</div>
```

CONQUERBLOCKS

Flexbox

Ahora vamos a ver qué podemos hacer con esos items y controlar cómo se forman

CONQUERBLOCKS

Dirección de los ejes flex-direction

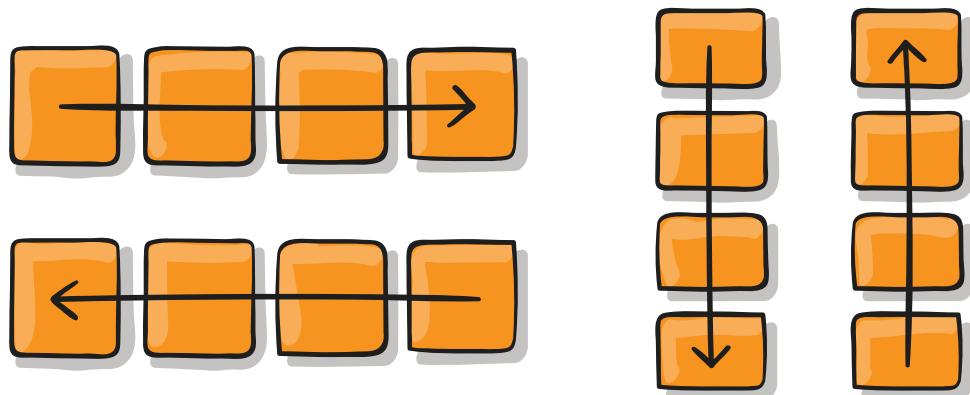
CONQUERBLOCKS

flex-direction

Existen dos propiedades principales para manipular la dirección y comportamiento de los ítems a lo largo del eje principal del contenedor.

CONQUERBLOCKS

Posicionamiento absoluto



CONQUERBLOCKS

Contenedor multilínea flex-wrap

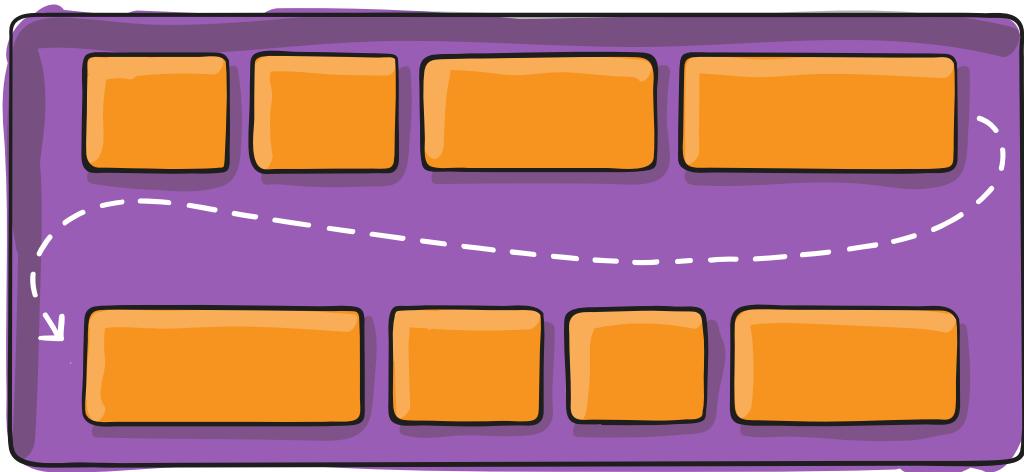
CONQUERBLOCKS

flex-wrap

¿Qué pasa si el contenido ocupa más del ancho?

CONQUERBLOCKS

flex-wrap



CONQUERBLOCKS

Huecos
gap

CONQUERBLOCKS

Gap

Separación entre los items

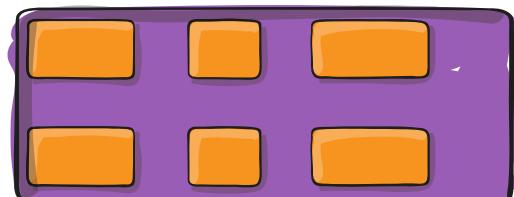
CONQUERBLOCKS

Gap

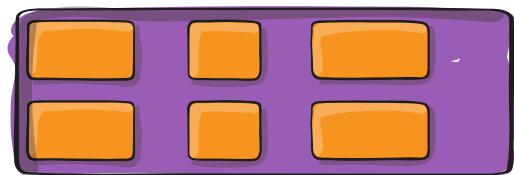
gap: 10px



gap: 30px



gap: 10px 30px

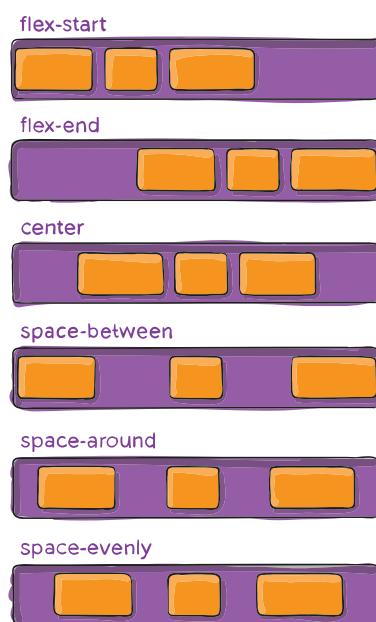


CONQUERBLOCKS

Alineación eje-x justify-content

CONQUERBLOCKS

justify-content



CONQUERBLOCKS

Alineación eje-y align-items

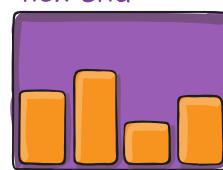
CONQUERBLOCKS

align-items

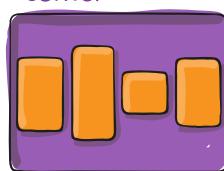
flex-start



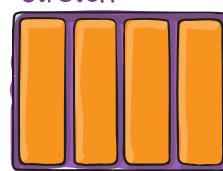
flex-end



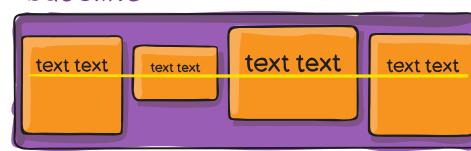
center



stretch



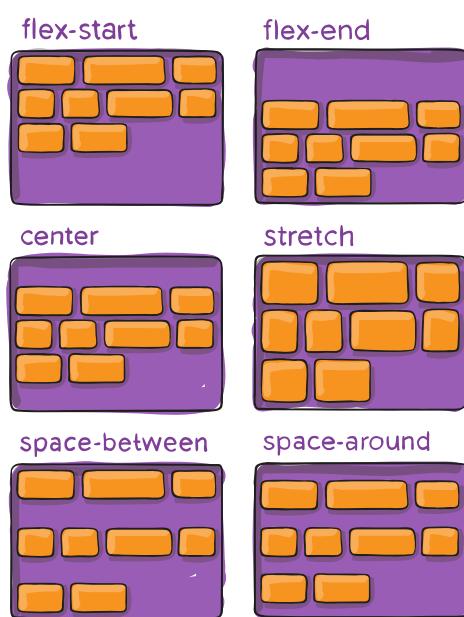
baseline



CONQUERBLOCKS

Alineación contenedor multilinea align-content

CONQUERBLOCKS



CONQUERBLOCKS

Alineación específica align-self

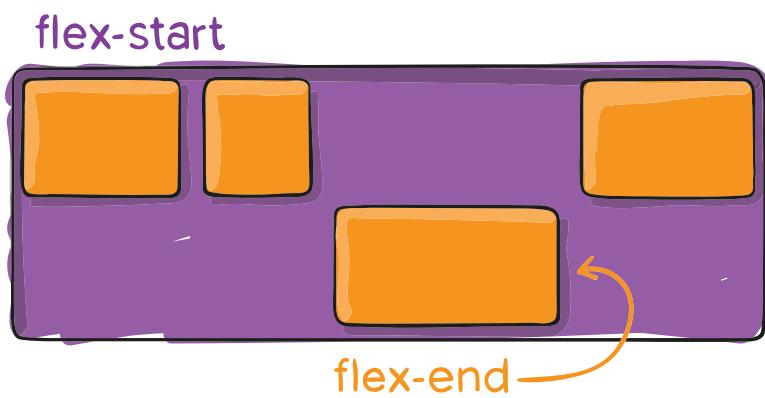
CONQUERBLOCKS

align-self

Esta propiedad es de los hijos
no del contenedor

CONQUERBLOCKS

align-self



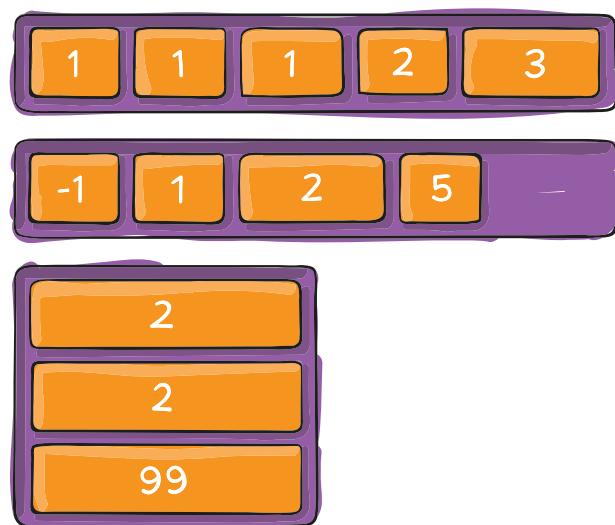
CONQUERBLOCKS

Orden de elementos order

CONQUERBLOCKS

Esta propiedad es de los hijos
no del contenedor

CONQUERBLOCKS



CONQUERBLOCKS

Flexibilidad flex-basis

CONQUERBLOCKS

Esta propiedad es de los hijos
no del contenedor

CONQUERBLOCKS

En primer lugar, tenemos la propiedad flex-basis, que define el tamaño base por defecto que tendrán los ítems antes de aplicarle una cierta distribución de espacio.

CONQUERBLOCKS

También se establece el tamaño antes de repartir el espacio disponible

CONQUERBLOCKS

Flexibilidad flex-grow

CONQUERBLOCKS

Esta propiedad es de los hijos
no del contenedor

CONQUERBLOCKS

La propiedad flex-grow actúa en situaciones donde:

- Hay un flex-basis definido.
- Los ítems cubren el tamaño total del contenedor flex padre.

CONQUERBLOCKS

En esas situaciones, la propiedad flex-grow indica el factor de crecimiento de los ítems en el caso de que no tengan un ancho o alto específico. Modifiquemos el ejemplo anterior y añadamos el siguiente fragmento de código

CONQUERBLOCKS

Por omisión, todos los elementos tienen un `flex-grow: 0` definido, de modo que no hay factor de crecimiento en el elemento, y tendrá el tamaño definido por la propiedad `flex-basis` que explicamos anteriormente.

CONQUERBLOCKS

Sin embargo, si colocamos un `flex-grow: 1` al primer hijo (ejemplo anterior), este crecerá hasta que la suma de los hijos ocupen el 100% del contenedor, mientras que el resto de hijos tendrá el tamaño base definido por `flex-basis`.

CONQUERBLOCKS

En resumen, flex-grow establece un factor de crecimiento observando en conjunto el resto de elementos, y veremos que actua siempre y cuando la suma del espacio de los elementos hijos no superen el 100% del contenedor padre.

CONQUERBLOCKS

Flexibilidad flex-shrink

CONQUERBLOCKS

Esta propiedad es de los hijos
no del contenedor

CONQUERBLOCKS

Por otro lado, tenemos la propiedad flex-shrink que es la opuesta a la propiedad flex-grow. Mientras que la anterior indica un factor de crecimiento, flex-shrink hace justo lo contrario: aplica un factor de decrecimiento.

CONQUERBLOCKS

En esas situaciones, la propiedad flex-grow indica el factor de crecimiento de los ítems en el caso de que no tengan un ancho o alto específico. Modifiquemos el ejemplo anterior y añadamos el siguiente fragmento de código

CONQUERBLOCKS

La propiedad flex-shrink actúa en situaciones donde:

- Hay un flex-basis definido.
- Los ítems no cubren el tamaño total del contenedor flex padre.

CONQUERBLOCKS

Sin embargo, si colocamos un `flex-grow: 1` al primer hijo (ejemplo anterior), este crecerá hasta que la suma de los hijos ocupen el 100% del contenedor, mientras que el resto de hijos tendrá el tamaño base definido por `flex-basis`.

CONQUERBLOCKS

En resumen, `flex-grow` establece un factor de crecimiento observando en conjunto el resto de elementos, y veremos que actua siempre y cuando la suma del espacio de los elementos hijos no superen el 100% del contenedor padre.

CONQUERBLOCKS

Apilamiento con z-index

CONQUERBLOCKS

z-index

Cuando modificamos la posición de las cajas, puede ocurrir apilamiento entre los diferentes elementos

CONQUERBLOCKS

z-index

z-index

establece el nivel de profundidad en el que está un elemento sobre los demás. De esta forma, podemos hacer que un elemento se coloque encima o debajo de otro.

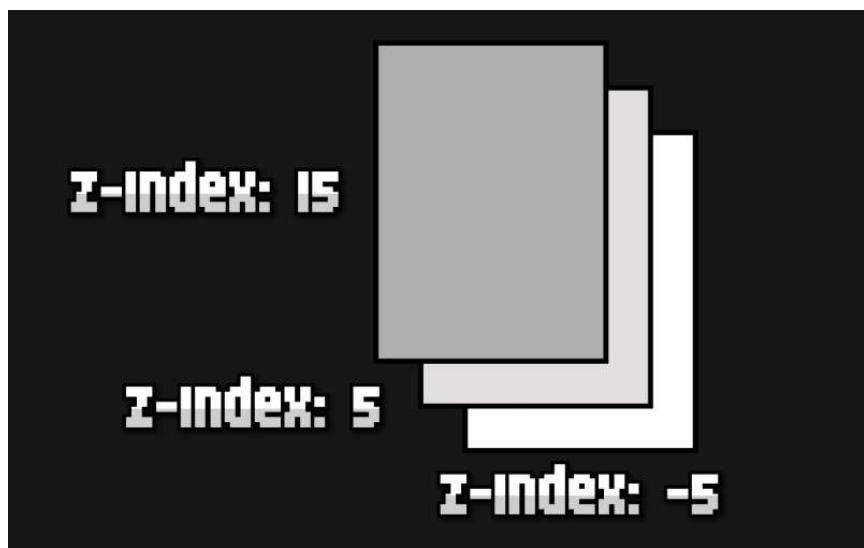
CONQUERBLOCKS

z-index

Su funcionamiento es muy sencillo, sólo hay que indicar un número que representará el nivel de profundidad del elemento. Los elementos un número más alto estarán por encima de otros con un número más bajo, que permanecerán ocultos detrás de los primeros.

CONQUERBLOCKS

z-index



CONQUERBLOCKS

z-index

Nota: Los niveles z-index, así como las propiedades top, left, bottom y right no funcionan con elementos que estén utilizando posicionamiento estático. **Deben tener un tipo de posicionamiento diferente a estático.**

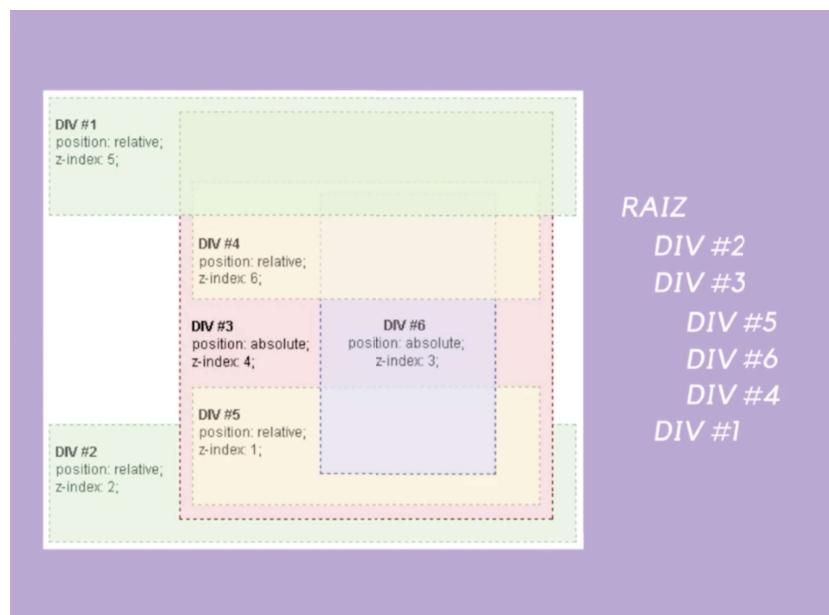
CONQUERBLOCKS

z-index

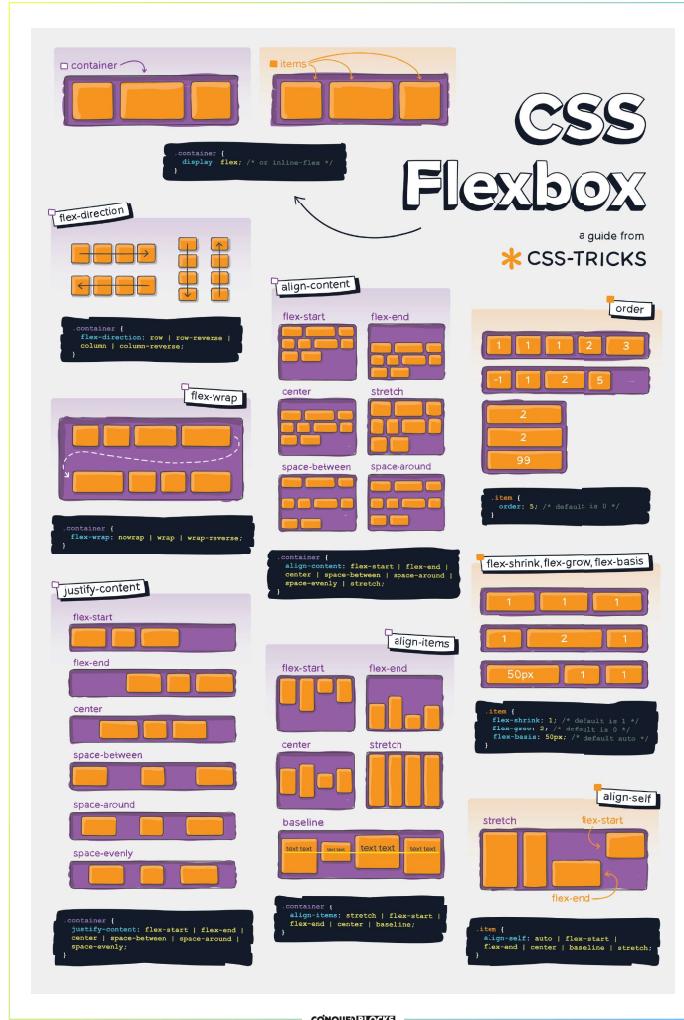
Nota: Los niveles z-index, así como las propiedades top, left, bottom y right no funcionan con elementos que estén utilizando posicionamiento estático. **Deben tener un tipo de posicionamiento diferente a estático.**

CONQUERBLOCKS

z-index



CONQUERBLOCKS



CONQUERBLOCKS

Recursos

RECURSOS

Manz: [Introducción a Flexbox](#)

CSS Tricks: [La guía definitiva de Flexbox](#)

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 23



CONQUERBLOCKS

<Índice>

Clase 23

Posicionamiento Grid Capítulo 1

Display Grid

Grid Template Columns y Rows

Gap

Order

Fr y Repeat

CONQUERBLOCKS

Display Grid

CONQUERBLOCKS

Display Grid

El sistema de elementos flexibles Flex es una gran mejora, sin embargo, está orientado a estructuras de una sola dimensión

CONQUERBLOCKS

Display Grid

Con el paso del tiempo, muchos frameworks CSS y librerías comenzaron a utilizar un sistema basado en un grid donde se definía una cuadrícula, a la que era posible darle tamaño, posición o colocación, cambiando el nombre de sus clases.

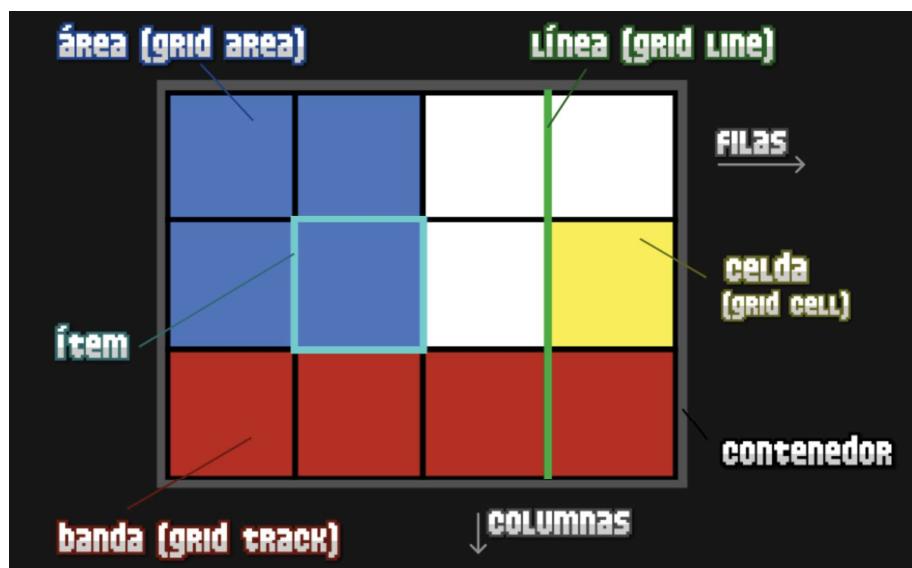
CONQUERBLOCKS

Display Grid

Grid toma la filosofía (y muchas de las bases y conceptos) que se utilizan en **Flexbox**

CONQUERBLOCKS

Display Grid



CONQUERBLOCKS

Display Grid

- Contenedor: El elemento padre contenedor que definirá la cuadrícula o rejilla.
- Ítem: Cada uno de los hijos que contiene la cuadrícula (elemento contenedor).
- Celda (grid cell): Cada uno de los cuadritos (unidad mínima) de la cuadrícula.

CONQUERBLOCKS

Display Grid

- Área (grid area): Región o conjunto de celdas de la cuadrícula.
- Banda (grid track): Banda horizontal o vertical de celdas de la cuadrícula.
- Línea (grid line): Separador horizontal o vertical de las celdas de la cuadrícula.

CONQUERBLOCKS

Display Grid

