



Programación Web

2024

CSS

- Cascading Style Sheets, Hoja de Estilos en Cascada
- Sin CSS, lo que vemos en una página HTML son los estilos por defecto del navegador
- Con CSS, podemos controlar de manera exacta cómo queremos que nuestros elementos se vean
- CSS es un lenguaje basado en reglas, en base a éstas decidimos qué grupo de estilos queremos aplicar

CSS

PW2024

```
.clase {  
  background-color: darkblue;  
  border: 1px solid red;  
}
```

Diagram labels:

- Selector: points to `.clase`
- Propiedad: points to `border`
- Valor: points to `1px solid red`

```
.clase {  
  background-color: darkblue;  
  background-color: orange;  
}
```

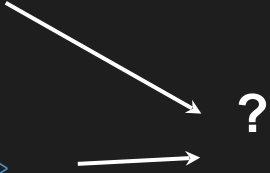
```
.clase {  
  font-size: 2rem;  
  /*color: white;*/  
}
```

Tipos de Selectores

- De tipo: h1, div, p, etc
- Por id: #titulo, #contenido
- Por clase: .container, .header, .footer
- Por atributo: a[href="https://example.org"]

CSS: Formas de utilizarlo

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    div {
      background-color: tomato;
    }
  </style>
  <title>Hola</title>
</head>
<body>
  <div style="background-color: gold;">
    Clase de programación web
  </div>
</body>
</html>
```



?

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

Especificidad

- Es la forma mediante la cual los navegadores deciden qué valores de una propiedad CSS son más relevantes para un elemento y, por lo tanto, serán aplicados
- Es un peso (importancia o valor) que se le asigna a una declaración CSS dada
- Cuando varias declaraciones tienen la misma especificidad, se aplica la última que se encuentra en el CSS

Especificidad - de menor a mayor

- De tipo (h1, p, etc.) y pseudo-elementos (::before, ::after)
- De clase (.contenedor), de atributo (a[href="<https://example.org>"]) y pseudo-clases (:hover, :focus, etc.)
- De id (#titulo)
- Los inline styles siempre sobre-escriben cualquier otra declaración de estilos, por lo que tienen la mayor especificidad

Pseudo-clases

- Es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado
- Ej.: `:hover` aplicará un estilo cuando el usuario haga hover sobre el elemento especificado por el selector
- Permiten aplicar un estilo a un elemento no sólo en relación con el contenido de la página, sino también en relación a factores externos como el historial del navegador (`:visited`), el estado de su contenido (`:checked`), etc.

```
selector:pseudoclase { propiedad: valor; }
```


Pseudo-elementos

- Se añaden a los selectores y permiten añadir estilos a una parte concreta del documento
- Ej.: `::marker` permite cambiar el estilo de los indicadores de una lista
- Ej.: `::placeholder` permite cambiar el estilo del texto usado como placeholder en un input

```
selector::pseudo-elemento { propiedad: valor; }
```

Combinadores

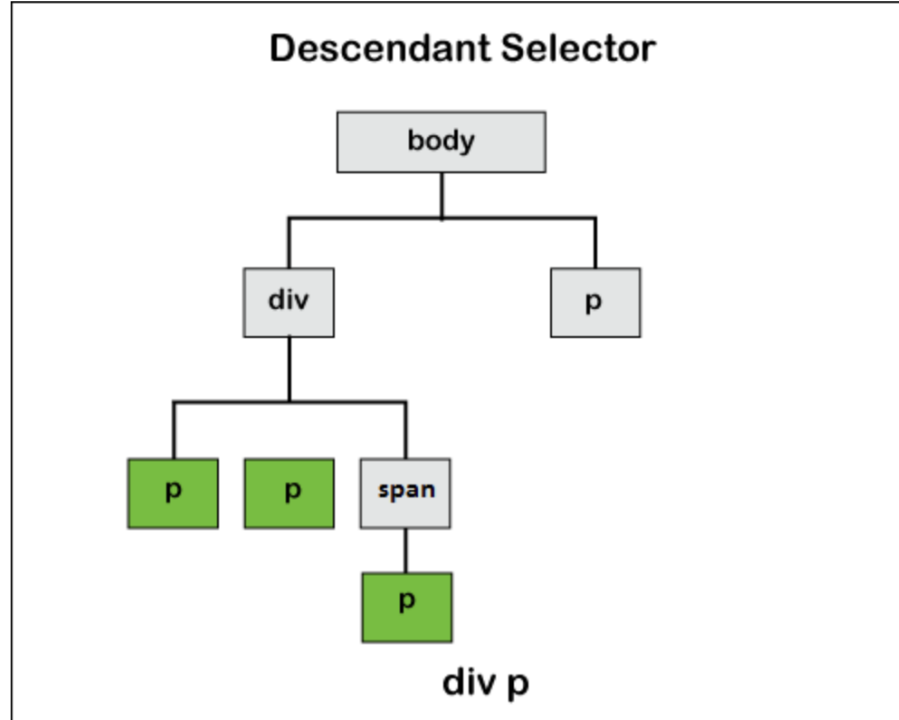
- Combinan selectores de manera que proporciona una relación útil entre ellos y la ubicación del contenido en el documento
- Diferentes tipos:
 - De descendientes
 - De elementos hijo
 - De hermanos adyacentes
 - De hermanos generales

Combinadores - De descendiente

- Seleccionan elementos que son descendientes de otros selectores
- No es necesario que sean hijos directos
- Ej:

```
body article p {  
  color: red;  
}
```

Combinadores - De descendiente

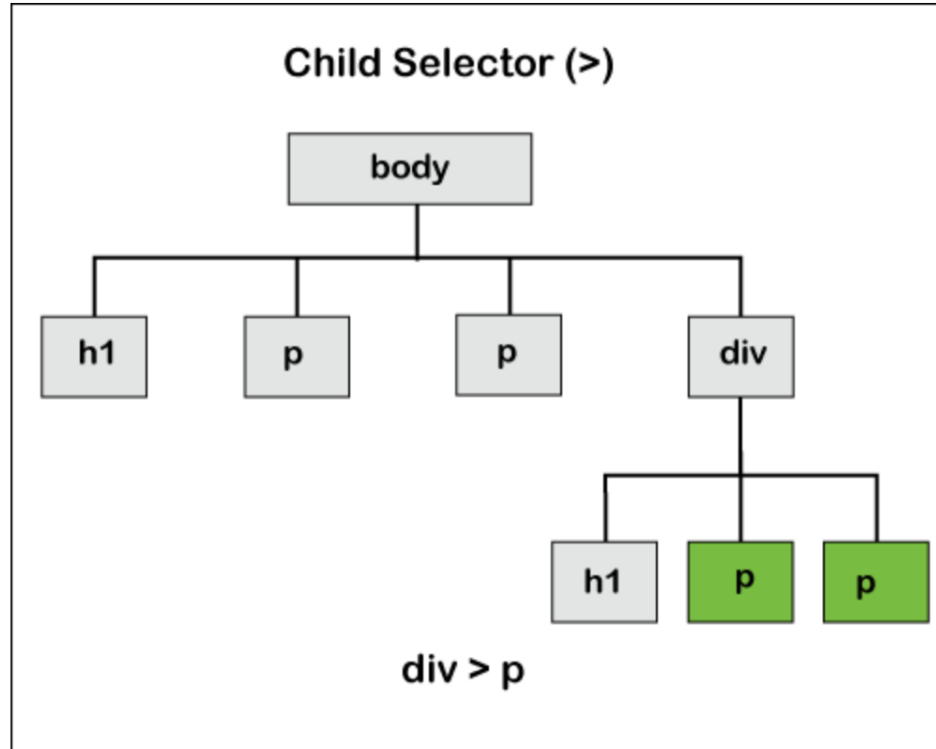


Combinadores - De elementos hijos

- Selecciona elementos que son hijos directos del selector
- Se denota utilizando >
- Ej.:

```
ul > li {  
    border-top: 5px solid red;  
}
```

Combinadores - De elementos hijos

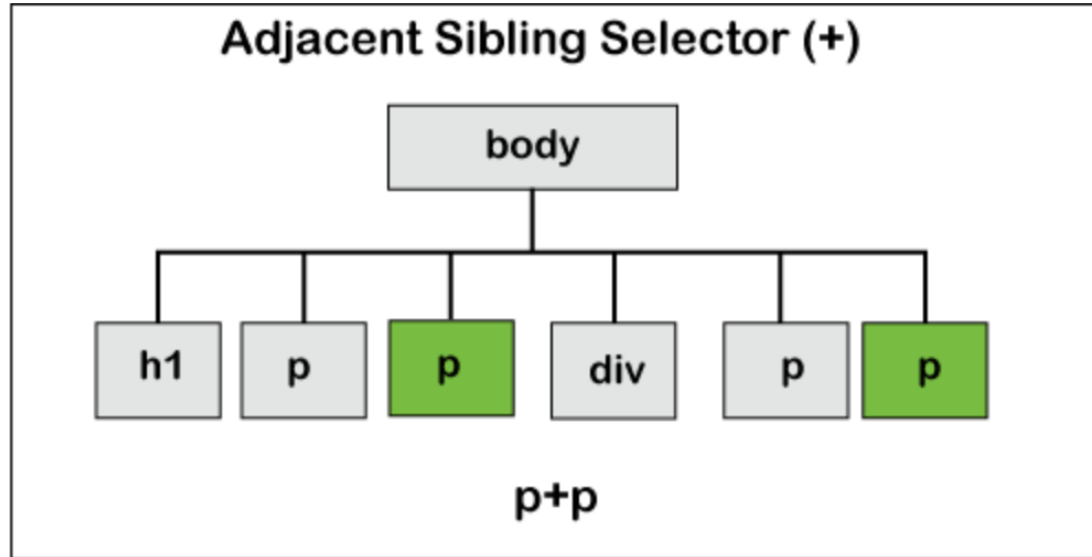


Combinadores - De hermanos adyacentes

- Se usa para seleccionar un elemento que se encuentra justo al lado de otro elemento en el mismo nivel de la jerarquía
- Se denota utilizando +
- Ej.:

```
p + p {  
  color:  red;  
}
```

Combinadores - De hermanos adyacentes

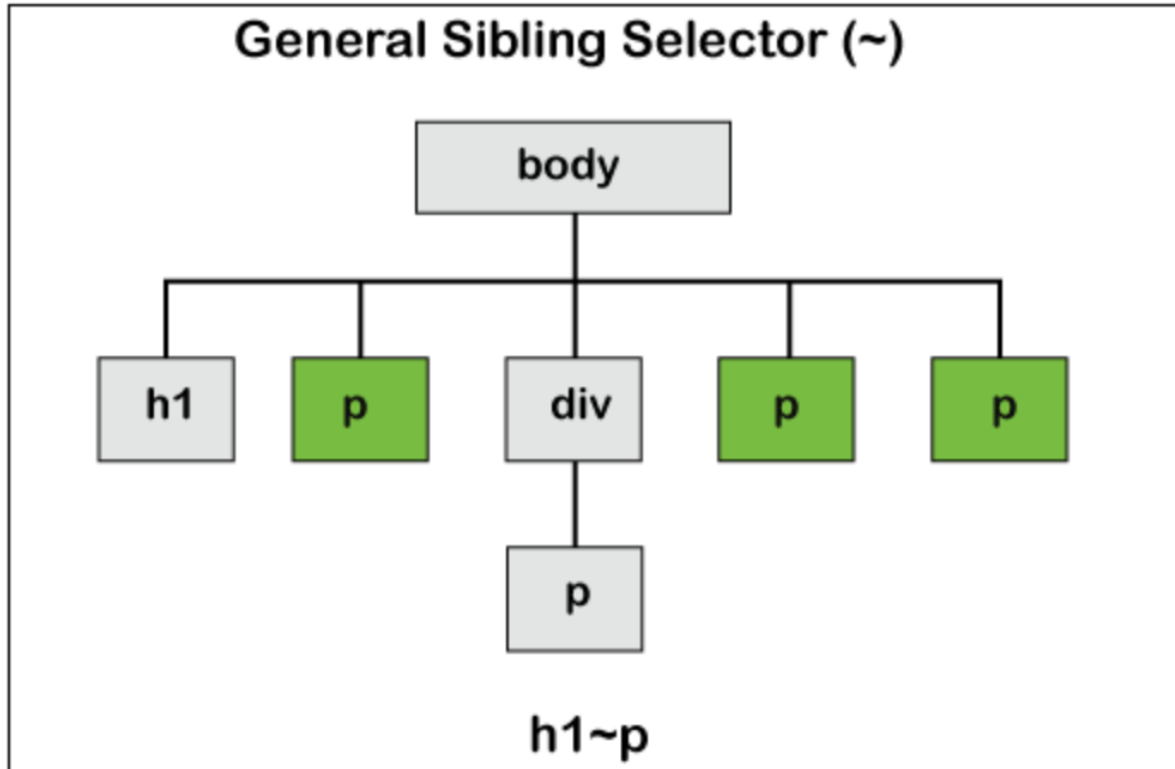


Combinadores - De hermanos generales

- Selecciona los hermanos de un elemento, incluso si no son directamente adyacentes
- Se denota utilizando ~
- Ej.:

```
h1 ~ p {  
  color: red;  
}
```

Combinadores - De hermanos generales



CSS: Unidades de medida

Absolute length

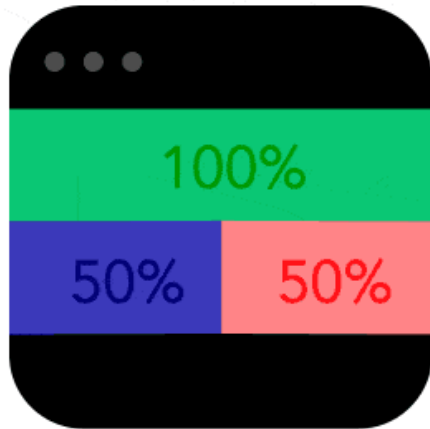
Units	Full Form	Font-size
px	pixel	font-size: 16px
pt	point	font-size: 12pt
pc	pica	font-size: 1pc
in	inches	font-size: 1in
cm	centimeter	font-size: 1cm
mm	millimeter	font-size: 10mm
q	quarter	font-size: 16q

Relative Length

Units	Definition
em	Relative to the font-size of the current element
ex	Relative to the font's x-height
%	Relative to the enclosing parent element in percent
ch	Relative to the width of the digit "0"
rem	Relative to the font-size of the root element
vw	Relative to 1% of the width of the viewport
vh	Relative to 1% of the height of the viewport
vmin	Relative to 1% of the viewport (smaller between vw & vh)
vmax	Relative to 1% of the viewport (bigger between vw & vh)

CSS: Unidades relativas vs estáticas

Relative Units



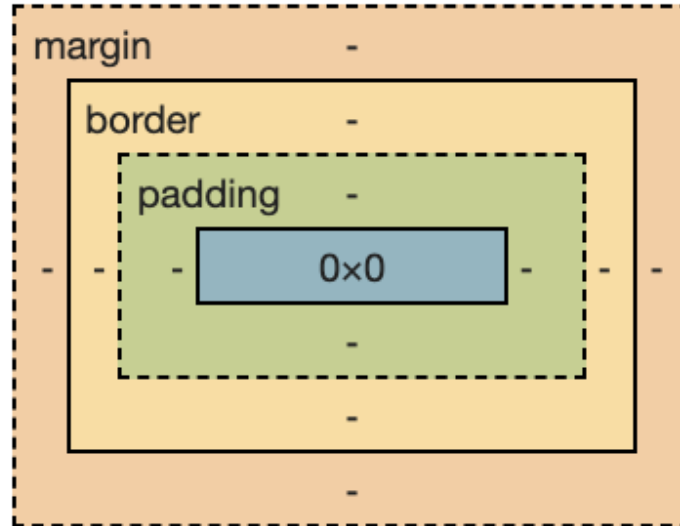
Static Units



CSS: Modelo de caja

- Box model o modelo de caja
- Todo en CSS tiene una caja alrededor
- Dos tipos de cajas:
 - Cajas en bloque (block)
 - Cajas en línea (inline)
- Si un elemento tiene asignado un modelo de caja en bloque, ocupará todo el ancho que tenga disponible
- Además, hacen un “salto de línea” en el documento
- Si un elemento tiene un modelo de caja en línea, ocupa solamente el espacio que necesita, y no hace “salto de línea”
- Se puede cambiar con la propiedad **display** en CSS

CSS: Modelo de caja



CSS: block, inline, inline-block

block	inline	Inline-block
Respetan todos los márgenes y paddings	Respetan márgenes y paddings horizontales, pero no verticales	Respetan todos los márgenes y paddings
Hace un salto de línea luego de mostrar el elemento	No hace salto de línea	No hace salto de línea
Toma todo el ancho posible si el mismo no se define	No se le puede setear ancho (altura tampoco)	Toma todo el ancho posible si el mismo no se define

CSS: Modelo de caja flexible - Flexbox

- En CSS, inicialmente se utilizaba el posicionamiento (static, relative, absolute, etc.), los elementos en línea o en bloque, u otras propiedades para posicionar o distribuir los elementos en pantalla
- Hoy en día utilizar sólo esos mecanismos nos limita mucho a la hora de enfrentar diferentes desafíos, como puede ser el manejo y la organización de múltiples elementos en diferentes resoluciones, etc.
- Flex (también llamado flexbox) es un sistema de elementos flexibles, en la que los elementos HTML se adaptan y colocan automáticamente y es más fácil personalizar los diseños de una página web.

CSS: Flexbox - Conceptos



CSS: Flexbox - Conceptos

Propiedad	Valor	Significado
<code>flex-direction</code>	<code>row</code> <code>row-reverse</code> <code>column</code> <code>column-reverse</code>	Cambia la orientación del eje principal.

Valor	Descripción
<code>row</code>	Establece la dirección del eje principal en horizontal .
<code>row-reverse</code>	Establece la dirección del eje principal en horizontal invertido .
<code>column</code>	Establece la dirección del eje principal en vertical .
<code>column-reverse</code>	Establece la dirección del eje principal en vertical invertido .

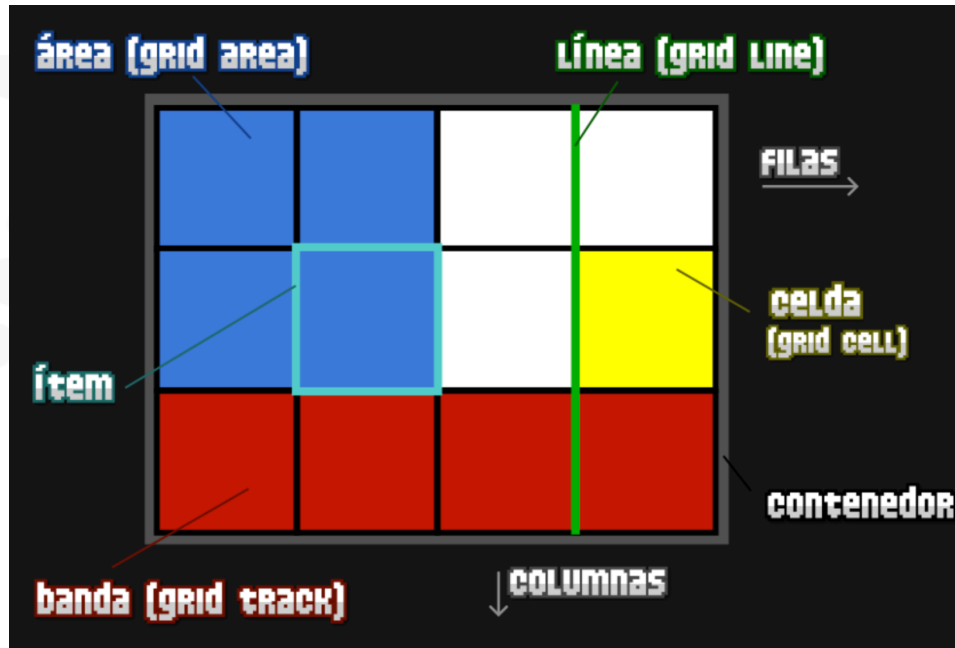
CSS: Flexbox - Alineación

Propiedad	Valor	Actúa en eje
<code>justify-content</code>	<code>start</code> <code>end</code> <code>center</code> <code>space-between</code> <code>space-around</code> <code>space-evenly</code>	1
<code>align-items</code>	<code>start</code> <code>end</code> <code>center</code> <code>stretch</code> <code>baseline</code>	2
<code>align-content</code>	<code>start</code> <code>end</code> <code>center</code> <code>space-between</code> <code>space-around</code> <code>space-evenly</code> <code>stretch</code>	2

Valor	Descripción
<code>start</code>	Agrupar los ítems al inicio del eje principal.
<code>end</code>	Agrupar los ítems al final del eje principal.
<code>center</code>	Agrupar los ítems al centro del eje principal.
<code>space-between</code>	Distribuye los ítems dejando espacio entre ellos .
<code>space-around</code>	Distribuye los ítems dejando espacio alrededor de ellos .
<code>space-evenly</code>	Distribuye como <code>space-around</code> , pero con un espacio exactamente igual alrededor de ellos.

CSS: Grillas - Grid

- Si Flex nos da la posibilidad de gestionar varios elementos de una manera unidimensional, Grid hace algo similar pero de forma bidireccional



CSS: Posicionamiento

- Cómo mencionamos anteriormente, existen otros valores para la propiedad display además de **flex**, **grid**, **block**, **inline**, **inline-block**, y **none**
- Estos otros valores nos permiten manejar o realizar acciones muy concretas con ciertos elementos, como por ejemplo dejarlos anclados al inicio de la pantalla sin importar en que parte de nuestra web el usuario se encuentre
- Estos son: **static**, **relative**, **absolute**, **fixed**, **sticky**

CSS: Posicionamiento

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 a un contenedor padre.
fixed	Posicionamiento fijo. Idem al absoluto, pero aunque hagamos scroll no se mueve.
sticky	Posicionamiento fijo con desplazamiento («pegajoso»). Muy usado para pegar elementos a una zona.

CSS: Posicionamiento - Relative

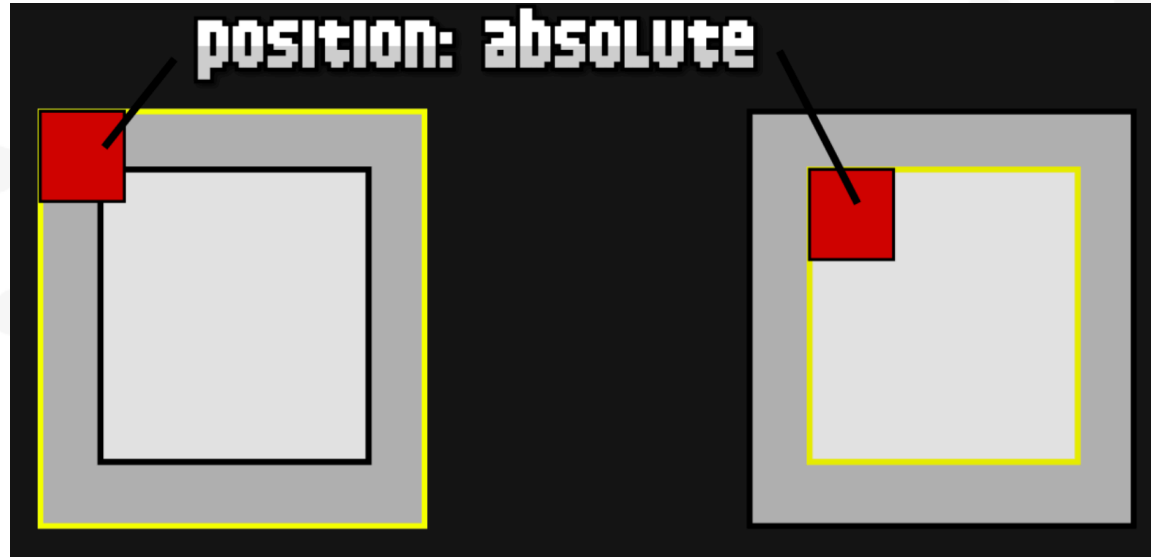
Nos permite posicionar el elemento a una ubicación relativa a su posición original

RELATIVE



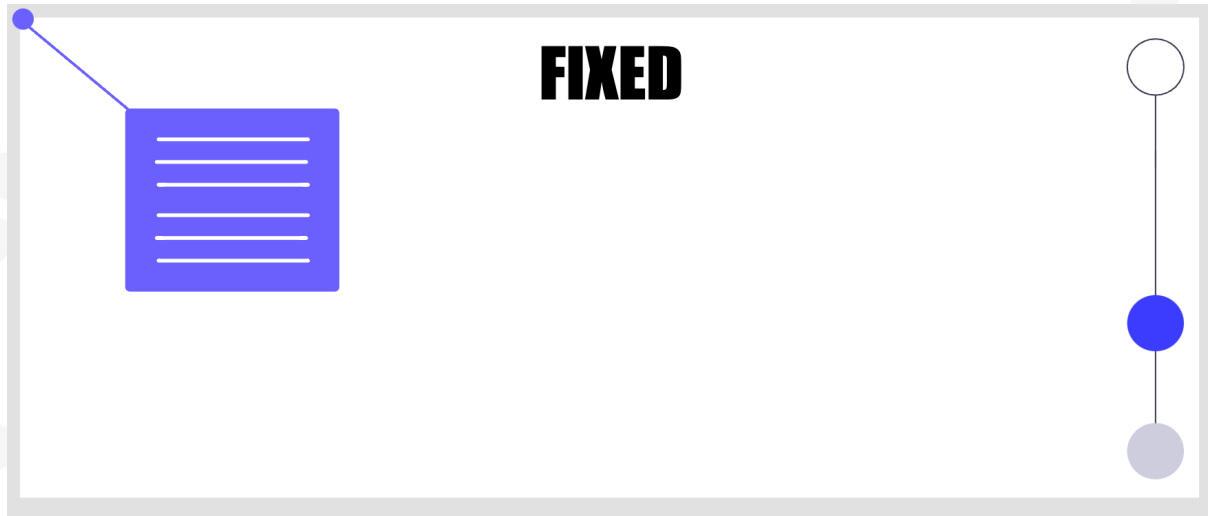
CSS: Posicionamiento - Absolute

Nos permite posicionar el elemento con respecto a su **contenedor posicionado** mas cercano



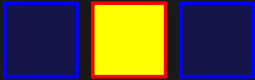
CSS: Posicionamiento - Fixed

Nos permite posicionar el elemento respecto a **la ventana** del navegador



CSS: Posicionamiento - Sticky

Mezcla entre **relative** y **fixed**

<pre>position: static;</pre>	<p>In this demo you can control the position property for the yellow box.</p> 	<pre>position: static;</pre>	<p>To see the effect of sticky positioning, select the position: sticky option and scroll this container.</p> <p>The element will scroll along with its container, until it is at the top of the container (or reaches the offset specified in top), and will then stop scrolling, so it stays visible.</p> <p>The rest of this text is only supplied to make sure the container overflows, so as to enable you to scroll it and see the effect.</p> <hr/> <p>Far out in the uncharted backwaters of the</p>
<pre>position: relative; top: 40px; left: 40px;</pre>		<pre>position: relative; top: 40px; left: 40px;</pre>	
<pre>position: absolute; top: 40px; left: 40px;</pre>		<pre>position: absolute; top: 40px; left: 40px;</pre>	
<pre>position: sticky; top: 20px;</pre>		<pre>position: sticky; top: 20px;</pre>	

<https://developer.mozilla.org/es/docs/Web/CSS/position>

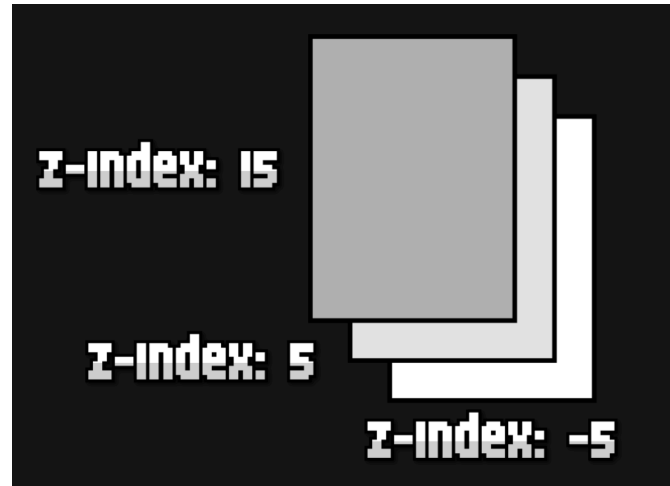
CSS: Modificando la posición

Al utilizar un modo de posicionamiento distinto a static, podemos utilizar una serie de propiedades para modificar la posición de un elemento

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.
inset	auto SIZE	Propiedad de atajo para utilizar todas las anteriores.

CSS: z-index

Al utilizar un modo de posicionamiento distinto a static, podemos utilizar la propiedad z-index para establecer profundidad entre los elementos posicionados



Propiedad	Valor	Significado
z-index	auto NUMBER	Coloca un elemento en el eje de profundidad, colocándolo debajo o encima de otros elementos.

CSS: Responsivo

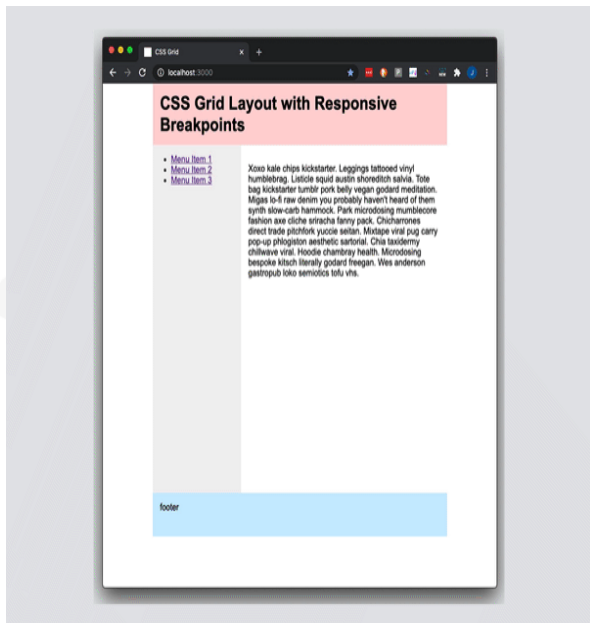
- Una aplicación web responsiva ofrece el mismo sitio a cualquier dispositivo
- El sitio tiene la fluidez necesaria para cambiar su organización y apariencia basándose en el tamaño y la orientación del dispositivo
- Usamos CSS para asegurarnos que este comportamiento exista, usando lo que llamamos breakpoints

CSS: Media Query

- Permiten aplicar estilos CSS según características del dispositivos como la resolución de la pantalla o el ancho del viewport del navegador
- Son comúnmente usadas para aplicar estilos de forma condicional
- Ej.:

```
div {  
  width: 100px;  
}  
  
@media (width > 425px) {  
  div {  
    width: 300px;  
  }  
}
```

CSS: Breakpoints



```
/* 🌐 CSS Media Queries */

/* 📌 Traditional Way */

/* Small screens (mobile devices) */
@media only screen and (max-width: 767px) {
  /* Styles go here */
}

/* Medium screens (tablets) */
@media only screen and (min-width: 768px) and (max-width: 1023px) {
  /* Styles go here */
}

/* 📌 Using the meta tag to make it easier */

/* <!-- Adding width on the meta tag for responsive design --> */
/* 📌 The following meta tag must be added inside the <head> tab in the html file */
/* <meta name="viewport" content="width=device-width, initial-scale=1.0"> */

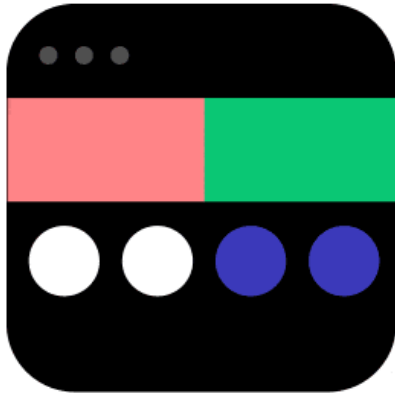
/* 📌 Shortcut (New Way) */

/* Small screens (mobile devices) */
@media (max-width: 767px) {
  /* Styles for small screens go here */
  body {
    background-color: lightblue;
  }
}

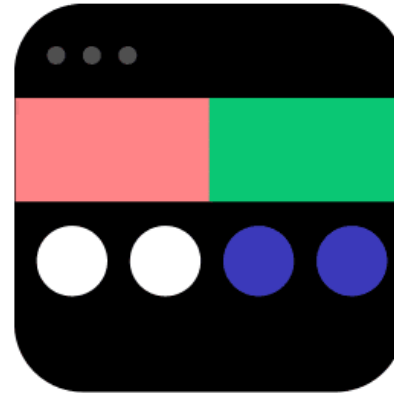
/* Medium screens (tablets) */
@media (min-width: 768px) and (max-width: 1023px) {
  /* Styles for medium screens go here */
  body {
    background-color: lightgreen;
  }
}
```

CSS: Ejemplo breakpoints

With Breakpoints



Without Breakpoints



CSS: Ejemplo valores máximos

Max width



No max width



Mobile first

- Implica diseñar una aplicación web pensando primero en los dispositivos móviles
- Luego, se hacen las modificaciones necesarias a nivel de CSS y estructura para adaptarlo a tablets, laptops, y monitores de gran tamaño
- Es importante porque fuerza al desarrollador a enfocarse solamente en lo que es esencial, dado que hay menos espacio disponible
- Luego, el diseño para dispositivos de mayor tamaño se fortalece dado que al expandir el diseño uno ya sabe lo que es importante y no pierde el foco

Mobile first

PW2024