```
<div class="grid"> ← contenedor →  
  <div class="item item-1">Item 1</div> ← cada uno de los items del grid →  
  <div class="item item-2">Item 2</div>  
  <div class="item item-3">Item 3</div>  
  <div class="item item-4">Item 4</div>  
</div>
```

CONQUERBLOCKS

Grid Template Columns y Grid Template Rows

CONQUERBLOCKS

Grid Template

En Grid CSS, la forma principal de definir una cuadrícula es indicar el tamaño de sus filas y sus columnas de forma explícita. Para ello, sólo tenemos que usar las propiedades CSS `grid-template-columns` y `grid-template-rows`

CONQUERBLOCKS

Grid Template

Propiedad	Valor	Descripción
grid-template-columns	[col1] [col2] ...	Establece el size de cada columna (col_1, col_2...).
grid-template-rows	[fila1] [fila2] ...	Establece el size de cada fila (fila_1, fila_2...).

```
.grid {  
  display: grid;  
  grid-template-columns: 50px 300px;  
  grid-template-rows: 200px 75px;  
}
```

CONQUERBLOCKS

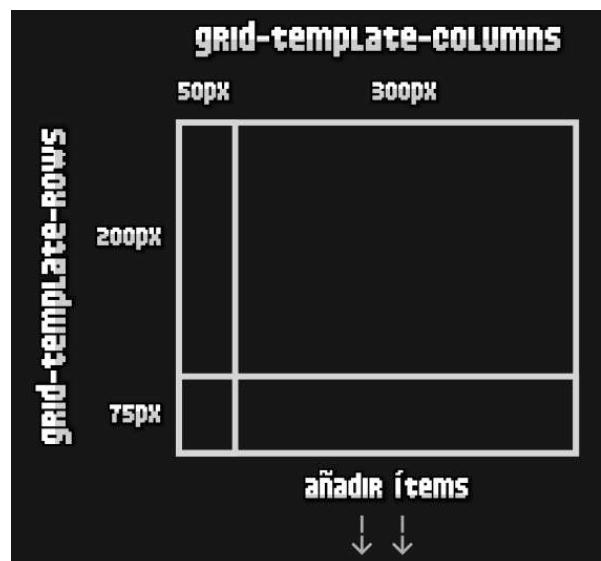
Grid Template Columns

Con la propiedad `display: grid` definimos que queremos crear un grid, y mediante las propiedades `grid-template-columns` y `grid-template-rows` definimos los tamaños de las columnas y las filas del mismo. Esto significa que, a priori, tendríamos una cuadricula o grid de 4 celdas en total

CONQUERBLOCKS

Grid Template Columns

```
.grid {  
  display: grid;  
  grid-template-columns: 50px 300px;  
  grid-template-rows: 200px 75px;  
}
```



CONQUERBLOCKS

Grid Template Columns

Ahora, ten en cuenta que corre de nuestra cuenta vigilar que el número de elementos hijos en el grid es el que debería.

CONQUERBLOCKS

Grid Template Columns

A medida que fueramos incluyendo más ítems en el grid, podríamos aumentar también el número de parámetros de la propiedad grid-template-columns y/o la propiedad grid-template-rows.

CONQUERBLOCKS

Grid Template Columns

En caso de tener más ítems de lo que se indica en la propiedad, los ítems restantes se incluirían sin formato. De tener menos, simplemente se ocuparían los ítems implicados.

CONQUERBLOCKS

Grid Template Columns

Unidades relativas y absolutas
Propiedad shorthand

CONQUERBLOCKS

Grid Template Columns

Demo

CONQUERBLOCKS

Fr y Repeat

CONQUERBLOCKS

Fr y Repeat

En el ejemplo anterior he utilizado píxels como unidades de las celdas de la cuadrícula, sin embargo, también podemos utilizar otras unidades: porcentajes, la palabra clave auto (que obtiene el tamaño restante) o la unidad especial de grid fr (fracción restante), que explicaremos a continuación.

CONQUERBLOCKS

Fr y Repeat

- Dos columnas: Mismo tamaño de ancho para cada una.
- Dos filas: La primera fila ocupará el doble (2fr) que la segunda fila (1fr).

```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  grid-template-rows: 2fr 1fr;  
}
```

CONQUERBLOCKS

Fr y Repeat



CONQUERBLOCKS

Fr y Repeat

De esta forma, es muy fácil predecir el espacio que va a ocupar la cuadrícula, ya que sólo tenemos que sumar todas las unidades para saber el tamaño total, y comparar con cada columna o fila para saber como de grande o pequeña es respecto al total. Así tendremos un mejor control del espacio restante de la cuadrícula, y resultará más intuitivo calcularlo.

CONQUERBLOCKS

Fr y Repeat

Se pueden combinar varias unidades diferentes, como por ejemplo píxeles (px), fracciones restantes (fr), porcentajes (%) y otras combinaciones similares.

CONQUERBLOCKS

Fr y Repeat

Repeat

CONQUERBLOCKS

Fr y Repeat

En algunos casos, en las propiedades grid-template-columns y grid-template-rows podemos necesitar indicar las mismas cantidades un número alto de veces, resultando repetitivo y molesto escribirlas varias veces. Se puede utilizar la función repeat() para indicar repetición de valores, señalando el número de veces que se repiten y el tamaño en cuestión.

CONQUERBLOCKS

Fr y Repeat

```
.grid {  
  display: grid;  
  
  grid-template-columns: 100px repeat(4, 50px) 200px;  
  grid-template-rows: repeat(2, 1fr 2fr);  
  
  /* Equivalente a... */  
  grid-template-columns: 100px 50px 50px 50px 50px 200px;  
  grid-template-rows: 1fr 2fr 1fr 2fr;  
}
```

CONQUERBLOCKS

Fr y Repeat

Asumiendo que tuvieramos un contenedor grid con 24 ítems hijos en el HTML, el ejemplo anterior crearía una cuadrícula con 6 columnas y 4 filas. Recuerda que en el caso de tener más ítems hijos, el patrón se seguiría repitiendo.

CONQUERBLOCKS

Fr y Repeat

Minmax

Si establecemos un rango, por ejemplo, grid-template-columns: minmax(200px, 500px), estaremos indicando que la celda de columna indicada, tendrá un tamaño de 500px, salvo que redimensionemos la ventana del navegador y la hagamos más pequeña, en cuyo caso, el tamaño de la celda podría ir disminuyendo hasta 200px, medida en la cuál se quedaría como mínimo.

CONQUERBLOCKS

Fr y Repeat

Demo

```
<div class="container">
  <div class="item item-1">Item 1</div>
  <div class="item item-2">Item 2</div>
  <div class="item item-3">Item 3</div>
  <div class="item item-4">Item 4</div>
</div>

<style>
.container {
  display: grid;
  grid-template-columns: repeat(2, minmax(400px, 600px));
  grid-template-rows: repeat(2, 1fr);
  gap: 5px;
}

.item {
  background: black;
  color: white;
  padding: 1em;
}
</style>
```

CONQUERBLOCKS

Gap

CONQUERBLOCKS

Gap

Por defecto, la cuadrícula tiene todas sus celdas pegadas a sus celdas contiguas. Aunque sería posible darle un margin a las celdas dentro del contenedor, existe una forma más apropiada: los huecos (gutters).

CONQUERBLOCKS

Gap

Para especificar los huecos (espacio entre celdas) podemos utilizar las propiedades column-gap y/o row-gap. En ellas indicaremos el tamaño de dichos huecos

CONQUERBLOCKS

Gap

Propiedad	Descripción
column-gap	Establece el SIZE de los huecos entre columnas (líneas verticales).
row-gap	Establece el SIZE de los huecos entre filas (líneas horizontales).

CONQUERBLOCKS

Gap

```
.grid {  
  column-gap: 100px;  
  row-gap: 10px;  
}
```



CONQUERBLOCKS

Order

CONQUERBLOCKS

Order

Funciona exactamente igual que como funciona en flex. Es una propiedad mediante la cual podemos modificar y establecer un orden de los elementos mediante números que actuarán como «peso» del elemento

CONQUERBLOCKS

Order

Por defecto, todos los elementos hijos de un contenedor flex tienen establecido un order por defecto al valor 0. Si indicamos una propiedad order con un valor numérico diferente, recolocará los ítems según dicho número, colocando antes los elementos con un número order más pequeño (incluso números negativos) y los elementos con números más altos después.

CONQUERBLOCKS

Order

Demo

CONQUERBLOCKS

Áreas

CONQUERBLOCKS

Los grids por áreas no son una alternativa a los grids explícitos (definidos por filas y columnas). Ambos pueden trabajar conjuntamente o por separado, según interese.

CONQUERBLOCKS

Mediante los Grids por área es posible indicar el nombre y posición concreta de cada área de una cuadrícula. Para ello utilizaremos la propiedad grid-template-areas en nuestro contenedor padre, donde debemos especificar el orden de las áreas en la cuadrícula.

CONQUERBLOCKS

Posteriormente, en cada ítem hijo, utilizamos la propiedad grid-area para indicar el nombre del área del que se trata y que el navegador pueda identificarlas:

Propiedad	Descripción
grid-template-areas	Indica la disposición de las áreas en el grid. Cada texto entre comillas simboliza una fila.
grid-area	Indica el nombre del área. Se usa sobre ítems hijos del grid.

CONQUERBLOCKS

```
<div class="container">
  <div class="item item-1"></div>
  <div class="item item-2"></div>
  <div class="item item-3"></div>
  <div class="item item-4"></div>
</div>

<style>
.container {
  display: grid;
  grid-template-areas: "head head"
                      "menu main"
                      "foot foot";
}

.item-1 { grid-area: head; background: blue; }
.item-2 { grid-area: menu; background: red; }
.item-3 { grid-area: main; background: green; }
.item-4 { grid-area: foot; background: orange; }
</style>
```

- El Item 1 sería nuestra cabecera (head), que ocuparía la primera fila (toda la parte superior).
- El Item 2 sería nuestro menú lateral (menu), que ocuparía el área izquierda del grid (debajo de la cabecera).
- El Item 3 sería nuestro contenido (main), que ocuparía el área derecha del grid (debajo de la cabecera).
- El Item 4 sería nuestro pie de cuadricula (foot), que ocuparía la última fila (área inferior del grid).

CONQUERBLOCKS



CONQUERBLOCKS

Cada una de estas filas se definen como un donde indicaremos el nombre de un área que posteriormente definiremos en nuestro código CSS. Cada fila puede tener ninguna o varias áreas que habría que separar por espacio. A continuación veremos algunos ejemplos de los valores que podemos indicar en esta propiedad y su significado

Valores	Descripción
"none"	Indica que no se creará ninguna plantilla de áreas.
"head"	Indica que se creará una fila de una columna con el área head.
"head menu"	Indica que se creará una fila de 2 columnas con el área head en una y el área menu en otra.
"head head"	Indica que se creará una fila de 2 columnas con el área head ocupando ambas.
". ."	Indica que se colocará una celda sin nombre (nula) en esta posición.

CONQUERBLOCKS

Demo

CONQUERBLOCKS

Ejemplos

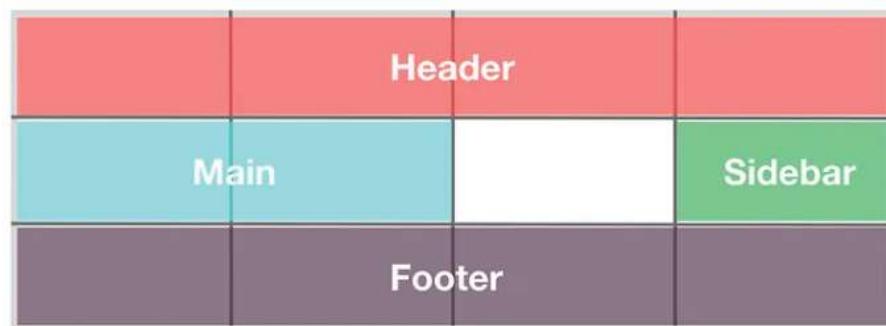
CONQUERBLOCKS

1	2	3
4	5	6
7		

100px	1fr	1fr
100px	1	2
100px	3	4
100px	5	6

50px	100px	
100px	1	2
50px	3	4
50px	5	6

CONQUERBLOCKS



Demo

CONQUERBLOCKS



Recursos

RECURSOS

Manz: [Introducción a Grid](#)

CSS Tricks: [La guía completa de Grid](#)

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 24



CONQUERBLOCKS

<Índice>

Clase 24

Posicionamiento Grid Capítulo 2

Recordemos lo visto

Resolvamos los ejemplos de la clase anterior

Función min-max

Auto-fill

Auto-fit

Alineación

CONQUERBLOCKS

Recordemos lo visto

CONQUERBLOCKS

Función minmax()

CONQUERBLOCKS

minmax

La función minmax() se puede utilizar como valor para definir rangos flexibles de celda.

Funciona de la siguiente forma

CONQUERBLOCKS

minmax

Función	Descripción
minmax(SIZE min, SIZE max)	Define un rango entre min y max .

CONQUERBLOCKS

minmax

Si establecemos un rango, por ejemplo, grid-template-column: minmax(200px, 500px), estaremos indicando que la celda de columna indicada, tendrá un tamaño de 500px, salvo que redimensionemos la ventana del navegador y la hagamos más pequeña, en cuyo caso, el tamaño de la celda podría ir disminuyendo hasta 200px, medida en la cuál se quedaría como mínimo.

CONQUERBLOCKS

minmax

```
<div class="container">
  <div class="item item-1">Item 1</div>
  <div class="item item-2">Item 2</div>
  <div class="item item-3">Item 3</div>
  <div class="item item-4">Item 4</div>
</div>

<style>
.container {
  display: grid;
  grid-template-columns: repeat(2, minmax(400px, 600px));
  grid-template-rows: repeat(2, 1fr);
  gap: 5px;
}

.item {
  background: black;
  color: white;
  padding: 1em;
}
</style>
```

Comprobarás que las celdas se hacen más pequeñas hasta un punto en el que se alcanza el mínimo.

CONQUERBLOCKS

minmax

Demo

CONQUERBLOCKS

Auto-fill

CONQUERBLOCKS

Auto-fill

En la función repeat() es posible utilizar las palabras claves auto-fill o auto-fit para indicar al navegador que queremos que rellene o ajuste el contenedor grid con múltiples elementos hijos dependiendo del tamaño del viewport

CONQUERBLOCKS

Auto-fill

Es decir, si utilizamos
repeat(auto-fill, minmax(300px, 1fr))
el navegador se va a encargar de que los
elementos hijos con el tamaño mínimo quepan en
la primera fila, y los que no quepan, se desplacen
a las siguientes filas del grid, de modo que se
aproveche lo mejor posible el contenedor

CONQUERBLOCKS

Auto-fill

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));  
  background: grey;  
  gap: 10px;  
}  
  
.item {  
  background: blue;  
  color: #fff;  
  font-size: 2rem;  
}
```

CONQUERBLOCKS

Auto-fill

Otra forma de explicarlo

Esta palabra clave nos ayuda a decirle al navegador que inserte el número de columnas o filas que sea necesario para llenar el espacio. Podríamos escribir la siguiente línea de código:

```
// grid repeat auto-fill  
  
grid-template-columns: repeat (auto-fill, minmax (150px, 1fr))
```

CONQUERBLOCKS

Auto-fill

Lo que quiere decir la función auto-fill en este caso es que el navegador puede ubicar el número de columnas que quepan en el ancho, mientras que su ancho mínimo sea de 150px. Entonces, cuando la pantalla cambie de tamaño, el navegador modificará automáticamente el número de columnas que haya según el ancho disponible. El máximo 1fr hace que las columnas siempre tengan el mismo ancho una respecto a la otra.

```
// grid repeat auto-fill  
grid-template-columns: repeat(auto-fill, minmax(150px, 1fr))
```

CONQUERBLOCKS

Auto-fill

Luego veremos un demo

CONQUERBLOCKS

Auto-fit

CONQUERBLOCKS

Auto-fit

Si cambiamos el ejemplo anterior a auto-fit no veremos ninguna diferencia. Sin embargo, si por ejemplo cambiamos el valor mínimo de 300px a 50px (de modo que no llegue a cubrir la primera fila completamente), comprobaremos que mientras auto-fill va llenando la fila del grid y deja el resto del espacio libre, auto-fit ajusta el tamaño de los ítems para que cubran el tamaño máximo de la fila.

CONQUERBLOCKS

Auto-fit

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));  
  background: grey;  
  gap: 10px;  
}  
  
.item {  
  background: blue;  
  color: #fff;  
  font-size: 2rem;  
}
```

CONQUERBLOCKS

Auto-fit

Otra forma de explicarlo

La diferencia entre las funciones auto-fill y auto-fit en CSS Grid es que la primera pondrá el número de columnas que quepan, sin importar el número de elementos a ubicar. Esto quiere decir que, cuando usamos auto-fill, aun si tenemos pocos elementos, este seguirá poniendo columnas aunque estén vacías si el espacio es mayor a la suma de sus anchos.

CONQUERBLOCKS

Auto-fit

La función auto-fit, en cambio, ajusta las columnas para ocupar todo el espacio disponible, sin dejar espacio restante. Ten presente que esta función respeta los anchos mínimos, al igual que auto-fill.

CONQUERBLOCKS

Auto-fit

Demo

CONQUERBLOCKS

Alineación

CONQUERBLOCKS

Alineación

Existen una serie de propiedades que se pueden utilizar para colocar y ajustar nuestra cuadrícula grid o ajustar los ítems a lo largo de ella, de forma sencilla y cómoda. Algunas de estas propiedades probablemente ya las conocerás del módulo CSS flex, sin embargo, en grid pueden tener un comportamiento diferente.

CONQUERBLOCKS

Alineación

Propiedad	Valores	Afecta a...
justify-items	start end center stretch	Elementos 1
align-items	start end center stretch	Elementos 2
justify-content	start end center stretch space-around space-between space-evenly	Contenido 1
align-content	start end center stretch space-around space-between space-evenly	Contenido 2

CONQUERBLOCKS

Alineación

- **justify-items:** Alinea los elementos (hijos) en horizontal (eje principal) dentro de cada celda.
- **align-items:** Alinea los elementos (hijos) en vertical (eje principal) dentro de cada celda.
- **justify-content:** Alinea el contenido (la cuadricula) en horizontal (eje secundario) en el contenedor padre.
- **align-content:** Alinea el contenido (la cuadricula) en vertical (eje secundario) en el contenedor padre.

CONQUERBLOCKS

Alineación

La primera propiedad, **justify-items** sirve para colocar los ítems de un contenedor grid a lo largo de sus celdas correspondientes, siempre en el eje principal (por defecto, en horizontal). Los valores que puede tomar esta propiedad son los siguientes

CONQUERBLOCKS

Alineación

Valor	Descripción
start	Coloca cada ítem al inicio de su celda en el eje principal.
end	Coloca cada ítem al final de su celda en el eje principal.
center	Coloca cada ítem en el centro de su celda en el eje principal.
stretch	Hace que cada ítem se estire y ocupe todo el espacio disponible de su celda en el eje principal.

CONQUERBLOCKS

Alineación

De forma análoga, la propiedad **align-items** sirve para colocar los ítems de un contenedor grid a lo largo de sus celdas correspondientes, pero en lugar de el eje principal, las coloca en el eje secundario (por defecto, en vertical). Los valores que puede tomar son los mismos que la propiedad anterior.

CONQUERBLOCKS

Alineación

Alineación de contenido

CONQUERBLOCKS

Alineación

La propiedad **justify-content** permite modificar la distribución del contenido de la cuadrícula en su contenedor padre, a lo largo de su eje principal (por defecto, el horizontal). Los valores que puede tomar son los siguientes

CONQUERBLOCKS

Alineación

Valor	Descripción
start	Coloca la cuadrícula en su conjunto al inicio del contenedor padre en su eje principal (horizontal).
end	Coloca la cuadrícula en su conjunto al final del contenedor padre en su eje principal (horizontal).
center	Coloca la cuadrícula en su conjunto al centro del contenedor padre en su eje principal (horizontal).
stretch	Estira la cuadrícula ocupando todo el espacio disponible del contenedor padre en su eje principal (horizontal).
space-between	Establece espacios sólo entre las celdas, en su eje principal (horizontal).
space-around	Establece espacios alrededor de las celdas, en su eje principal (horizontal).
space-evenly	Idem al anterior, pero solapando los espacios, de modo que sean todos de tamaño equivalente.

CONQUERBLOCKS

Alineación

De forma análoga, la propiedad **align-content** sirve para colocar el contenido de la cuadrícula en su contenedor padre, pero a lo largo de su contenedor secundario (por defecto, el vertical). Los valores que puede tomar son exactamente los mismos que la propiedad anterior.

CONQUERBLOCKS

Alineación

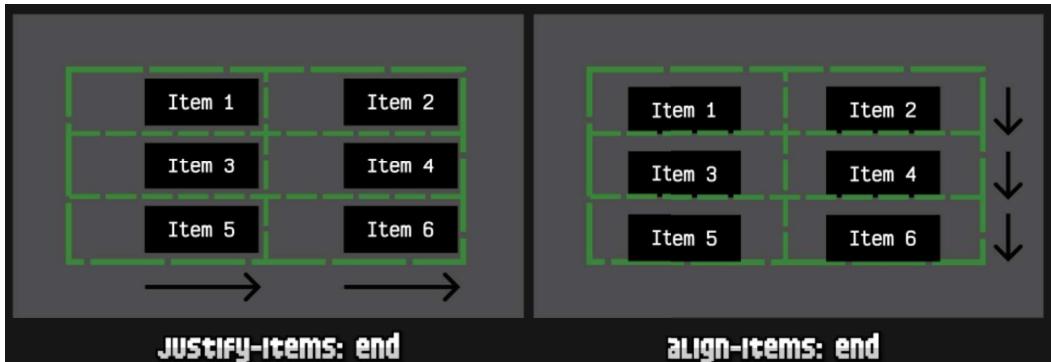
```
<div class="container">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
  <div class="item">Item 4</div>
  <div class="item">Item 5</div>
  <div class="item">Item 6</div>
</div>

<style>
.container {
  display: grid;
  grid-template-columns: repeat(2, 250px);
  grid-template-rows: repeat(3, 50px);
  gap: 10px;
  background: grey;
  height: 300px;
}

.item {
  padding: 10px;
  background: black;
  color: white;
}
</style>
```

CONQUERBLOCKS

Alineación



CONQUERBLOCKS

Alineación



CONQUERBLOCKS

Alineación

Alineaciones específicas

¿Os acordáis de flexbox?

CONQUERBLOCKS

Alineación

Alineaciones específicas

En el caso de que queramos que uno de los ítems hijos tenga una distribución diferente al resto, podemos aplicar en el elemento hijo la propiedad justify-self (eje principal) o align-self (eje secundario) sobreescribiendo su distribución su general, y aplicando una específica.

CONQUERBLOCKS

Alineación

Propiedad	Descripción
justify-self	Altera la alineación del ítem hijo en el eje horizontal y la sobreescribe con la indicada.
align-self	Altera la alineación del ítem hijo en el eje vertical y la sobreescribe con la indicada.

CONQUERBLOCKS

Alineación

Alineaciones específicas

Recuerda que estas propiedades funcionan exactamente igual que sus análogas justify-items o align-items y tienen los mismos valores, sólo que en lugar de indicarse en el elemento padre contenedor, se hace sobre un elemento hijo y repercute en dicho elemento hijo específicamente.

CONQUERBLOCKS

Alineación

Demos

CONQUERBLOCKS

Grid-template

CONQUERBLOCKS

Si acostumbras a utilizar estas propiedades frecuentemente, puedes utilizar la propiedad grid-template, que sirve de atajo para muchas cosas, y una de ellas, resumir en una sola propiedad los valores que tenemos en grid-template-columns y en grid-template-rows

CONQUERBLOCKS

Esta propiedad convierte en un proceso bastante cómodo el crear grids de unas dimensiones concretas de forma resumida.

CONQUERBLOCKS

En el caso de utilizar el valor `none`, las propiedades `grid-template-rows`, `grid-template-columns` y la propiedad `grid-template-areas`, se establecen a sus valores por defecto, desactivando su funcionamiento.

CONQUERBLOCKS

Propiedad	Valores	Descripción
<code>grid-template</code>	<code>none</code> <code>grid-template-rows</code> / <code>grid-template-columns</code>	Atajo para definir dimensiones del grid.

CONQUERBLOCKS

Demo

CONQUERBLOCKS

Esto no es todo...

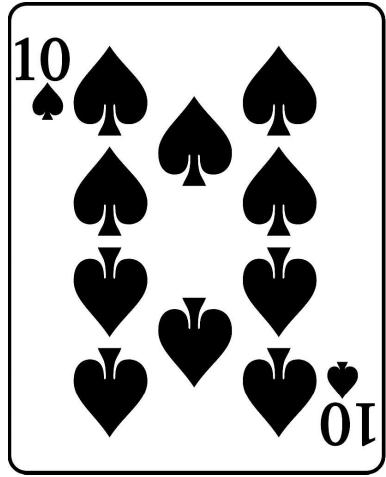
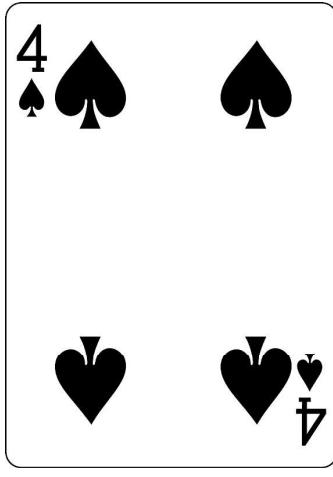
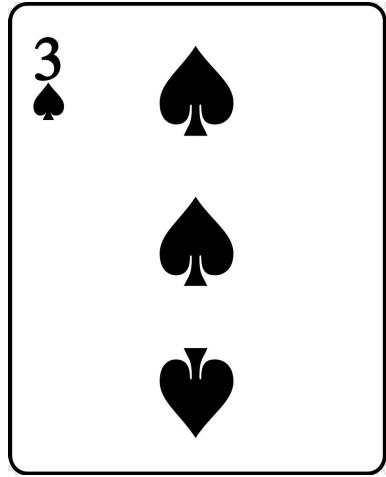
CONQUERBLOCKS

Juego de jardín

CONQUERBLOCKS

¿Quieres entretenerte?

CONQUERBLOCKS



Tendrás que buscar información sobre: [transform: rotate\(180deg\)](#)

[Imagen de la pica](#)

CONQUERBLOCKS

The screenshot shows a user interface for a game or application. On the left, there are two sets of orange blocks labeled 'space-around' and 'space-between'. On the right, there are four sets of orange blocks labeled 'start', 'end', 'center', and 'stretch'. Below these examples, there is a code editor window containing the following CSS:

```
container {
  justify-content: start | end | center
  stretch | space-around | space-between
  space-evenly
}

container {
  align-items: start | end |
  center | stretch;
}
```

CONQUERBLOCKS

Recursos

RECURSOS

Manz: [Introducción a Grid](#)

CSS Tricks: [La guía completa de Grid](#)

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

[@bienvenidosaez](#)

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 27a



CONQUERBLOCKS

<Índice>

Clase 27

Estilando tablas y listas

Tablas con estilo

Tablas con estilo avanzadas

Listas no numeradas con estilo

Listas numeradas con estilo

CONQUERBLOCKS

Tablas con estilo

CONQUERBLOCKS

Tablas con estilo

Ejemplo 1

NAME (DESC)	WINS	DRAWS	LOSSES	TOTAL
Dick	1	1	2	3
Harry	0	2	2	2
Tom	2	0	1	5

CONQUERBLOCKS

Tablas con estilo

Ejemplo 2

Rank	Name	Points	Team
1	Domenic	88,110	dcode
2	Sally	72,400	Students
3	Nick	52,300	dcode

CONQUERBLOCKS

Tablas con estilo

Ejemplo 3

2019 Fourth Quarter Report

	OCTOBER	NOVEMBER	DECEMBER
Projected	\$820,180	\$841,640	\$732,270
Actual	\$850,730	\$892,580	\$801,240
Utilization	83%	90%	75%

CONQUERBLOCKS

Tablas con estilo avanzado

CONQUERBLOCKS

Tablas con estilo avanzado

Table #09

INVOICE	CUSTOMER	SHIP	PRICE	PURCHASED PRICE	STATUS
1001	Mark Otto	Japan	\$3000	\$1200	Progress
1001	Mark Otto	Japan	\$3000	\$1200	Open
1001	Mark Otto	Japan	\$3000	\$1200	On hold
1001	Mark Otto	Japan	\$3000	\$1200	Progress
1001	Mark Otto	Japan	\$3000	\$1200	On hold
1001	Mark Otto	Japan	\$3000	\$1200	Open
1001	Mark Otto	Japan	\$3000	\$1200	Open
1001	Mark Otto	Japan	\$3000	\$1200	Progress

Ejemplo 5

CONQUERBLOCKS

Tablas con estilo avanzado

Table #8

Order	Name	Occupation	Contact	Education
1392	James Yates	Web Designer Far far away, behind the word mountains	+63 983 0962 971	NY University
4616	Matthew Wasil	Graphic Designer Far far away, behind the word mountains	+02 020 3994 929	London College
9841	Sampson Murphy	Mobile Dev Far far away, behind the word mountains	+01 352 1125 0192	Senior High
9548	Gaspar Semenov	Illustrator Far far away, behind the word mountains	+92 020 3994 929	College
4616	Matthew Wasil	Graphic Designer Far far away, behind the word mountains	+02 020 3994 929	London College
9841	Sampson Murphy	Mobile Dev Far far away, behind the word mountains	+01 352 1125 0192	Senior High
9548	Gaspar Semenov	Illustrator Far far away, behind the word mountains	+92 020 3994 929	College

Ejemplo 6

CONQUERBLOCKS

Tablas con estilo avanzado

Create Your Domain Name

TLD	Duration	Registration	Renewal	Transfer	Register
.com	1 Year	\$70.00	\$5.00	\$5.00	<button>Sign Up</button>
.net	1 Year	\$75.00	\$5.00	\$5.00	<button>Sign Up</button>
.org	1 Year	\$65.00	\$5.00	\$5.00	<button>Sign Up</button>
.biz	1 Year	\$60.00	\$5.00	\$5.00	<button>Sign Up</button>
.info	1 Year	\$50.00	\$5.00	\$5.00	<button>Sign Up</button>
.me	1 Year	\$45.00	\$5.00	\$5.00	<button>Sign Up</button>

Ejemplo 7

CONQUERBLOCKS

Listas con estilo

CONQUERBLOCKS

Listas no numeradas con estilo

Cras justo odio

Dapibus ac facilisis in

Morbi leo risus

Porta ac consectetur ac

Vestibulum at eros

Lista de elementos sencilla

CONQUERBLOCKS

Listas no numeradas con estilo

Cras justo odio
Dapibus ac facilisis in
Morbi leo risus
Porta ac consectetur ac
Vestibulum at eros

Lista de elementos con elemento hover y clase active con similar estilo

CONQUERBLOCKS

Listas no numeradas con estilo

Cras justo odio	14
Dapibus ac facilisis in	2
Morbi leo risus	1

Lista de elementos con badges en un span

CONQUERBLOCKS

Listas no numeradas con estilo



Listas con iconos

Password requirements:

- At least 10 characters
- At least one lowercase character
- At least one special character, e.g., ! @ # ?

CONQUERBLOCKS

Listas no numeradas con estilo

- Bread
- Milk
- Apples
- Grapefruit
- Cat food
- Cheese
- Wine

Listas personalísimas con marker por elemento

CONQUERBLOCKS

Listas no numeradas con estilo

❶ This is the first item

❷ This is the second item

❸ This is the third item

❹ This is the fourth item

❺ This is the fifth item

❻ This is the sixth item

¿Lo intentamos?

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

[@bienvenidosaez](https://www.instagram.com/bienvenidosaez)

Youtube

[youtube.com/bienvenidosaez](https://www.youtube.com/bienvenidosaez)

CONQUERBLOCKS

{CSS}

Clase 27b



CONQUERBLOCKS

<Índice>

Clase 27

Estilando tablas y listas

Tablas con estilo

Tablas con estilo avanzadas

Listas no numeradas con estilo

Listas numeradas con estilo

CONQUERBLOCKS

Tablas con estilo

CONQUERBLOCKS

Tablas con estilo

Ejemplo 1

NAME (DESC)	WINS	DRAWS	LOSSES	TOTAL
Dick	1	1	2	3
Harry	0	2	2	2
Tom	2	0	1	5

CONQUERBLOCKS

Tablas con estilo

Ejemplo 2

Rank	Name	Points	Team
1	Domenic	88,110	dcode
2	Sally	72,400	Students
3	Nick	52,300	dcode

CONQUERBLOCKS

Tablas con estilo

Ejemplo 3

2019 Fourth Quarter Report

	OCTOBER	NOVEMBER	DECEMBER
Projected	\$820,180	\$841,640	\$732,270
Actual	\$850,730	\$892,580	\$801,240
Utilization	83%	90%	75%

CONQUERBLOCKS

Tablas con estilo avanzado

CONQUERBLOCKS

Tablas con estilo avanzado

Table #09

INVOICE	CUSTOMER	SHIP	PRICE	PURCHASED PRICE	STATUS
1001	Mark Otto	Japan	\$3000	\$1200	Progress
1001	Mark Otto	Japan	\$3000	\$1200	Open
1001	Mark Otto	Japan	\$3000	\$1200	On hold
1001	Mark Otto	Japan	\$3000	\$1200	Progress
1001	Mark Otto	Japan	\$3000	\$1200	On hold
1001	Mark Otto	Japan	\$3000	\$1200	Open
1001	Mark Otto	Japan	\$3000	\$1200	Open
1001	Mark Otto	Japan	\$3000	\$1200	Progress

Ejemplo 5

CONQUERBLOCKS

Tablas con estilo avanzado

Table #8

Order	Name	Occupation	Contact	Education
1392	James Yates	Web Designer Far far away, behind the word mountains	+63 983 0962 971	NY University
4616	Matthew Wasil	Graphic Designer Far far away, behind the word mountains	+02 020 3994 929	London College
9841	Sampson Murphy	Mobile Dev Far far away, behind the word mountains	+01 352 1125 0192	Senior High
9548	Gaspar Semenov	Illustrator Far far away, behind the word mountains	+92 020 3994 929	College
4616	Matthew Wasil	Graphic Designer Far far away, behind the word mountains	+02 020 3994 929	London College
9841	Sampson Murphy	Mobile Dev Far far away, behind the word mountains	+01 352 1125 0192	Senior High
9548	Gaspar Semenov	Illustrator Far far away, behind the word mountains	+92 020 3994 929	College

Ejemplo 6

CONQUERBLOCKS

Tablas con estilo avanzado

Create Your Domain Name

TLD	Duration	Registration	Renewal	Transfer	Register
.com	1 Year	\$70.00	\$5.00	\$5.00	<button>Sign Up</button>
.net	1 Year	\$75.00	\$5.00	\$5.00	<button>Sign Up</button>
.org	1 Year	\$65.00	\$5.00	\$5.00	<button>Sign Up</button>
.biz	1 Year	\$60.00	\$5.00	\$5.00	<button>Sign Up</button>
.info	1 Year	\$50.00	\$5.00	\$5.00	<button>Sign Up</button>
.me	1 Year	\$45.00	\$5.00	\$5.00	<button>Sign Up</button>

Ejemplo 7

CONQUERBLOCKS

Listas con estilo

CONQUERBLOCKS

Listas con estilo

REPASEMOS
::before
::after
content

CONQUERBLOCKS

Listas con estilo

list-style-image
list-style-position
list-style-type
list-style
::marker
counter()

CONQUERBLOCKS

Listas no numeradas con estilo

Cras justo odio
Dapibus ac facilisis in
Morbi leo risus
Porta ac consectetur ac
Vestibulum at eros

Lista de elementos sencilla

CONQUERBLOCKS

Listas no numeradas con estilo

Cras justo odio
Dapibus ac facilisis in
Morbi leo risus
Porta ac consectetur ac
Vestibulum at eros

Lista de elementos con elemento hover y clase active con similar estilo

CONQUERBLOCKS

Listas no numeradas con estilo

Cras justo odio	14
Dapibus ac facilisis in	2
Morbi leo risus	1

Lista de elementos con badges en un span

CONQUERBLOCKS

Listas no numeradas con estilo



Listas con iconos

Password requirements:

- At least 10 characters
- At least one lowercase character
- At least one special character, e.g., ! @ # ?

CONQUERBLOCKS

Listas no numeradas con estilo

- Bread
- Milk
- Apples
- Grapefruit
- Cat food
- Cheese
- Wine

Listas personalísimas con marker por elemento

CONQUERBLOCKS

Listas no numeradas con estilo

❶ This is the first item

❷ This is the second item

❸ This is the third item

❹ This is the fourth item

❺ This is the fifth item

❻ This is the sixth item

¿Lo intentamos?

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

[@bienvenidosaez](#)

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 28



CONQUERBLOCKS

<Índice>

Clase 28

Interactividad

Outline

Cursor

Pseudoclases de interactividad

Scroll-behavior

Enlaces y estados

CONQUERBLOCKS

Outline

CONQUERBLOCKS

Outline

La familia de propiedades **outline**-* nos permiten modificar el comportamiento del contorno de los elementos

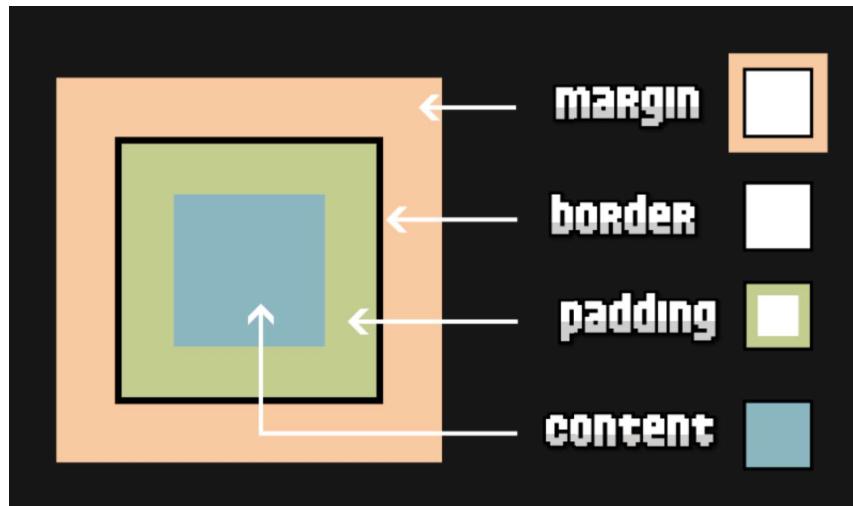
CONQUERBLOCKS

Outline

Recordemos el modelo de cajas

CONQUERBLOCKS

Outline



CONQUERBLOCKS

Outline

Propiedades con outline

Propiedad	Valor	Significado
outline-color	invert COLOR	Invierte el color de la misma.
outline-style	STYLE none	Idem a los bordes. Estilo de la línea divisoria.
outline-width	medium thin thick SIZE	Tamaño de los bordes. Tamaño predefinido o específico.
outline-offset	SIZE	Desplazamiento del contorno.

CONQUERBLOCKS

Outline

Propiedad outline (shorthand)

```
div {  
    /* outline: <color> <style> <width> */  
}
```

CONQUERBLOCKS

Outline

Diferencia entre border y outline

CONQUERBLOCKS

Outline

Posición y Espacio:

- **border:** Forma parte del box model de un elemento. Esto significa que afecta el tamaño total del elemento y puede causar que otros elementos se desplacen. Si un elemento tiene un ancho de 100px y se le da un borde de 10px, el ancho total del elemento será 120px.
- **outline:** No forma parte del box model. Es dibujado alrededor del elemento pero no afecta su tamaño ni desplaza a otros elementos. Un elemento con un ancho de 100px seguirá teniendo ese ancho incluso si se le da un outline de 10px.

CONQUERBLOCKS

Outline

Partes:

- **border**: Puede tener bordes diferentes en cada uno de sus cuatro lados (border-top, border-right, border-bottom, border-left).
- **outline**: Es uniforme y no puede tener estilos diferentes en los diferentes lados. Es una línea que rodea el elemento completo.

CONQUERBLOCKS

Outline

Forma:

- **border**: Sigue la forma del elemento. Si el elemento es un círculo, el borde será un círculo.
- **outline**: Siempre es rectangular, incluso si se aplica a un elemento con bordes redondeados.

CONQUERBLOCKS

Outline

Offset:

- **border**: No tiene una propiedad específica para establecer un desplazamiento.
- **outline**: Puede tener un desplazamiento usando la propiedad outline-offset, que determina cuánto espacio hay entre el borde del elemento y el contorno.

CONQUERBLOCKS

Outline

Colores y estilos:

- **border**: Puede tener diferentes colores y estilos en cada lado.
- **outline**: Tiene un solo color y estilo para todo el contorno.

CONQUERBLOCKS

Outline

Uso común:

- **border:** Es comúnmente usado para definir el borde de cajas, imágenes, botones y otros elementos.
- **outline:** A menudo se usa para resaltar un elemento cuando está enfocado (por ejemplo, al navegar con el teclado), o para aplicar ciertos efectos de diseño sin afectar el layout.

CONQUERBLOCKS

Outline

En resumen, mientras que border es una herramienta versátil para estilizar los bordes de un elemento y afecta el box model, outline es más adecuado para proporcionar un contorno visual que no afecta el layout del documento.

CONQUERBLOCKS

Cursor

CONQUERBLOCKS

Cursor

Un poder conlleva una gran responsabilidad

CONQUERBLOCKS

Cursor

LA PROPIEDAD CURSOR

↳ default	↔ e-resize
⊕ crosshair	↗ ne-resize
↳ help	↖ nw-resize
⊗ move	↑ n-resize
:pointer	↘ se-resize
🕒 progress	↙ sw-resize
I text	↓ s-resize
⌚ wait	↔ w-resize
none	↔ ew-resize
↳ context-menu	↑↓ ns-resize
⊕ cell	↗↖ nesw-resize
→ vertical-text	↖↗ nwse-resize
↳ alias	↔ col-resize
↳ copy	↑↓ row-resize
🖼 url(images/nyan.png)	⊗ all-scroll
🚫 not-allowed	🚫 no-drop
✋ grab	○ grabbing
🔍 zoom-in	🔍 zoom-out

CONQUERBLOCKS

Cursor

Demo

CONQUERBLOCKS

Cursor

Cursos personalizados

CONQUERBLOCKS

Pseudoclases de interactividad

CONQUERBLOCKS

Pseudoclases de interactividad

:hover

- Se aplica cuando el puntero del mouse se coloca sobre un elemento, sin hacer clic en él.
- Ejemplo de uso: Cambiar el color de fondo de un botón cuando el mouse pasa por encima.

CONQUERBLOCKS

Pseudoclases de interactividad

:active:

- Se aplica cuando un elemento está siendo activado por el usuario, generalmente cuando se hace clic en él.
- Ejemplo de uso: Cambiar el color de fondo de un botón mientras se presiona.

CONQUERBLOCKS

Pseudoclases de interactividad

:focus:

- Se aplica a un elemento cuando gana el foco. Esto puede ocurrir, por ejemplo, cuando un usuario hace clic en un campo de entrada o navega a él con el teclado.
- Es especialmente importante para la accesibilidad, ya que indica claramente qué elemento está seleccionado cuando se navega con el teclado.

CONQUERBLOCKS

Pseudoclases de interactividad

:focus:

- Ejemplo de uso: Aplicar un borde resaltado a un campo de entrada cuando está enfocado.

CONQUERBLOCKS

Pseudoclases de interactividad

:focus-within:

- Selecciona el elemento si uno de sus miembros hijos ha ganado el foco.

CONQUERBLOCKS

Pseudoclases de interactividad

:focus:

- Ejemplo de uso: Aplicar un borde resaltado a un campo de entrada cuando está enfocado.

CONQUERBLOCKS

Pseudoclases de interactividad

:visited:

- Se aplica a los enlaces que el usuario ya ha visitado.
- Por razones de privacidad, las propiedades que pueden ser aplicadas con :visited son limitadas. Por ejemplo, no puedes cambiar el background-color de un enlace visitado.
- Ejemplo de uso: Cambiar el color de un enlace después de haber sido visitado.

CONQUERBLOCKS

Pseudoclases de interactividad

:link:

- Se aplica a los enlaces que el usuario no ha visitado.
- Es el estado predeterminado de un enlace antes de que sea visitado.
- Ejemplo de uso: Establecer el color inicial de un enlace.

CONQUERBLOCKS

Pseudoclases de interactividad

Cuidado con los dispositivos móviles

CONQUERBLOCKS

Scroll-behavior

CONQUERBLOCKS

Scroll-behavior

La propiedad scroll-behavior nos permite activar un desplazamiento suave (smooth scroll) al pulsar en un enlace de anclas <a>. Como hemos dicho, por defecto, al pulsar sobre estas anclas, nos desplazamos directa e instantáneamente a nuestro objetivo.

CONQUERBLOCKS

Scroll-behavior

Por defecto, la propiedad indicada es auto. Sin embargo, estableciendo el valor smooth a esta propiedad en el elemento <html>, podemos suavizar este comportamiento, de modo que se realiza de forma progresiva, muy suave:

CONQUERBLOCKS

Scroll-behavior

```
<nav>
  <a href="#slide-1">Slide 1</a> ·
  <a href="#slide-2">Slide 2</a> ·
  <a href="#slide-3">Slide 3</a>
</nav>

<div class="container">
  <div class="slide" id="slide-1">
    <h1>Slide 1</h1>
  </div>
  <div class="slide" id="slide-2">
    <h1>Slide 2</h1>
  </div>
  <div class="slide" id="slide-3">
    <h1>Slide 3</h1>
  </div>
</div>
```

Ejemplo

CONQUERBLOCKS

Scroll-behavior

Demo

CONQUERBLOCKS

Pseudoclases de interactividad en hijos

CONQUERBLOCKS

Order

Estos selectores podemos usarlos para modificar también los hijos de los elementos

CONQUERBLOCKS

Order

Demo

CONQUERBLOCKS

Tarea

CONQUERBLOCKS

Home

Our services

Contact Us

Home

Our services

Contact Us

Marketing

Social Media

Web design

CONQUERBLOCKS

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 29



CONQUERBLOCKS

<Índice>

Clase 29

Degrados, sombras y efectos

Degrados

Sombras en textos

Sombras en cajas

Sombras idénticas

Filtros CSS

Modos de fusión

CONQUERBLOCKS

Degrados

CONQUERBLOCKS

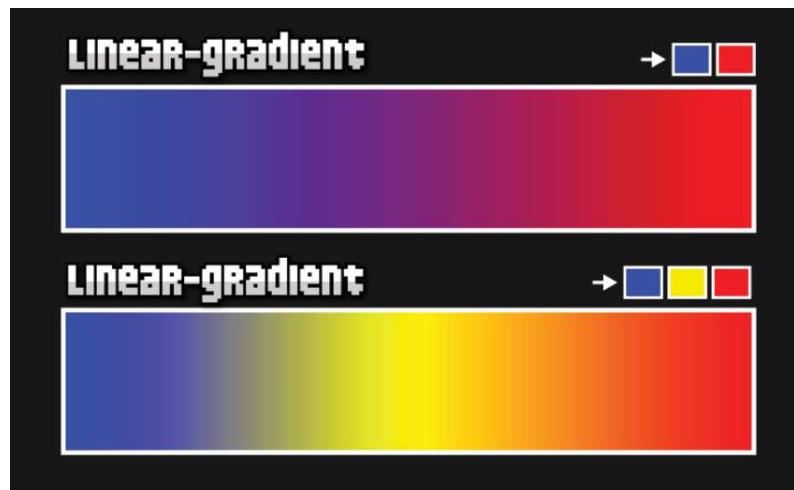
Degrados

Recuerda que podemos hacer
Degrados lineales
Degrados radiales
Degrados cónicos

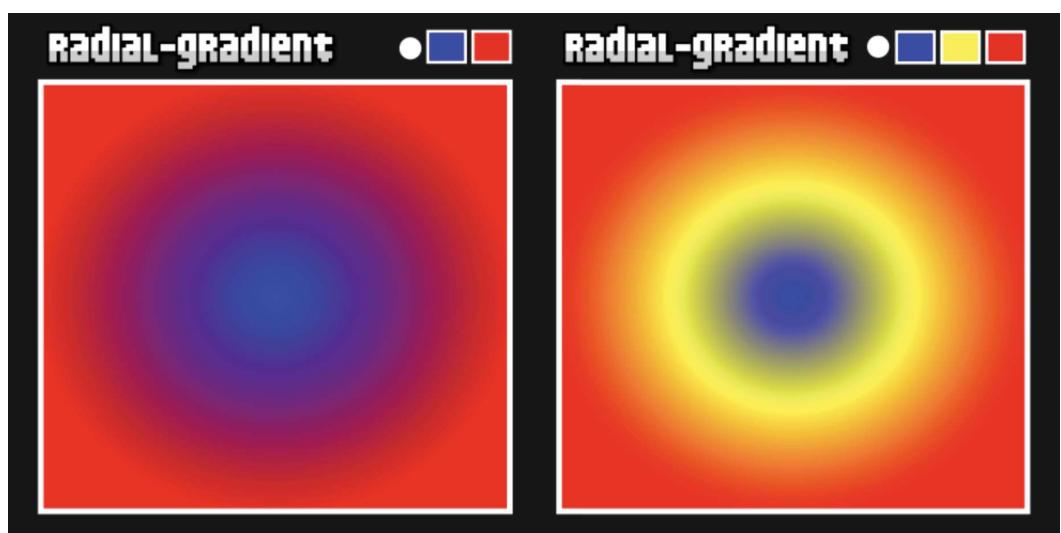
Clase 16

CONQUERBLOCKS

Degradados

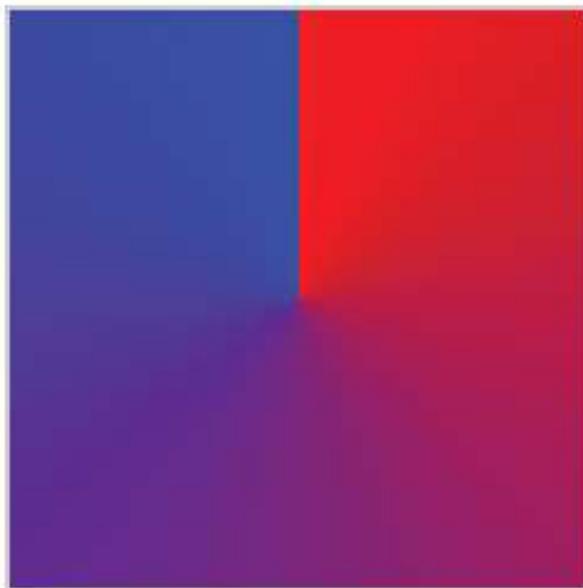


CONQUERBLOCKS



CONQUERBLOCKS

Degradados



CONQUERBLOCKS

Sombras en textos

CONQUERBLOCKS

Sombras en textos

Crear sombras en textos mediante CSS es muy útil, puesto que es una forma interesante de suavizar y hacer más legibles los textos, o simplemente hacerlos más atractivos para el usuario que está viendo la página. Para ello, utilizaremos la propiedad `text-shadow`.

CONQUERBLOCKS

Sombras en textos

Propiedad	Valor	Significado
<code>text-shadow</code>	<code>none</code>	No aplica ninguna sombra en el texto (o la quita si existía previamente).
<code>text-shadow</code>	<code>POSX</code> <code>POSY</code>	Aplica sombra color negro , desplazándola <code>POSX</code> (horizontal) y <code>POSY</code> (vertical).
<code>text-shadow</code>	<code>POSX</code> <code>POSY</code> <code>SIZE</code>	Idem a la anterior, pero establece un desenfoque a la sombra (0 sin desenfoque).
<code>text-shadow</code>	<code>POSX</code> <code>POSY</code> <code>SIZE</code> <code>COLOR</code>	Idem al anterior, pero indicando un <code>COLOR</code> personalizado para la sombra.

CONQUERBLOCKS

Sombras en textos

Desplazamiento

- Para crear una sombra, los parámetros mínimos que hay que especificar son , el desplazamiento horizontal, y , el desplazamiento vertical. Podemos aplicar cantidades negativas

CONQUERBLOCKS

Sombras en textos

Desenfoque

- El tercer parámetro de las sombras en text-shadow hace referencia al desenfoque que queremos sobre dicha sombra. Funciona dandole un tamaño , que por defecto es 0, manteniendo la sombra de una forma idéntica al texto original. Sin embargo, a medida que vamos elevando ese tamaño, la sombra estará más desenfocada o difuminada

CONQUERBLOCKS

Sombras en textos

Color de la sombra

- Por defecto, su valor es currentColor, que generalmente suele ser black.
- Podemos utilizar cualquier color

CONQUERBLOCKS

Sombras en textos

```
.text {  
    text-shadow: 4px 4px;          /* Equivalente a 4px 4px 0 black */  
    text-shadow: 4px 4px 2px;      /* Equivalente a 4px 4px 2px black */  
  
    text-shadow: 4px 4px 5px #00005;  
}
```

CONQUERBLOCKS

Sombras en textos

Demo

CONQUERBLOCKS

Sombras en textos

Sombras de texto múltiples

- Al igual que al utilizar imágenes de fondo, es posible utilizar múltiples sombras de texto. Esto nos dará suficiente flexibilidad para crear textos con efectos, bordes o sombras que resultan de aplicar múltiples sombras de fondo con diferentes parámetros. Para ello, simplemente sepáramos el conjunto de valor con comas

CONQUERBLOCKS

Sombras en textos

```
.text {  
  color: white;  
  font-size: 4rem;  
  text-shadow:  
    -1px -1px 3px #999,  
    0 0 1px darkred,  
    1px 1px 1px red,  
    2px 2px 1px #555,  
    3px 3px 1px #222,  
    4px 4px 1px #000;  
}
```

Frase de texto

CONQUERBLOCKS

Sombras en textos

Ejemplos

CONQUERBLOCKS

Sombras en textos

Generador online

CONQUERBLOCKS

Sombras en caja

CONQUERBLOCKS

Sombras en caja

box-shadow

- Se denominan sombras sobre cajas a las sombras en CSS que se pueden crear en una etiqueta o elemento HTML.
- Para ello, se utiliza la propiedad box-shadow, que funciona de forma muy similar a la que vimos en las sombras de texto, sólo que con algunos añadidos interesantes.

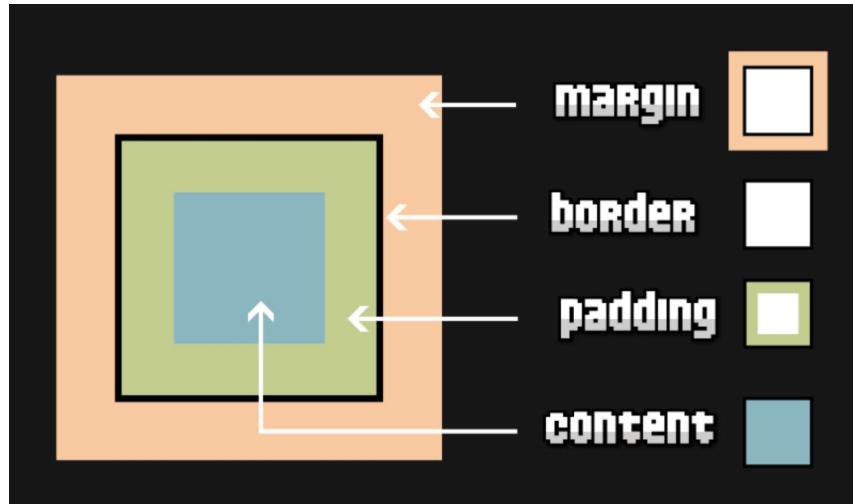
CONQUERBLOCKS

Sombras en cajas

Propiedad	Valor	Significado
box-shadow	none	Elimina (o simplemente no establece) sombra en un elemento.
box-shadow	POSX POSY	Crea sombra color negro desplazándola en horizontal y/o vertical.
box-shadow	POSX POSY SIZE	Idem a anterior, pero difuminando la sombra.
box-shadow	POSX POSY SIZE SIZE	Idem a anterior, pero aplica un factor de crecimiento a la sombra.
box-shadow	POSX POSY SIZE COLOR	Idem a la anterior, cambiando el color de la sombra.
box-shadow	POSX POSY SIZE COLOR inset	Idem a la anterior, pero aplicando sombra interna en lugar de externa.

CONQUERBLOCKS

Sombras en caja



CONQUERBLOCKS

Sombras en caja

Desplazamiento

- Para crear una sombra, los parámetros mínimos que hay que especificar son , el desplazamiento horizontal, y , el desplazamiento vertical. Podemos aplicar cantidades negativas

CONQUERBLOCKS

Sombras en caja

Desenfoque

- El tercer parámetro de las sombras en text-shadow hace referencia al desenfoque que queremos sobre dicha sombra. Funciona dandole un tamaño , que por defecto es 0, manteniendo la sombra de una forma idéntica al texto original.
- Ten en cuenta que las sombras con box-shadow siempre serán rectangulares

CONQUERBLOCKS

Sombras en caja

Factor de crecimiento

- Un cuarto parámetro opcional permite indicar un factor de crecimiento para la sombra.
- Este parámetro es opcional y se indica sólo cuando es necesario. En caso contrario, este parámetro valdrá 0 y será omitido

CONQUERBLOCKS

Sombras en caja

Factor de crecimiento

- Podemos indicar una unidad que hará crecer la sombra en todos sus lados el tamaño indicado, de forma que crezca un poco más de lo que ocupa.
- Algo que puede ser realmente útil, por ejemplo, en el caso de que el desplazamiento sea 0

CONQUERBLOCKS

Sombras en caja

```
.element {  
    box-shadow: 0 0 10px 5px;  
}
```

CONQUERBLOCKS

Sombras en caja

Color de la sombra

- Por defecto, su valor es currentColor, que generalmente suele ser black.
- Podemos utilizar cualquier color

CONQUERBLOCKS

Sombras en caja

Sombra interior

- Nuestras sombras creadas con box-shadow permiten indicar la palabra clave inset, lo que hará que la sombra en lugar de colocarse por fuera de nuestro elemento y ser una sombra exterior (por defecto), pasará a ser una sombra interior y colocarse **por dentro del elemento**

CONQUERBLOCKS

Sombras en caja

```
.element {  
    /* Sombra exterior que se desplaza hacia la zona inferior-derecha */  
    box-shadow: 10px 10px 5px black;  
  
    /* Sombra interior que se desplaza hacia la zona inferior-derecha */  
    box-shadow: -10px -10px 5px black inset;  
}
```

CONQUERBLOCKS

Sombras en caja

Sombras de caja múltiples

- Al igual que al utilizar imágenes de fondo, es posible utilizar múltiples sombras de caja. Esto nos dará suficiente flexibilidad para crear sombras con efectos.
- La sintaxis es muy sencilla

CONQUERBLOCKS

Sombras en caja

```
.element {  
  box-shadow:  
    5px 5px 10px black,  
    10px 10px 10px red,  
    20px 20px 10px blue,  
    10px 10px 10px rgb(0,0,0, 0.5) inset;  
}
```

CONQUERBLOCKS

Sombras en cajas

[Generador online](#)

CONQUERBLOCKS

Sombras idénticas

CONQUERBLOCKS

Sombras idénticas

¿Has probado hacer un box-shadow de una imagen en png?

CONQUERBLOCKS

Sombras idénticas

Probemos

CONQUERBLOCKS

Sombras idénticas

Para crear sombras idénticas se debe utilizar la función drop-shadow() que es una función que puede utilizarse en la propiedad filter (no es una propiedad independiente) y que tiene la misma sintaxis exacta de text-shadow

CONQUERBLOCKS

Sombras idénticas



CONQUERBLOCKS

Sombras idénticas

```
.box {  
    box-shadow: 5px 5px 15px black;  
}  
  
.drop {  
    filter: drop-shadow(5px 5px 15px black);  
}
```

CONQUERBLOCKS

Sombras idénticas

Demo

CONQUERBLOCKS

Filtros en CSS

CONQUERBLOCKS

Filtros

Los filtros CSS son una característica muy atractiva de CSS que permite aplicar ciertos efectos de imagen, propios de aplicaciones de retoque fotográfico, como sepia, variaciones de brillo o contraste (u otros) al vuelo en el propio navegador, sin hacer cambios permanentes sobre una imagen

CONQUERBLOCKS

Filtros

Dichos filtros funcionan a través de la propiedad filter, a la cuál hay que especificarle una función concreta de las existentes, como por ejemplo la función de blanco y negro (grayscale)

CONQUERBLOCKS

Filtros

Dichos filtros funcionan a través de la propiedad filter, a la cuál hay que especificarle una función concreta de las existentes, como por ejemplo la función de blanco y negro (grayscale)

CONQUERBLOCKS

Filtros

Función	Significado	Valor	Mínimo	Máximo	>100%
grayscale	Escala de blanco y negro	PERCENT NUMBER	0% (sin cambios)	100% = Grayscale	= 100%
blur	Desenfoque Gausiano	SIZE	0px (sin cambios)		
sepia	Grado de color sepia	PERCENT NUMBER	0% (sin cambios)	100% = sepia	= 100%
saturate	Grado de saturación	PERCENT NUMBER	0% = desaturado	100% (sin cambios)	Sí
opacity	Grado de transparencia	PERCENT NUMBER	0% = invisible	100% (sin cambios)	= 100%
brightness	Brillo	PERCENT NUMBER	0% = negro	100% (sin cambios)	Sí
contrast	Contraste	PERCENT NUMBER	0% = gris	100% (sin cambios)	Sí
hue-rotate	Rotación de color (matiz)	ANGLE	0deg (sin cambios)		
invert	Invertir	PERCENT NUMBER	0% (sin cambios)	100% = invertido	= 100%
drop-shadow	Sombra idéntica	POSX POSY SIZE COLOR	(Ver apartado función drop-shadow())		

CONQUERBLOCKS

Filtros

Demo

CONQUERBLOCKS

Modos de fusión CSS

CONQUERBLOCKS

Modos de fusión

Los filtros CSS están muy bien en multitud de situaciones, pero hay casos en los que se nos pueden quedar cortos. Los modos de fusión son un interesante añadido que aparece en CSS para dotar de otra característica muy común en los programas de diseño y edición de imagen: la fusión de dos elementos sumando la información de color de los píxeles de cada uno de ellos

CONQUERBLOCKS

Modos de fusión

Mediante las propiedades mix-blend-mode y background-blend-mode podemos utilizar modos de fusión en los diferentes elementos que seleccionemos en una página web, aplicando composición para mezclar los colores, superponiendo dos elementos, consiguiendo una variación como resultado final.

CONQUERBLOCKS

Modos de fusión

Propiedad	Valor	Significado
mix-blend-mode	normal BLEND	Aplica un modo de fusión específico a un elemento.
background-blend-mode	normal BLEND	Aplica un modo de fusión específico a un fondo.
isolation	auto isolate	Establece si un elemento debe aislarse del resto.

CONQUERBLOCKS

Modos de fusión

multiply (multiplicar)	soft-light (luz suave)
screen (trama)	difference (diferencia)
overlay (superponer)	exclusion (exclusión)
darken (oscurecer)	hue (tono)
lighten (acllarar)	saturation (saturación)
color-dodge (sobreexpone color)	color
color-burn (subexponer color)	luminosity (luminosidad)
hard-light (luz fuerte)	

CONQUERBLOCKS

Filtros

Demo

CONQUERBLOCKS

Tarea

CONQUERBLOCKS

Text-1

IPSUM

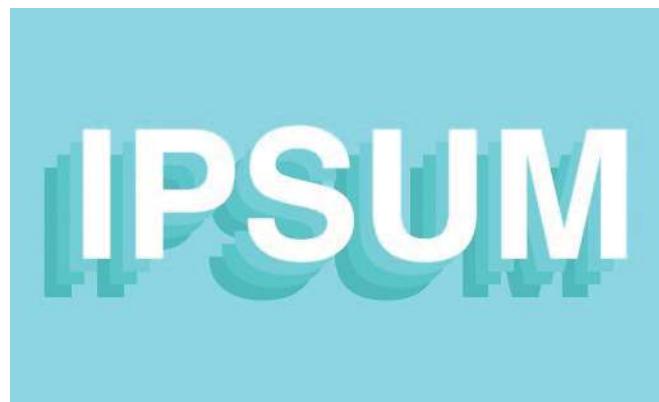
CONQUERBLOCKS

Text-2

IPSUM

CONQUERBLOCKS

Text-3



CONQUERBLOCKS

Text-4



CONQUERBLOCKS

Text-5

IPSUM

CONQUERBLOCKS

Box-1

Basic Drop Shadow

CONQUERBLOCKS

Box-2

Inner Shadow

CONQUERBLOCKS

Box-3

Offset Drop Shadow (Bottom Right)

CONQUERBLOCKS

Box-4

Offset Drop Shadow (Top Left)

CONQUERBLOCKS

Box-5

Horizontal Offset Value

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

CSS Clase 30
Resolución de menú desplegable
y sobras en texto e imágenes



Menú desplegable

Home

Our services

Contact Us

Home

Our services

Contact Us

Marketing

Social Media

Web design

CONQUERBLOCKS

Sombras en cajas y texto

CONQUERBLOCKS

Text-1

IPSUM

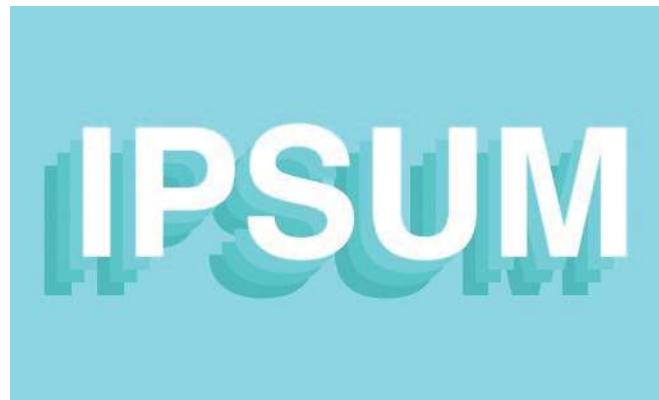
CONQUERBLOCKS

Text-2

IPSUM

CONQUERBLOCKS

Text-3



CONQUERBLOCKS

Text-4



CONQUERBLOCKS

Text-5

IPSUM

CONQUERBLOCKS

Box-1

Basic Drop Shadow

CONQUERBLOCKS

Box-2

Inner Shadow

CONQUERBLOCKS

Box-3

Offset Drop Shadow (Bottom Right)

CONQUERBLOCKS

Box-4

Offset Drop Shadow (Top Left)

CONQUERBLOCKS

Box-5

Horizontal Offset Value

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}

Clase 31



CONQUERBLOCKS

<Índice>

Clase 31

Responsive Web Design

¿Qué es?

Viewport

Instrucción @media

Breackpoints típicos

Mobile First vs Desktop First

Min/Max Height/Width Min-content/Max-content

CONQUERBLOCKS

¿Qué es RWD?

CONQUERBLOCKS

RWD

Ya no es una tendencia

Más del **60% del tráfico en internet** se hace
desde un dispositivo móvil

Es **obligatorio** aplicarlo nuestros proyectos

CONQUERBLOCKS

RWD

El diseño web adaptable, o responsive web design, es una metodología de diseño y desarrollo cuyo objetivo es crear sitios web que proporcionen una experiencia de visualización óptima en un amplio rango de dispositivos, desde computadoras de escritorio hasta teléfonos móviles y tabletas.

CONQUERBLOCKS

RWD

Esto implica que el **diseño** debe ser capaz de ajustarse fluidamente a diferentes tamaños y resoluciones de pantalla, y a diferentes entornos, garantizando legibilidad y usabilidad sin importar el dispositivo.

CONQUERBLOCKS

RWD

¿Cómo funcionaban antes del RWD los sitios web?

CONQUERBLOCKS

RWD

¿Cómo funcionaban antes del RWD los sitios web?

CONQUERBLOCKS

RWD

¿Por qué tu web debe ser responsive?

CONQUERBLOCKS

RWD

Beneficios

Mejor experiencia de usuario

Se acabaron los contenidos duplicado => SEO

Reducción de costes de desarrollo

Aumenta el alcance móvil de tus contenidos

Soporte para futuros dispositivos

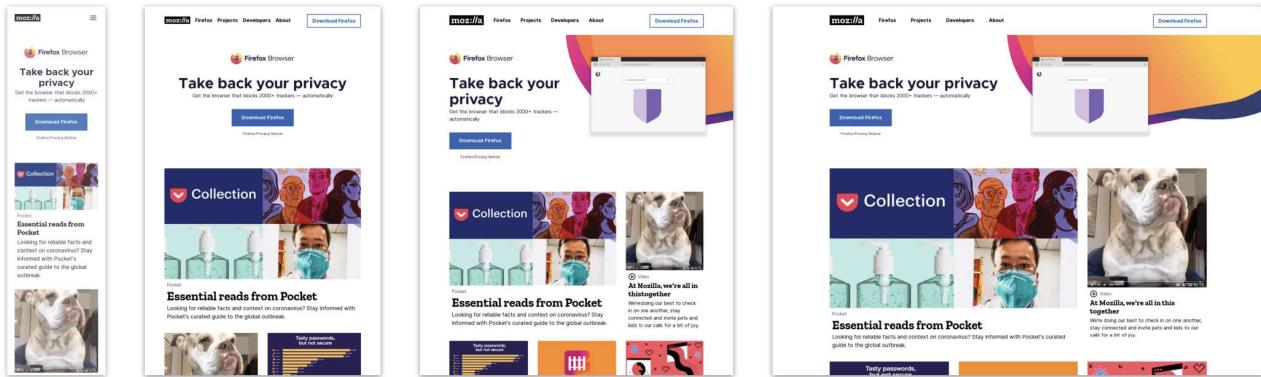
Mantenimiento mucho más sencillo

CONQUERBLOCKS

RWD

Mozilla

9

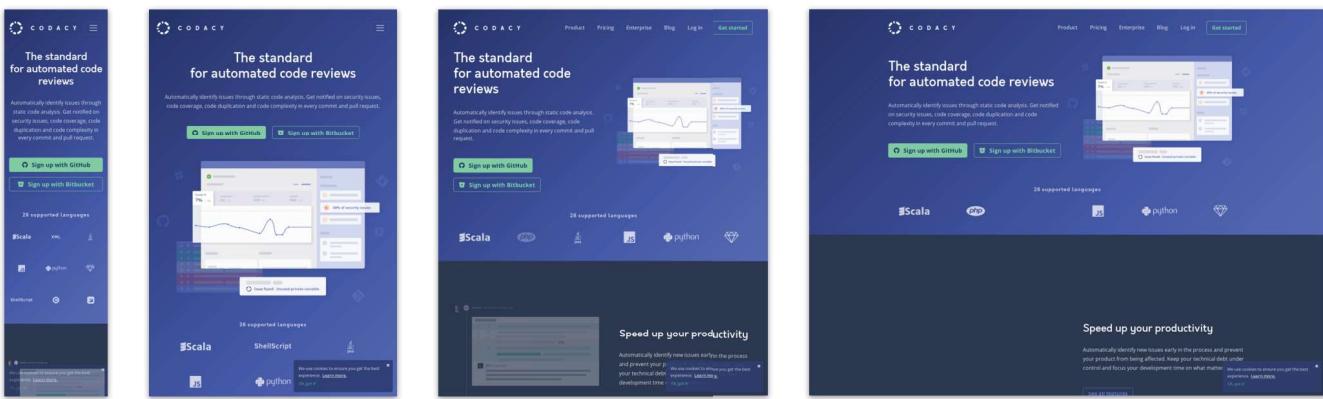


CONQUERBLOCKS

RWD

Codacy

3

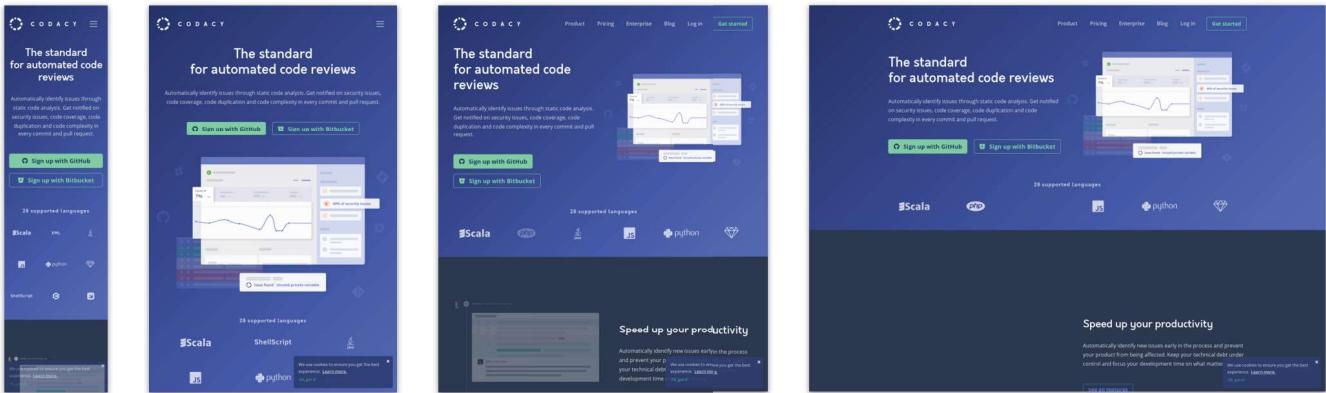


CONQUERBLOCKS

RWD

Codacy

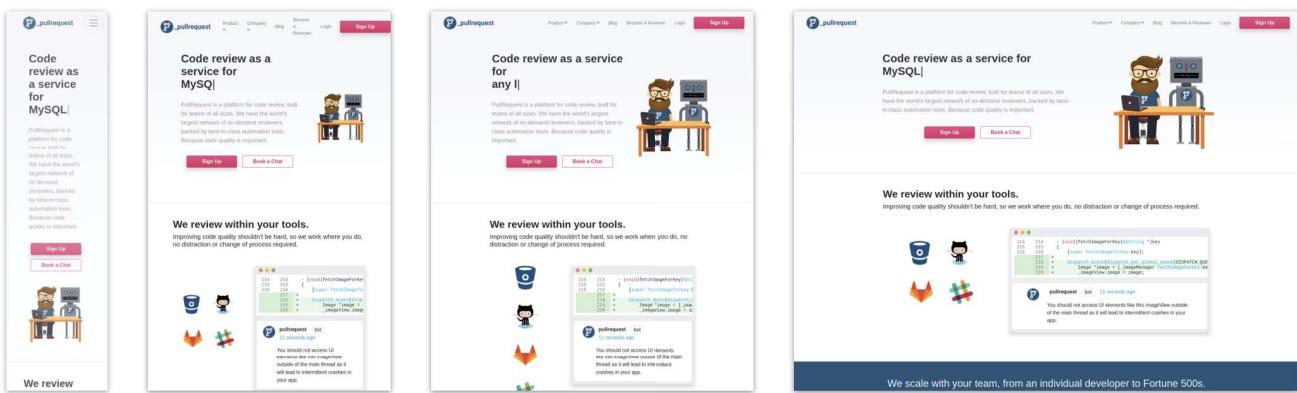
3



CONQUERBLOCKS

5

PullRequest



CONQUERBLOCKS

Principios básicos

CONQUERBLOCKS

Principios básicos

El primero de ellos es la diferencia entre diseño responsivo y diseño adaptativo. Como se puede ver en la imagen a continuación, un diseño responsive responde (valga la rebuznancia) en todo momento a las dimensiones del dispositivo, mientras que un diseño adaptable es aquel que se adapta, pero no necesariamente responde en todo momento

CONQUERBLOCKS

Principios básicos

Responsive vs Adaptive

Responsive



Adaptive



CONQUERBLOCKS

Principios básicos

No cortemos el flow normal de elementos

Flow



Static



CONQUERBLOCKS

Principios básicos

Usemos medidas relativas siempre que podamos

Relative Units



Static Units



CONQUERBLOCKS

Principios básicos

Usaremos breackpoints

With Breakpoints



Without Breakpoints



CONQUERBLOCKS

Principios básicos

Controlemos anchos máximos

Max width



No max width



CONQUERBLOCKS

Principios básicos

Usemos una metodología apropiada

Desktop first



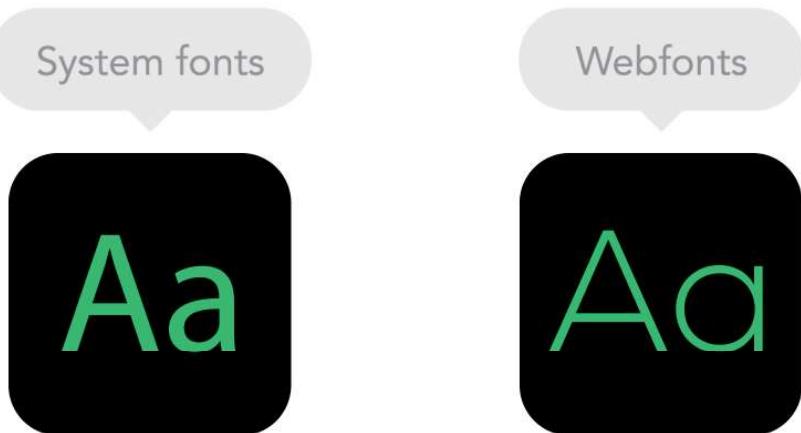
Mobile first



CONQUERBLOCKS

Principios básicos

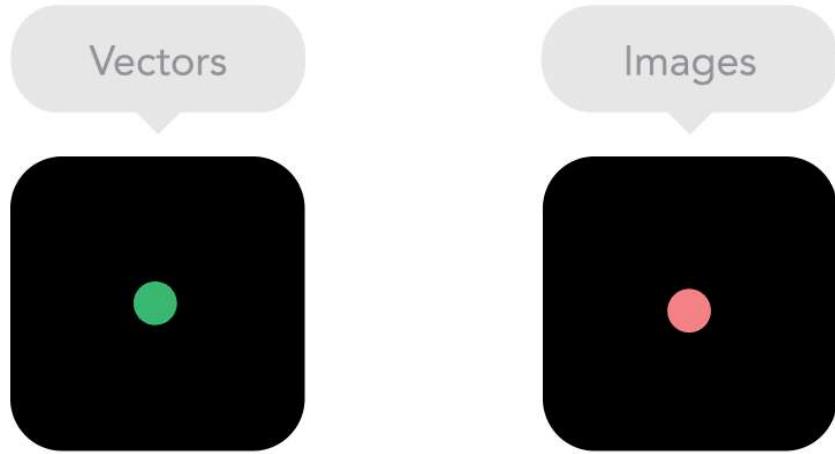
Usemos webfonts en lugar de system fonts



CONQUERBLOCKS

Principios básicos

Bitmap images vs Vectors



CONQUERBLOCKS

Principios básicos

Artículo completo

CONQUERBLOCKS

Preparación previa

CONQUERBLOCKS

Preparación previa

- ¿Cuál es tu público objetivo?
- ¿Móvil o escritorio o ambos?
- Conoce las resoluciones más utilizadas de tu público potencial, para eso tenemos Google Analytics
- Elige una estrategia acorde a los datos anteriores

CONQUERBLOCKS

Preparación previa

Comparador de resoluciones

CONQUERBLOCKS

Preparación previa

Cuidado con Google Analytics

Sólo mira analíticas de sitios que estén preparados para todas las resoluciones

CONQUERBLOCKS

Preparación previa

También es aconsejable echar un vistazo a información externa como las que nos proporcionan estadísticas globales anónimas de Global StatCounter, para hacernos una idea de los atributos más comunes.

CONQUERBLOCKS

Preparación previa

Una vez tengamos decidida una estrategia, el departamento de diseño la aplicará desde el inicio del proyecto

CONQUERBLOCKS

Viewport

CONQUERBLOCKS

Viewport

En muchos casos puede que oigas hablar del viewport del navegador. Esa palabra hace referencia a la región visible del navegador, o sea, la parte de la página que está visualizándose actualmente en el navegador

CONQUERBLOCKS

Viewport

Los usuarios podemos redimensionar la ventana del navegador para reducir el tamaño del viewport y simular que se trata de una pantalla y dispositivo más pequeño

CONQUERBLOCKS

Viewport

Si queremos editar ciertos comportamientos del viewport del navegador, podemos editar el documento HTML para especificar el siguiente campo meta, antes de la parte del </head>:

```
<meta name="viewport" content="initial-scale=1, width=device-width">
```

CONQUERBLOCKS

Viewport

Parámetro	Valor	Significado
width	device-width	Indica un ancho para el viewport.
height	device-height	Indica un alto para el viewport.
initial-scale	1	Escala inicial con la que se visualiza la página web.
minimum-scale	0.1	Escala mínima a la que se puede reducir al hacer zoom.
maximum-scale	10	Escala máxima a la que se puede aumentar al hacer zoom.
user-scalable	no o fixed yes o zoom	Posibilidad de hacer zoom en la página web.

CONQUERBLOCKS

Viewport

```
<meta name="viewport" content="initial-scale=1, width=device-width">
```

CONQUERBLOCKS

Viewport

En dispositivos móviles, el viewport cambia de tamaño y se mueve cuando el usuario hace zoom o cambia la orientación de su dispositivo

CONQUERBLOCKS

Viewport

Con esto conseguiremos preparar nuestra web para dispositivos móviles y prepararnos para la introducción de reglas media query en el documento CSS.

Es importante no olvidar este paso.

CONQUERBLOCKS

Viewport

Conclusión

La gestión del viewport es crucial para el diseño responsivo, ya que permite a los diseñadores y desarrolladores controlar cómo se adapta y se muestra el contenido en diferentes tamaños de pantalla, asegurando una experiencia de usuario coherente y accesible.

CONQUERBLOCKS

Instrucción @media

CONQUERBLOCKS

@media

Una vez nos adentramos en el mundo del Responsive Design, nos damos cuenta en que hay situaciones en las que determinados aspectos o componentes visuales deben aparecer con ciertas diferencias dependiendo del dispositivo donde se están visualizando, ya que no todos los dispositivos tienen los mismos tamaños o características

CONQUERBLOCKS

@media

Por ejemplo, una zona donde se encuentra el buscador de la página puede estar colocada en un sitio concreto en la versión de escritorio, pero en móvil quizás nos interese que ocupe otra zona (o que tenga otro tamaño o forma) para aprovechar mejor el espacio de la versión del dispositivo móvil.

CONQUERBLOCKS

@media

¿Qué son las media queries?

CONQUERBLOCKS

@media

Existe un concepto denominado media queries, mediante el cuál podemos hacer excepciones para que unos determinados estilos de diseño sólo se apliquen si se cumplen una serie de condiciones, generalmente relacionadas con el dispositivo mediante el cuál se está viendo la página.

CONQUERBLOCKS

@media

Regla	Descripción
@media (<condición>)	Si se cumple la condición, se aplican los estilos de su interior.
@media not (<condición>)	Si no se cumple la condición, se aplican los estilos de su interior.

Básicamente, se trata de establecer una condición que se aplicará a la página actual, y en el caso de cumplirse, se aplicarán los estilos indicados en su interior. De no cumplirse, no se aplicarán. Si utilizamos la keyword not antes de la condición, se invierte la condición

CONQUERBLOCKS

@media

```
@media (*condición*) {  
    .container {  
        background: green;  
    }  
}  
  
@media not (*condición*) {  
    .container {  
        background: red;  
    }  
}
```

En el ejemplo anterior, si se cumple la condición establecida, se aplicará un color verde. Sin embargo, si no se cumple, se aplicará un color rojo. Observa que es similar al funcionamiento de un if / else en programación

CONQUERBLOCKS

@media

```
@media (*condición*) {  
    .container {  
        background: green;  
    }  
}  
  
@media not (*condición*) {  
    .container {  
        background: red;  
    }  
}
```

En el ejemplo anterior, si se cumple la condición establecida, se aplicará un color verde. Sin embargo, si no se cumple, se aplicará un color rojo. Observa que es similar al funcionamiento de un if / else en programación

CONQUERBLOCKS

Condiciones múltiples

```
@media (*condición*) and (*condición*) {  
    .container {  
        background: orangered;  
    }  
}
```

En el ejemplo anterior, si se cumple la condición establecida, se aplicará un color verde. Sin embargo, si no se cumple, se aplicará un color rojo. Observa que es similar al funcionamiento de un if / else en programación

CONQUERBLOCKS

Tipos de “medios”

Tipo de medio	Significado
all	Todos los dispositivos o medios. El que se utiliza por defecto .
screen	Monitores o pantallas de ordenador. Es el más común.
print	Documentos de <u>medios impresos</u> o pantallas de previsualización de impresión.
speech	Lectores de texto para invidentes (Antes <u>aural</u> , el cuál ya está obsoleto).

CONQUERBLOCKS

@media

Tipo de medio y condición

```
@media screen and (*condición*) {  
    /* ... */  
}  
  
@media print {  
    /* ... */  
}
```

CONQUERBLOCKS

@media

¿Qué podemos meter como condición?

En los ejemplos anteriores hemos indicado *condiciones* en el interior de los paréntesis, pero no hemos visto como definir ninguna. Vamos a echar un vistazo a que tipo de características podemos definir en las condiciones de los media queries.

CONQUERBLOCKS

¿Qué podemos meter como condición?

Tipo de característica	Valores	¿Cuándo se aplica?
width	SIZE	Si el dispositivo tiene el tamaño de ancho exactamente .
min-width	SIZE	Si el dispositivo tiene un tamaño de ancho mayor al indicado.
max-width	SIZE	Si el dispositivo tiene un tamaño de ancho menor al indicado.
height	SIZE	Si el dispositivo tiene el tamaño de alto exactamente .
min-height	SIZE	Si el dispositivo tiene un tamaño de alto mayor al indicado.
max-height	SIZE	Si el dispositivo tiene un tamaño de alto menor al indicado.
aspect-ratio	aspect-ratio	Si el dispositivo encaja con la proporción de aspecto indicada.
orientation	landscape portrait	Si el dispositivo está en colocado en modo vertical o apaisado .

¿Qué podemos meter como condición?

```
.container {
  background: grey;
  height: 100px;
}

@media (max-width: 800px) and (orientation: landscape) {
  .container {
    background: red;
  }
}
```

@media

¿Lo probamos?

CONQUERBLOCKS

@media

- Azul para resoluciones menores a 400 píxeles de ancho (móviles).
- Rojo para resoluciones entre 400 píxeles y 800 píxeles de ancho (tablets).
- Verde para resoluciones mayores a 800 píxeles (desktop).

CONQUERBLOCKS

@media

Media query range

Aunque la forma de definir condiciones anterior es útil, a menudo resulta confusa y difícil de escribir.

Es posible utilizar una modalidad denominada rangos de condiciones, que son algo más versátiles que las anteriores y mucho menos tediosas.

CONQUERBLOCKS

@media

Media query range

La diferencia radica en que se puede utilizar operadores matemáticos que harán que las condiciones sean mucho más legibles y fáciles de interpretar. Veamos el mismo ejemplo anterior con sintaxis de rangos:

CONQUERBLOCKS

@media

```
@media (width ≤ 400px) {  
  .menu {  
    background: blue;  
  }  
}  
  
@media (400px ≤ width ≤ 800px) {  
  .menu {  
    background: red;  
  }  
}  
  
@media (width ≥ 800px) {  
  .menu {  
    background: green;  
  }  
}
```

Media query range

CONQUERBLOCKS

@media

Media desde el html

Por último, hay que tener en cuenta que los media queries también es posible indicarlos desde HTML, utilizando la etiqueta `<link>` y el atributo `media` para establecer la condición

CONQUERBLOCKS

@media

Media desde el html

```
<link rel="stylesheet"  
      href="mobile.css"  
      media="(max-width: 640px)">  
  
<link rel="stylesheet"  
      href="tablet.css"  
      media="(min-width: 640px) and (max-width: 1280px)">  
  
<link rel="stylesheet"  
      href="desktop.css"  
      media="(min-width: 1280px)">
```

CONQUERBLOCKS

Breakpoints

CONQUERBLOCKS

Breakpoints

Los breakpoints en el diseño web responsive son las **medidas específicas** donde el contenido y el diseño de un sitio web cambiarán para adaptarse a diferentes tamaños de pantalla

CONQUERBLOCKS

Breakpoints

Estos breakpoints están comúnmente basados en las anchuras de los dispositivos y se definen utilizando la regla @media en CSS.

CONQUERBLOCKS

Breakpoints

La teoría

CONQUERBLOCKS

Breakpoints

Teléfonos Móviles (pequeños)

320px: Ancho de pantalla para teléfonos móviles más antiguos.

360px – 400px: Ancho de pantalla para teléfonos móviles modernos.

CONQUERBLOCKS

Breakpoints

Teléfonos Móviles (grandes)

414px – 480px: Pantallas más grandes de dispositivos móviles y phablets.

CONQUERBLOCKS

Breakpoints

Tabletas (verticales/portrait)

600px – 768px: Ancho de pantalla para tabletas pequeñas y dispositivos intermedios.

CONQUERBLOCKS

Breakpoints

Tabletas (horizontales/landscape) / Pantallas pequeñas de laptops

768px - 1024px: Ancho de pantalla para tabletas en orientación horizontal y pantallas pequeñas de laptops.

CONQUERBLOCKS

Breakpoints

Laptops / Desktops

1024px - 1280px: Ancho de pantalla para laptops y monitores pequeños.

1280px - 1366px: Un breakpoint común para laptops de tamaño estándar.

CONQUERBLOCKS

Breakpoints

Desktops (grandes)

1440px - 1600px: Pantallas de desktops de alta resolución.

CONQUERBLOCKS

Breakpoints

Desktops (extra grandes) / Monitores de TV:

1920px y más: Para pantallas de alta resolución, incluyendo monitores 4K.

CONQUERBLOCKS

Breakpoints

Optimiza tus esfuerzos

CONQUERBLOCKS

Breakpoints

Es importante tener en cuenta que estos valores pueden variar según el público objetivo y los dispositivos más comunes entre los usuarios del sitio web. Además, con la variedad de dispositivos en constante cambio, es recomendable usar un enfoque de diseño web "first mobile" o "mobile-first", donde se comienza con el diseño para móviles y luego se escalan los estilos para pantallas más grandes.

CONQUERBLOCKS

Mobile First vs Desktop First

CONQUERBLOCKS

Mobile First vs Desktop First

¿Por qué es mejor usar Mobile First en lugar
de Desktop First?

CONQUERBLOCKS

Mobile First vs Desktop First

Les obliga a los diseñadores a pensar

CONQUERBLOCKS

Mobile First vs Desktop First

Prioridad al usuario móvil

CONQUERBLOCKS

Mobile First vs Desktop First

Esencial y Útil vs Visual

CONQUERBLOCKS

Mobile First vs Desktop First

Mejora la velocidad de carga

=

mejor posicionamiento SEO

CONQUERBLOCKS

Mobile First vs Desktop First

100% la UX estará mejor pensada y adaptada de pequeño a grande que si lo hacemos al contrario

CONQUERBLOCKS

Mobile First vs Desktop First

Desarrollo progresivo, siempre mejor, tanto para pensar como para maquetar/programar

CONQUERBLOCKS

Mobile First vs Desktop First

Compatibilidad de serie con el “Tap”

CONQUERBLOCKS

Mobile First vs Desktop First

Google y el
Progressive Web Apps (PWA)
mandan...
Hagámosle caso

CONQUERBLOCKS

**min-max-width-height
min-max-content**

CONQUERBLOCKS

min - max

W3C

CONQUERBLOCKS

«Despedida»

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS

{CSS}



Clase 32

<Índice>

Clase 32

Variables en CSS

¿Qué son?

Declaración y uso

Scope/ámbito de variables

Ventajas

Caso práctico

¿Qué son?

¿Qué son?

Custom Properties

Las variables CSS, también conocidas como propiedades personalizadas, son entidades definidas por los desarrolladores que contienen valores específicos que se pueden reutilizar en todo el documento CSS

¿Qué son?

Hacen que la hoja de estilos sea más mantenible y los valores fácilmente cambiados en un solo lugar. Estas variables pueden almacenar colores, fuentes, tamaños o cualquier otro valor de CSS que deseas utilizar en varios lugares de tu hoja de estilos

El objetivo principal suele ser evitar escribir múltiples veces ese valor, y en su lugar, ponerle un nombre más lógico y fácil de recordar, que hará referencia al valor real.

Ejemplos de uso:
Tamaños, tipografías, márgenes, paddings etc...

Definición y uso

Definición y uso

Para definir una custom property haremos uso de los dos guiones -- previos al nombre que queramos utilizar. Además, debemos fijarnos en el elemento que definimos la variable, en este ejemplo la pseudoclase :root:

Definición y uso

```
:root {  
    --background-color: black;  
}
```

Definición y uso

La pseudoclase :root hace referencia al elemento raíz del documento, o lo que es lo mismo, al elemento <html>. La diferencia de utilizar html o :root como selector es que este último tiene algo más de especificidad CSS. Mientras que html tiene 001, :root tendría 010.

Definición y uso

Al colocarla en :root estamos definiendo que la custom property estará definida para el ámbito de esa etiqueta <html> (o cualquier elemento hijo), es decir, a todo el documento. Sin embargo, ya veremos que podemos aplicar estas variables sólo a partes concretas del DOM de nuestra página.

Definición y uso

Uso de variables

Definición y uso

```
:root {  
  --background-color: indigo;  
}  
  
body {  
  background: var(--background-color);  
}
```

Definición y uso

En este caso estamos aplicando a la propiedad background el valor que contiene --background-color para el elemento <body>. Esto último es muy importante entenderlo, ya que una custom property puede tener diferentes valores dependiendo del ámbito en el que se encuentra.

Definición y uso

Fallback

Además, es muy recomendable que la expresión var() tenga dos parámetros. El primero de ellos, la custom property en cuestión, el segundo de ellos, el valor por defecto en el caso de que esa propiedad no esté definida en el ámbito actual:

Definición y uso

```
body {  
    background: var(--background-color, blue);  
}
```

Definición y uso

En este caso, la variable `--background-color` no está definida, por lo que el `<body>` tendrá siempre el color de fondo azul, ya que es el fallback indicado.

Definición y uso

Demo fallback

Scope / ámbito de variables

Scope / ámbito de variables

El ejemplo anterior es muy básico y puede que no se aprecie el detalle de los ámbitos con las custom properties, así que vamos a verlo con un ejemplo más específico.

Scope / ámbito de variables

```
.parent {  
    --background-color: black;  
  
    color: white;  
}  
  
.first {  
    --background-color: purple;  
}  
  
.child {  
    background: var(--background-color, blue);  
}
```

```
<div class="parent">  
    <div class="first child">First child</div>  
    <div class="second child">Second child</div>  
</div>  
<div class="third child">Third child</div>
```

Scope / ámbito de variables

- Los dos primeros elementos .child tomarán color negro, ya que se le aplica a .parent (e hijos).
- El primer elemento .child se sobreescribe con color púrpura, ya que se le aplica a .first.
- El tercer elemento no tendrá ninguna variable definida, por lo que tomará color azul.

Scope / ámbito de variables

Esto nos permite mucha flexibilidad y potencia a la hora de utilizar custom properties en CSS, ya que son tremadamente útiles y versátiles, permitiéndonos utilizar incluso la cascada de CSS a nuestro favor.

Scope / ámbito de variables

```
:root { --color: blue; }
div { --color: green; }
#alert { --color: red; }
* { color: var(--color); }

<p>I inherited blue from the root element!</p>
<div>I got green set directly on me!</div>
<div id='alert'>
  While I got red set directly on me!
  <p>I'm red too, because of inheritance!</p>
</div>
```

Scope / ámbito de variables

DEMO

Ventajas y usos

Ventajas y usos

Usos típicos

Ventajas y usos

Tamaños de fuente

Puedes establecer variables para diferentes tamaños de fuente que quieras utilizar en todo tu sitio. Por ejemplo, puedes tener variables para tamaños de fuente para encabezados, texto del cuerpo, subtítulos, etc.

```
:root {  
  --font-size-body: 16px;  
  --font-size-heading: 32px;  
  --font-size-subheading: 24px;  
}
```

Ventajas y usos

Familias de Fuentes

Define variables para las familias de fuentes para mantener la consistencia y facilitar los cambios de las fuentes a través del proyecto.

```
:root {  
  --font-family-primary: 'Open Sans', sans-serif;  
  --font-family-secondary: 'Roboto', sans-serif;  
}
```

Ventajas y usos

Peso de Fuentes

Las variables pueden contener los diferentes pesos de fuente que tu diseño requiere, permitiendo un cambio sencillo y coherente en los pesos de fuente en todo el sitio.

```
:root {  
    --font-weight-normal: 400;  
    --font-weight-bold: 700;  
}
```

Ventajas y usos

Altura de línea

Para mantener una legibilidad y espaciado consistentes, puedes usar variables para la altura de línea de los distintos elementos de texto.

```
:root {  
    --line-height-body: 1.5;  
    --line-height-heading: 1.2;  
}
```

Ventajas y usos

Color de texto

Las variables son muy útiles para definir los colores de texto que se usarán en todo el sitio web, facilitando la implementación de un esquema de color consistente.

```
:root {  
    --text-color-primary: #333;  
    --text-color-secondary: #555;  
}
```

Ventajas y usos

```
body {  
    font-family: var(--font-family-primary);  
    font-size: var(--font-size-body);  
    line-height: var(--line-height-body);  
    color: var(--text-color-primary);  
}  
  
h1 {  
    font-size: var(--font-size-heading);  
    line-height: var(--line-height-heading);  
    font-weight: var(--font-weight-bold);  
}
```

Ventajas y usos

Conclusión

Usar variables para la tipografía no solo mejora la coherencia en el diseño de tu sitio web, sino que también hace que sea mucho más fácil hacer ajustes globales a tu tipografía en el futuro.

Ventajas y usos

Ventajas

Ventajas y usos

Tematización

Puedes definir un conjunto de variables para colores, fuentes y otros valores estéticos que hacen un tema. Esto facilita el cambio de tema de un sitio web solo con cambiar los valores de estas variables.

Ventajas y usos

Consistencia de Diseño

Utilizar variables para espaciados (márgenes y paddings), tamaños de fuente, y medidas de elementos garantiza que tu diseño mantenga consistencia en todo el sitio web.

Ventajas y usos

Reutilización de código

Al usar variables, reduces la repetición de valores específicos en tu CSS, lo que hace que tu código sea más limpio y fácil de mantener.

Ventajas y usos

Facilidad de mantenimiento

Cuando necesitas actualizar un valor, como un color principal o un tamaño de fuente, puedes hacerlo en un solo lugar, y el cambio se aplicará en todo tu CSS donde se use esa variable.

Ventajas y usos

Adaptabilidad a Cambios de Contexto

Puedes cambiar el valor de las variables en diferentes contextos usando media queries, lo cual es útil para diseños responsives

Ventajas y usos

Interactividad con JS

Las variables CSS pueden ser leídas y modificadas con JavaScript, lo que te permite cambiar estilos dinámicamente en respuesta a la interacción del usuario.

Ventajas y usos

Mejora la legibilidad

Al nombrar las variables de forma descriptiva, haces que tu código sea más fácil de entender para otros desarrolladores (o para ti mismo en el futuro).

Ventajas y usos

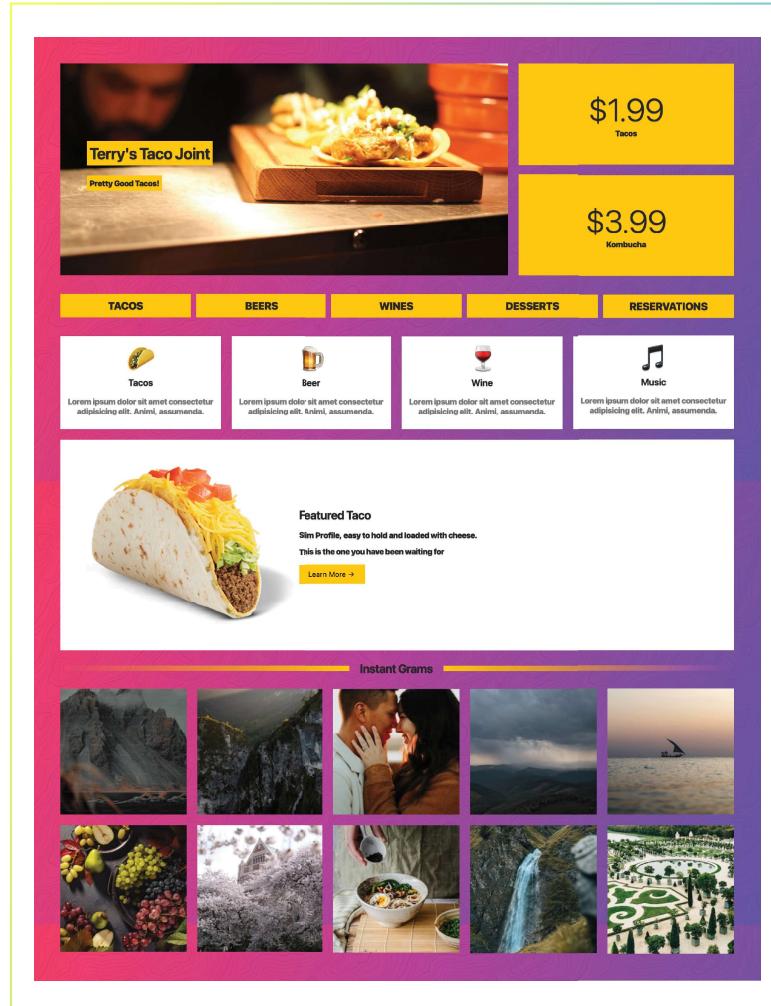
Mejora la legibilidad

Al nombrar las variables de forma descriptiva, haces que tu código sea más fácil de entender para otros desarrolladores (o para ti mismo en el futuro).

Ventajas y usos

Las variables CSS hacen que trabajar con hojas de estilo sea más parecido a programar, con beneficios similares en cuanto a la abstracción y reutilización de código. Su uso se ha vuelto una práctica estándar en el desarrollo front-end moderno.

Caso práctico



<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

{CSS}



Clase 33

<Índice>

Clase 33

Transformaciones en CSS

¿Qué son? y uso

Translaciones

Escalado

Rotaciones

Deformaciones

Transformaciones en 3D y perspectiva

¿Qué son? y uso

¿Qué son?

Las **transformaciones** son una de las características de CSS más interesantes y potentes que se introducen en el lenguaje para convertir las hojas de estilo en un sistema capaz de realizar efectos visuales 2D y 3D

¿Qué son?

La propiedad **transform**

Propiedad	Descripción	Valor
transform	Aplica una (o varias) transformaciones CSS al elemento.	función 1, función 2, función 3...

¿Qué son?

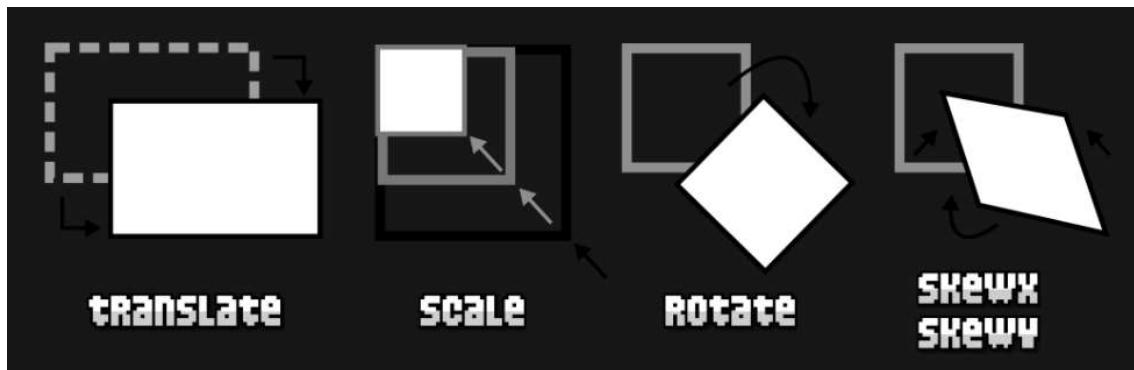
¿Os acordáis de los naipes?

```
.element {  
    width: 100px;  
    height: 100px;  
    background: indigo;  
    transform: rotate(45deg);  
}
```

```
<div class="element"></div>
```

¿Qué son?

¿Qué tipo de funciones podemos aplicar?



¿Qué son?

¿Qué tipo de funciones podemos aplicar?

Tipo de transformación	Descripción
Translación 2D	Desplaza un elemento en el eje X (<u>izquierda</u> , <u>derecha</u>) y/o en el eje Y (<u>arriba</u> , <u>abajo</u>)
Escalado 2D	Escala el elemento una determinada cantidad más grande o más pequeña. También se puede voltear.
Rotación 2D	Gira el elemento sobre su eje X o sobre su eje Y. También se puede girar sobre sí mismo.
Deformación 2D	Inclina el elemento sobre su eje X o sobre su eje Y.

¿Qué son?

Transformaciones múltiples

¿Qué son?

Transformaciones múltiples

```
.element {  
    transform: rotate(5deg); /* No hace efecto */  
    transform: scale(2);    /* Sobreescribe la anterior */  
}
```

¿Qué son?

Transformaciones múltiples

```
.element {  
    transform: rotate(5deg); /* No hace efecto */  
    transform: scale(2);    /* Sobreescribe la anterior */  
}
```

¿Qué son?

Transformaciones múltiples

```
.element {  
    transform: rotate(5deg) scale(2) translate(25px, 150px);  
}
```

¿Qué son?

Orden de las transformaciones

¿Qué son?

Otro detalle que conviene tener en cuenta a la hora de aplicar múltiples transformaciones es que el orden de transformación importa. No es lo mismo realizar una rotación y luego una translación, que la misma translación primero y luego la misma rotación. Veámoslo en un ejemplo.

¿Qué son?

```
.element {  
    width: 50px;  
    height: 50px;  
    background: grey;  
    transform: translate(150px, 100px) rotate(25deg);  
}
```

¿Qué son?

Punto de origen

Propiedades	Formato	Significado
transform-origin	POSX POSY	Cambia el punto de origen del elemento en una transformación.

¿Qué son?

¿Cómo afecta al posicionamiento de elementos?

Translación

Translación

Las funciones de translación son aquellas que realizan una transformación en la que mueven un elemento de un lugar a otro. Para ello, las utilizaremos en el interior de la propiedad CSS transform y elegiremos una de las siguientes funciones de translación.

Translación

Funciones	Significado
translateX(x)	Traslada el elemento una distancia de SIZE x horizontalmente.
translateY(y)	Traslada el elemento una distancia de SIZE y verticalmente.
translate(x, y)	Propiedad de atajo de las dos anteriores.
translate(x)	Equivalente a translate(x, 0)
translateZ(z)	Ver en Transformaciones 3D
translate3d(x, y, z)	Ver en Transformaciones 3D

Translación

Respecto al ámbito 2D, tenemos 3 funciones de translación: `translateX()` y `translateY()` para cada eje, y la propiedad de atajo `translate()`, que es una mezcla de las dos anteriores en una sola función de translación.

Translación

Si especificamos un valor positivo en el eje X (horizontal), lo moveremos hacia la derecha, y si especificamos un valor negativo, lo moveremos hacia la izquierda. Ocurre exactamente lo mismo con el eje Y (vertical), que con valores negativos lo movemos hacia arriba y con valores positivos lo movemos hacia abajo

Translación

```
.element {  
    transform: translateX(20px) translateY(-30px);  
}  
  
/* La transformación anterior es equivalente a esta (atajo) */  
.element {  
    transform: translate(20px, -30px);  
}
```

Translación

Propiedad translate

En nuevas versiones de los navegadores, ya se soporta la propiedad individual `translate`, y no hace falta utilizarla dentro de la propiedad `transform`.

Propiedad	Valor	Significado
<code>translate</code>	<code>none</code>	No aplica desplazamiento. Valor por defecto.
<code>translate</code>	<code>SIZE</code>	Desplaza un elemento el tamaño especificado en el eje X.
<code>translate</code>	<code>SIZE</code> <code>SIZE</code>	Desplaza un elemento una cierta cantidad en el eje X y eje Y.
<code>translate</code>	<code>SIZE</code> <code>SIZE</code> <code>SIZE</code>	Desplaza un elemento el eje X, eje Y y eje Z.

Translación

```
.element {  
    translate: 50px;          /* Equivalente a translateX(50px) */  
    translate: 50px 150px;    /* Equivalente a translate(50px, 150px) */  
    translate: 0 150px;      /* Equivalente a translateY(150px) */  
    translate: 50px 150px 100px; /* Equivalente a translate(50px, 150px, 100px) */  
    translate: 0px 0px 30px;   /* Equivalente a translateZ(30px) */  
}
```

Translación

Demos

Escalado

Escalado

Las funciones de **escalado** son aquellas que realizan una transformación en la que aumentan o reducen el tamaño de un elemento. Para ello, las utilizaremos en el interior de la propiedad CSS transform y elegiremos una de las siguientes funciones de escalado.

Escalado

Funciones	Significado
scaleX(fx)	Reescala el elemento un número de NUMBER <code>fx</code> veces (sólo en horizontal).
scaleY(fy)	Reescala el elemento un número de NUMBER <codefy> veces (sólo en vertical).</codefy>
scale(fx, fy)	Propiedad de atajo de las dos anteriores (escalado simétrico).
scale(fx)	Equivalente al anterior: <code>scale(fx, fx)</code> .
scaleZ(fz)	Ver en Transformaciones 3D
scale3D(fx, fy, fz)	Ver en Transformaciones 3D

Escalado

Por ejemplo, la propiedad `transform: scale(2, 0.5)` realiza una transformación de escalado del elemento, ampliéndolo al doble de su tamaño original en el eje X (horizontal) y a la mitad en el eje Y (vertical).

Escalado

La propiedad **scale**

En nuevas versiones de los navegadores, ya se soporta la propiedad individual scale, y no hace falta utilizarla dentro de la propiedad transform.

Propiedad	Valor	Significado
scale	none	No aplica escalado. Valor por defecto.
scale	NUMBER	Aplica el factor de escala simétrico al eje X/Y. Igual a <code>scale: x x 1</code> .
scale	NUMBER	Aplica los factores de escala al eje X y eje Y. Igual a <code>scale: x y 1</code> .
scale	NUMBER	Aplica un factor de escala a cada eje. Igual a <code>scale: x y z</code> .

Escalado

Función espejo con CSS

Escalado

DEMO

Rotación

Rotación

Las funciones de rotación son aquellas que realizan una transformación en la que giran un elemento respecto a un eje específico. Para ello, las utilizaremos en el interior de la propiedad CSS transform y elegiremos una de las siguientes funciones de rotación.

Rotación

Rotación

Las funciones de rotación simplemente giran el elemento una cierta cantidad respecto al eje involucrado. Disponemos de las siguientes funciones de rotación

Rotación

Funciones	Significado
rotateX(x)	Establece una rotación 2D en ANGLE x sólo para el eje horizontal X.
rotateY(y)	Establece una rotación 2D en ANGLE y sólo para el eje vertical Y.
rotateZ(z)	Establece una rotación 2D en ANGLE z sobre si mismo.
rotate(z)	Alias a la anterior.
rotate3d(x, y, z, a)	Ver en Transformaciones 3D

Demo

Rotación

La propiedad **rotate**

En nuevas versiones de los navegadores, ya se soporta la propiedad individual **rotate**, y no hace falta utilizarla dentro de la propiedad **transform**.

Propiedad	Valor	Significado
rotate	none	No aplica rotación. Valor por defecto.
rotate	ANGLE	Rota el elemento sobre si mismo. Equivalente a rotateZ() .
rotate	AXIS ANGLE	Rota el elemento sobre el eje (x, y o z) indicado.
rotate	NUMBER NUMBER NUMBER ANGLE	Indica un vector de rotaciones con el ángulo indicado.

Rotación

```
.element {  
  rotate: 45deg;          /* Equivale a transform: rotateZ(45deg); */  
  
  rotate: x 45deg;        /* Equivale a transform: rotateX(45deg); */  
  rotate: y 120deg;       /* Equivale a transform: rotateY(120deg); */  
  
  rotate: 0 0 1 45deg;    /* Equivale a transform: rotateZ(45deg); */  
  rotate: 1 0 0 15deg;    /* Equivale a transform: rotateX(15deg); */  
  rotate: 0 1 1 5deg;     /* Equivale a transform: rotateY(5deg) rotateZ(5deg); */  
}
```

Rotación

DEMO

Deformaciones

Deformaciones

Por último, las funciones de deformación establecen un ángulo para torcer, tumbar o inclinar un elemento en 2D. A diferencia de las anteriores, no tiene función correspondiente para 3D.

Deformaciones

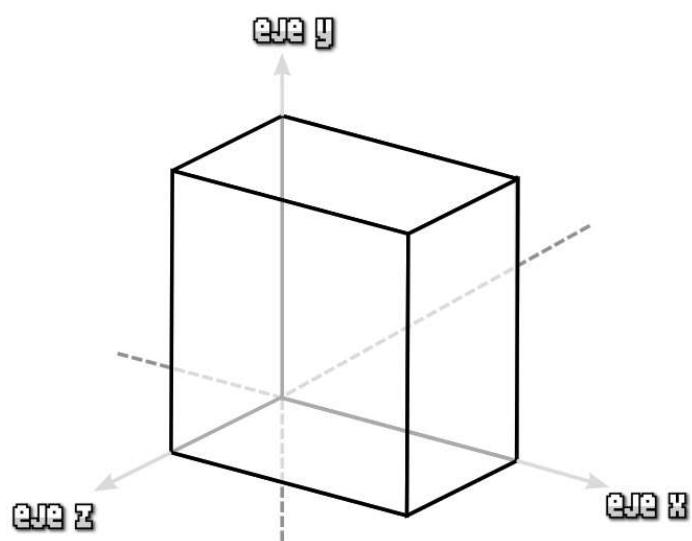
Funciones	Significado
<code>skewX(xdeg)</code>	Establece un ángulo de <code>ANGLE</code> <code>xdeg</code> para una deformación 2D respecto al eje X.
<code>skewY(ydeg)</code>	Establece un ángulo de <code>ANGLE</code> <code>ydeg</code> para una deformación 2D respecto al eje Y.

Deformaciones

Demo

Transformaciones en 3D y perspectiva

3D



3D

Demo de perspectiva 1

Demo de perspectiva 2

Demo TRANSFORMACIONES 3D

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

{CSS}

Clase 34



<Índice>

Clase 34

Animaciones en CSS

¿Qué son?

Transiciones

Animaciones y parámetros

@keyframes

Funciones de tiempo

¿Qué son?

¿Qué son?

Cambios de estilos

Por ejemplo con :hover

El cambio se hace de forma instantánea

Cambiemos esto...

¿Qué son?

Interactividad
Dinamismo
Pero sin pasarse...

Transiciones

Transiciones

Utilizando transiciones tenemos a nuestra disposición una gran flexibilidad que nos permitirá dotar de atractivos efectos de transición que harán que nuestros diseños sean más elegantes y agradables.

Transiciones

Las transiciones se basan en un principio muy básico: conseguir un efecto suavizado entre un estado inicial y un estado final al realizar una acción.

Transiciones

Las propiedades CSS que podemos utilizar relacionadas con las transiciones son las siguientes

Transiciones

Propiedades de transición

Propiedades	Descripción	Valor
transition-duration	Tiempo de duración.	0 TIME
transition-property	Propiedades CSS afectadas por la transición.	all none propiedad css
transition-timing-function	Ritmo de la transición.	Ver funciones de tiempo
transition-delay	Tiempo de retardo inicial.	0 TIME

Transiciones

Ten en cuenta que por defecto las transiciones están habilitadas para todas las propiedades CSS involucradas. Sin embargo, tienen una duración de 0s. Salvo que cambiemos esta duración, no se realizará transición.

Transiciones

transition-duration

Con la propiedad transition-duration especificaremos la duración de la transición, desde el inicio de la transición, hasta su finalización. Recuerda que por defecto, las transiciones tienen una duración de 0s, por lo que si no cambias este valor, cambiará de golpe y será lo mismo que no tener transición.

Transiciones

Ojo: Si establecemos una duración demasiado alta, el navegador realizará la transición a trozos intermitentes, lo que hará que se vea a golpes. Se recomienda siempre comenzar con valores cortos, para que las transiciones sean rápidas y elegantes.

Transiciones

```
.element {  
    transition-duration: 0.5s;  
    width: 200px;  
    height: 200px;  
    background: grey;  
}  
  
.element:hover {  
    background: indigo;  
}
```

Demo

Transiciones

transition-property

En primer lugar, la propiedad `transition-property` se utiliza para especificar la propiedad a la que que afectará la transición. Podemos especificar la propiedad concreta (`width` o `color`, por ejemplo) o simplemente especificar `all` para que se aplique a todos los elementos con los que se encuentre. Por otro lado, `none` hace que no se aplique ninguna transición.

Transiciones

Propiedades	Valor
<code>all</code>	Aplica la transición a todas las propiedades css.
<code>none</code>	No aplica transición. El cambio se producirá de golpe (brusco).
<code>propiedad css</code>	Aplica la transición sólo a la propiedad css especificada.

Transiciones

Ten en cuenta que puedes animar todas las propiedades CSS afectadas utilizando all

Transiciones

```
.element {  
    /* Primer ejemplo: Anima todas las propiedades que cambien de estado */  
    transition-property: all;  
    transition-duration: 1s;  
  
    /* Segundo ejemplo: Anima sólo el ancho, el resto no transicionan, cambian de golpe */  
    transition-property: width;  
    transition-duration: 1s;  
}
```

Transiciones

Nota: Debes saber que no todas las propiedades CSS son animables (o transicionables). Por norma general, la mayoría de las propiedades que son cuantificables (valores numéricos, porcentajes, valores hexadecimales como colores, etc...) son animables, sin embargo, otras propiedades no lo son. Por ejemplo, la propiedad background-image para gradientes no se puede animar actualmente.

Transiciones

Demo

Transiciones

transition-delay

Por último, la propiedad transition-delay nos ofrece la posibilidad de retrasar el inicio de la transición un número de segundos determinado.

Si se omite, la transición comienza inmediatamente.

Transiciones

Veamos un pequeño ejemplo de todas estas propiedades de transición. Utilizaremos la función de tiempo linear, que luego te la cuento

Transiciones

```
a {  
    display: inline-block;  
    background: #ddd;  
    color: #222;  
    padding: 8px 16px;  
  
    transition-property: all;  
    transition-duration: 0.5s;  
    transition-timing-function: linear;  
}  
  
a:hover {  
    background: #fff;  
    color: darkred;  
    padding: 32px;  
    border: 1px solid darkred;  
}
```

Demo

Transiciones

- La propiedad background de color de fondo cambiará de #ddd a #fff
- La propiedad color de color de texto cambiará de #222 a #666
- La propiedad padding del tamaño del relleno cambiará de 8px a 25px
- La propiedad border cambiará de 0 a 1px solid #888.

Transiciones

Este último es un caso especial, puesto que cambia de estilos porque toma el estilo inicial por defecto, que es un borde de 0px de grosor. Cada una de estas transiciones se realizarán a un ritmo lineal, durante 0.5s de duración.

Transiciones

transition (shorthand)

Como siempre, podemos resumir todas estas operaciones en una propiedad de atajo denominada transition. Los valores del ejemplo superior, se podrían escribir como se puede ver a continuación (si no necesitas algún valor, se puede omitir)

Transiciones

```
.box {  
  /* transition: <property> <duration> <timing-function> <delay> */  
  transition: all 0.5s linear;  
  
  background: grey;  
  width: 200px;  
  height: 200px;  
}  
  
.box:hover {  
  background: indigo;  
}
```

Transiciones

Observa que estamos omitiendo el último valor. Si quisieramos un retardo de 2s a la hora de hacer una transición, bastaría con añadir un 2s después de linear.

Transiciones

Transiciones de entrada y de salida

Transiciones

Si nos fijamos bien, la transición del ejemplo anterior se aplica sólo al mover el ratón sobre el elemento (transición de entrada). Sin embargo, si movemos el ratón fuera del enlace, no se produce transición, sino que realiza el cambio de forma brusca. Esto ocurre porque le estamos diciendo que solo realice la transición cuando tenemos el ratón encima (:hover), en caso contrario no lo hará porque no hay definidas propiedades de transición.

Transiciones

```
a {  
    background: black;  
    color: white;  
    padding: 8px;  
    transition: background 0.2s linear;  
}  
  
a:hover {  
    background: red;  
    transition: background 2s linear;  
}
```

Transiciones

También es posible que queramos indicar, por ejemplo, diferentes duraciones dependiendo de la propiedad CSS. Por ejemplo, que tarde mucho en cambiar el ancho del elemento, pero muy poco en cambiar el color de fondo.

Para ello, podemos separar con comas las diferentes propiedades que queremos transicionar

Transiciones

```
.box {  
    width: 200px;  
    height: 200px;  
    background: grey;  
    transition:  
        width 3s,  
        background-color 0.5s;  
}  
  
.box:hover {  
    width: 400px;  
    background: deeppink;  
}
```

@keyframes

@keyframes

Cuando ya hemos utilizado propiedades CSS como animation (o de su familia, animation-*), nos falta una parte muy importante: definir los fotogramas de la animación.

@keyframes

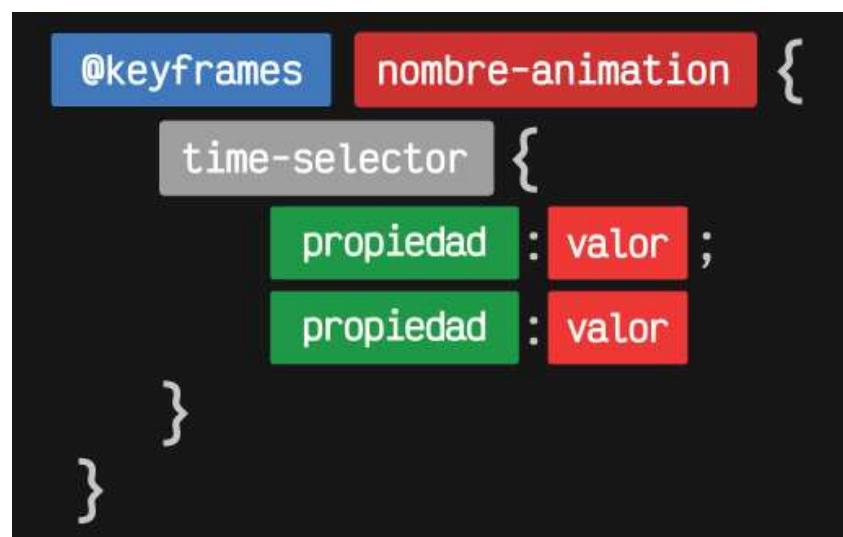
Una animación está formada por varios fotogramas, una secuencia de imágenes (30-60 fotogramas por segundo, por ejemplo) que mostradas una detrás de otra generan el efecto de movimiento que conocemos de una animación.

@keyframes

En CSS, los fotogramas se crean a partir de propiedades CSS, y no hace falta definir tantos fotogramas. Sólo crearemos fotogramas clave y el resto de fotogramas los generará el navegador.

Para definir esos fotogramas clave, utilizaremos la regla @keyframes, la cuál es muy sencilla de utilizar. Se basa en el siguiente esquema

@keyframes



@keyframes

Cada uno de estos time-selector será un momento clave de cada uno de los fotogramas clave de nuestra animación, y ya veremos que pueden definirse muchos en una misma animación.

@keyframes

Selectores from y to

Parte	Descripción
@keyframes STRING name	Regla para darle un nombre y definir los fotogramas clave de una animación.
from	Fotograma clave inicial con los estilos CSS a aplicar. Equivalente a 0%.
to	Fotograma clave final con los estilos CSS a aplicar. Equivalente a 100%.
PERCENT	Porcentaje específico de la animación con los estilos CSS a aplicar. Permite decimales.

```
@keyframes change-color {  
    from { background: red; } /* Primer fotograma */  
    to { background: green; } /* Segundo y último fotograma */  
}
```

@keyframes

Como vemos, hemos nombrado change-color la animación, que parte de un primer fotograma clave con el fondo rojo hasta un último fotograma clave con fondo verde.

@keyframes

Recuerda que no basta con definir la animación mediante @keyframes, también hay que asociar la animación al elemento o etiqueta HTML al que queremos aplicársela

@keyframes

```
.element {  
    width: 100px;  
    height: 100px;  
    animation: change-color 1.5s linear infinite;  
}  
  
@keyframes change-color {  
    from { background: red; } /* Primer fotograma */  
    to { background: green; } /* Segundo y último fotograma */  
}
```

@keyframes

La magia de la regla @keyframes es que nosotros solo declaramos los fotogramas clave, mientras que el navegador irá generando los fotogramas intermedios para conseguir una animación fluida.

@keyframes

Los selectores from y to son muy similares a colocar 0% y 100%, así que los modificaremos y de esta forma podremos ir añadiendo nuevos fotogramas intermedios. Vamos a añadir un fotograma intermedio e indentando, ahora sí, correctamente el código

@keyframes

```
@keyframes change-color {  
    0% {  
        background: red;          /* Primer fotograma */  
    }  
    50% {  
        background: yellow;     /* Segundo fotograma */  
        width: 400px;  
    }  
    100% {  
        background: green;      /* Último fotograma */  
    }  
}  
  
.animated {  
    background: grey;  
    color: #FFF;  
    width: 150px;  
    height: 150px;  
    animation: change-color 2s ease 0s infinite;  
}
```

Ejemplo de %

@keyframes

En este caso, la animación va a progresar por tres fotogramas clave, por lo que irá desde el color rojo, al color amarillo, para finalizar en el color verde. Esta animación funcionará durante 2 segundos y se realizará una sola vez (por defecto las animaciones solo se realizan una vez, salvo que le indiques infinite o el número deseado)

Funciones de tiempo

Funciones de tiempo

Tanto cuando hablamos de transiciones como cuando hablamos de animaciones tenemos dos propiedades que se aplican exactamente igual y se encargan de definir el ritmo o transcurso de la animación o transición en cuestión.

Funciones de tiempo

Propiedades	Descripción	Valor
transition-timing-function	Ritmo de la transición	FUNCTION cubic-bezier(A, B, C, D)
animation-timing-function	Ritmo de la animación	FUNCTION cubic-bezier(A, B, C, D)

Funciones de tiempo

Función predefinida

Quizás, la forma más sencilla de indicar el ritmo de una animación, sería especificar una de las siguientes palabras clave (valores) que existen para indicar en las propiedades anteriores. Cada una de ella, realiza la animación a un ritmo diferente

Funciones de tiempo

Si no se indica ninguna función de tiempo concreta, CSS utilizará la función de tiempo ease. Sin embargo, podemos utilizar cualquiera de las anteriores.

Funciones de tiempo

```
.element {  
    width: 200px;  
    height: 50px;  
    margin: 5px;  
    background: indigo;  
    color: white;  
    animation: animated 2s alternate infinite;  
}  
  
@keyframes animated {  
    to { width: 500px; }  
}  
  
.ease { animation-timing-function: ease; }  
.linear { animation-timing-function: linear; }  
.ease-in { animation-timing-function: ease-in; }  
.ease-out { animation-timing-function: ease-out; }  
.ease-in-out { animation-timing-function: ease-in-out; }
```

```
<div class="element ease">ease</div>  
<div class="element linear">linear</div>  
<div class="element ease-in">ease-in</div>  
<div class="element ease-out">ease-out</div>  
<div class="element ease-in-out">ease-in-out</div>
```

Funciones de tiempo

steps

Las funciones de salto permiten establecer una transición o animación en un número concreto de pasos muy específico, que se pasa por parámetro n. En este ejemplo, estamos indicando que la animación debe tener exactamente 5 pasos, por lo que en lugar de ser una transición o animación fluida, será una escalonada en el número de pasos dados

Funciones de tiempo

```
.container {  
    width: 600px;  
    background: grey;  
}  
  
.element {  
    width: 200px;  
    height: 200px;  
    background: indigo;  
    animation: move 5s infinite;  
    animation-timing-function: steps(5);  
}  
  
@keyframes move {  
    to { transform: translateX(500px); }  
}
```

```
<div class="container">  
    <div class="element"></div>  
</div>
```

Ejemplos de librerías

AOS

Animate On Scroll Library

[Enlace](#)

Animate.css

Just-add-water CSS animations

[Enlace](#)

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

{CSS}



Clase 35

<Índice>

Clase 35

Preprocesadores CSS => SASS

¿Qué son?

¿Por qué son importantes?

Ventajas

Desventajas

Instalación inicial básica

Elementos básicos

¿Qué son?

¿Qué son?

Programas que trabajan con archivos

No modifican el estándar de CSS

No hacen magia, aunque parezca que sí

Nos ayudan a hacer mejor las cosas y ser más
organizados

Nos ayudan a modularizar

Nuestro flujo de trabajo es más cómodo

Concatenan, minifican etc...

¿Qué son?



CSS 1.0

CSS3

SASS

¿Qué son?



Esto va muy rápido

¿Por qué son importantes?

¿Por qué son importantes?

Nos traen a lenguajes no programáticos, todas las propiedades de los lenguajes de programación.

Nos da toda la flexibilidad que necesitamos

Ventajas de usar preprocesadores

Ventajas

Variables: Sass permite el uso de variables para almacenar colores, fuentes y otros valores CSS. Esto hace que el mantenimiento del código sea mucho más fácil, ya que puedes cambiar un valor en un solo lugar y tenerlo actualizado en todo el proyecto.

@keyframes

Anidación: Sass permite anidar selectores de manera que reflejen la jerarquía del DOM. Esto hace que los estilos sean más legibles y que la estructura del código sea más clara.

@keyframes

Mixins: Los mixins son una manera de reutilizar conjuntos de propiedades CSS en todo el sitio. Esto evita la repetición de código y facilita la gestión de patrones de diseño comunes.

@keyframes

Herencia y Extensión: Sass permite extender selectores heredando estilos de un selector a otro. Esto es útil para mantener las hojas de estilo DRY (Don't Repeat Yourself).

@keyframes

Funciones y Operaciones: Sass permite realizar operaciones matemáticas y usar funciones para manipular colores y otros valores. Esto puede ser útil para crear diseños responsivos y dinámicos.

@keyframes

Control Directives: Ofrece directivas de control como @if, @for, @each y @while, que permiten escribir estilos más dinámicos y con lógica condicional.

@keyframes

Organización: Sass facilita la organización de tus hojas de estilo, permitiéndote dividirlas en varios archivos y luego importarlas en un archivo principal. Esto hace que el código sea más modular y fácil de mantener.

@keyframes

Compatibilidad con CSS Puro: Cualquier archivo CSS válido es también un archivo Sass válido. Esto significa que puedes adoptar Sass gradualmente en tu proyecto sin necesidad de reescribir todo tu CSS existente.

@keyframes

Comunidad y Herramientas: Sass tiene una gran comunidad de usuarios y una amplia gama de frameworks y librerías (como Compass o Bourbon), lo que facilita el desarrollo y ofrece soluciones preconstruidas para problemas comunes.

@keyframes

Mejora en el Rendimiento de Desarrollo: Aunque Sass necesita ser compilado en CSS, herramientas modernas como Webpack o Gulp hacen este proceso transparente y rápido, mejorando la eficiencia en el desarrollo.

@keyframes

En resumen, Sass añade una capa de potencia y flexibilidad a CSS, lo que puede resultar en un desarrollo más rápido, un código más limpio y fácil de mantener, y una mayor escalabilidad en proyectos web complejos.

Desventajas

Desventajas

Curva de aprendizaje inicial
Preprocesamiento para modificar

Instalación básica

Instalación básica

[Live Sass Compiler](#)

Elementos básicos

Elementos básicos

[Tutorial de SASS](#)

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez