



DISEÑO DE INTERFACES WEB, 2º CFGS DAW

3

CSS3: Aspectos avanzados

OBJETIVOS DE LA UNIDAD:

- Conocer los aspectos más avanzados de CSS3.



2. CSS3

1. ¿QUÉ ES CSS (Cascading Style Sheets) ?

CSS (Cascading Style Sheets, Hojas de Estilo en Cascada en español) es un **lenguaje de estilo** utilizado para definir la presentación y el diseño de **documentos HTML** o de cualquier lenguaje de marcado **basado en XML**.

El lenguaje CSS se ha ido creando a lo largo del tiempo en **varios niveles**. Cada nivel de CSS se ha construido sobre el anterior, generalmente añadiendo funcionalidades nuevas.

En la página oficial de W3C (w3.org/Style/CSS/) se pueden consultar todas las publicaciones relacionadas con las novedades del estándar CSS en sus diferentes versiones.





2. CSS3

2. SOPORTE CSS EN NAVEGADORES

Cada navegador ofrece un **soporte CSS distinto**. Por este motivo, es necesario comprobar que nuestras webs se **visualizan correctamente** en los diferentes navegadores. A continuación, se presentan algunas **herramientas útiles** para verificar la compatibilidad y resolver problemas de CSS:

- **Can I use (caniuse.com)**: Esta herramienta en línea permite verificar el soporte de diferentes características de CSS en varios navegadores y versiones.
- **Browserstack (browserstack.com)**: Es una plataforma que permite probar sitios web en diferentes navegadores y dispositivos para verificar la compatibilidad de CSS.
- **Developer Tools en los navegadores**: Chrome, Firefox, Safari y Edge ofrecen herramientas integradas de desarrollo que permiten inspeccionar y modificar el código CSS para solucionar problemas y ajustar el diseño.
- **CSS Validator (jigsaw.w3.org/css-validator/)**: Es una herramienta de la W3C que verifica la validez del código CSS y muestra errores y advertencias para corregir problemas.



2. CSS3

2.1 RESET CSS

Un archivo "**reset.css**" es un archivo CSS que se utiliza en desarrollo web para establecer estilos predeterminados uniformes en todos los navegadores.

Su objetivo es "**reiniciar**" o "**resetear**" los estilos predeterminados de los elementos HTML, lo que permite a los diseñadores web partir de una base consistente al **diseñar sus sitios web**.

Cómo usar reset.css:

- Guardar el archivo reset.css en tu proyecto web.
- Enlazarlo en tu página HTML **antes** de otros estilos personalizados.
- Luego, agrega tus propios estilos personalizados.

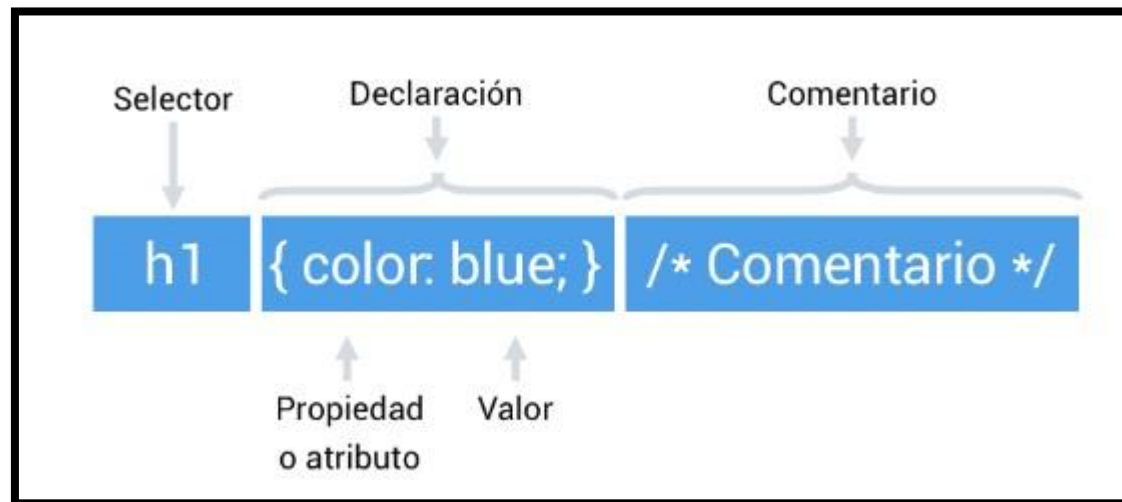
Puedes crear tu propio reset.css o elegir algunos ya diseñados. Algunos ejemplos son [Normalize.css](#) o [Reset.css](#)



2. CSS3

3. SINTAXIS CSS

En CSS se utiliza la **siguiente sintaxis** para asignar valores a las propiedades de cada selector:





2. CSS3

4. ESTILOS CSS EN UN DOCUMENTO HTML

Hay tres formas de aplicar **estilos CSS** en un documento HTML:

- [En línea.](#)
- [Incrustado en la cabecera.](#)
- [Mediante hojas de estilo externas.](#)





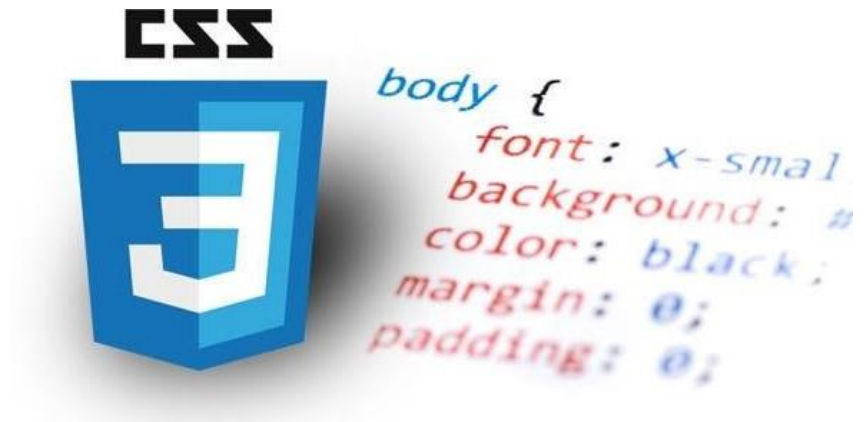
2. CSS3

5. SELECTORES

Los selectores nos ayudan a **indicar el elemento** sobre el que se van a aplicar los estilos. Los selectores pueden apuntar a **elementos específicos, clases, identificadores o incluso atributos de un elemento**.

Disponemos de los siguientes selectores:

- **Selector universal**
- **Selector etiqueta**
- **Selector clase**
- **Selector identificador**
- **Selector descendiente**
- **Combinación de selectores**
- **Selector de hijos**
- **Selector adyacente**





2. CSS3

5.1 SELECTOR UNIVERSAL

Sintaxis: `* { atributo:valor; }`

Ejemplo: `* { color: grey; }`

`/* El estilo se aplicará a todos los elementos de la página*/`

El **selector universal (*)** es útil cuando deseas aplicar un estilo a **todos los elementos** en una página web sin excepción. Es una forma rápida y sencilla de establecer reglas generales que afectarán a todos los elementos en el documento. Por ejemplo, si deseas **eliminar todos los márgenes y rellenos predeterminados** de los elementos en tu página para empezar desde cero, puedes usar el selector universal para restablecerlos.



2. CSS3

5.2 SELECTOR ETIQUETA

Sintaxis: etiqueta { atributo:valor }

Ejemplo: p {color: green;}

/* El estilo se aplicará a todos los elementos <p>.*/*

5.3 SELECTOR CLASE

Sintaxis: .clase { atributo:valor }

Ejemplo: .blend{color: red;}

/* El estilo se aplicará a cualquier elemento que tenga la clase .blend */



2. CSS3

5.4 SELECTOR IDENTIFICADOR

Sintaxis: #id { atributo:valor }

Ejemplo: #cent {color: blue;}

/* El estilo se aplicará al elemento que tenga el id #cent */

5.5 SELECTOR DESCENDIENTE

Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento padre, a cualquier nivel.

Sintaxis: selector1 selector2 selectorN {atributo: valor;}

/* El estilo se aplica sobre el selector N */

Ejemplo: div p { color: black;}

/* El estilo se aplica a todos los párrafos que se encuentren dentro de una etiqueta div */



2. CSS3

5.6 COMBINACIÓN DE SELECTORES

La combinación de selectores nos permite dar un estilo a todos los selectores indicados.

Sintaxis: selector1, selector2, selector3{atributo: valor;}
/* El estilo se aplica sobre los selectores indicados */

Ejemplo: div, p { color: orange;}
/* El estilo se aplica a todos los divs y párrafos */

5.7 SELECTOR DE HIJOS

Se usa para seleccionar un elemento que es hijo directo de otro elemento.

Sintaxis: selector1 > selector2 {atributo: valor;}
/* El estilo se aplica sobre el selector 2 */

Ejemplo: div > p { color: white;}
/* El estilo se aplica a todos los párrafos que sean hijos de un div */



2. CSS3

5.8 SELECTOR ADYACENTE

Se usa para seleccionar elementos que son hermanos, es decir, su elemento padre es el mismo y están seguidos en el código HTML.

Sintaxis: selector1 + selector2 { atributo: valor; }
/* El estilo se aplica al selector 2 */

Ejemplo: div + p { color: black; }
/* El estilo se aplica a todos los párrafos que sean hermanos de un div */



[Código Ejemplo](#)



2. CSS3



5.9 SELECTOR DE ATRIBUTOS

[Código Ejemplo](#)

Los **selectores de atributos** permiten elegir elementos HTML en función de sus atributos y/o valores de esos atributos. Los tipos de selectores de atributos son los siguientes:

- **[nombre_atributo]**, selecciona los elementos que tienen establecido el atributo llamado nombre_atributo independientemente de su valor.
- **[nombre_atributo=valor]**, selecciona los elementos que tienen establecido un atributo llamado nombre_atributo con un valor igual a valor.
- **[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.
- **[nombre_atributo|=valor]**, selecciona los elementos que tienen establecido un atributo llamado nombre_atributo, cuyo valor es una lista de valores, donde alguno comienza por valor.
- **[nombre_atributo\$=valor]** Selecciona aquellas etiquetas cuyo atributo acabe en ese valor.
- **[nombre_atributo^=valor]** Selecciona aquellas etiquetas cuyo atributo comience por ese valor.



2. CSS3

5.10 PSEUDO-CLASES PARA SELECCIÓN DE HIJOS O HERMANOS

Las pseudo-clases más importantes para la selección de hijos o hermanos son las que se muestran en la siguiente tabla

[Códigos Ejemplo](#)

Pseudo-clase	Descripción
:first-child	Primer hijo
:last-child	Último hijo
:first-of-type	Primer hermano de su tipo
:last-of-type	Último hermano de su tipo
:only-child	Hijos únicos
:only-of-type	Únicos hermanos de su tipo
:empty	Elementos que no tienen hijos
:nth-child(n)	Enésimo elemento hijo
:nth-last-child(n)	Enésimo elemento hijo contando desde el último
:nth-of-type(n)	Enésimo hermano de su tipo
:nth-last-of-type(n)	Enésimo hermano de su tipo comenzando desde el último



2. CSS3

ACTIVIDAD



A partir del código HTML **aplicar los estilos CSS solicitados.**

```

<title>Ejercicio de Selectores CSS Complejos</title>
<link rel="stylesheet" href="style.css">
<head>
</head>
<body>
  <div id="contenedor">
    <h1 id="principal">Título Principal</h1>
    <p class="importante">Este es un párrafo importante</p>
    <p>Este es un párrafo normal.</p>
  </div>
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
  </ul>
  <section>
    <h2>Sección 1</h2>
    <p class="resaltado">Este párrafo está resaltado</p>
  </section>
  <section>
    <h2>Sección 2</h2>
    <p class="resaltado">Este párrafo también es resaltado</p>
  </section>

```

Tarea en la Moodle

```

/* Peticiones de estilos */

1 /* 1. Selecciona el título principal dentro del div con el id "principal" */
2 /* 2. Selecciona el párrafo con la clase "importante" y estílolo en rojo */
3 /* 3. Selecciona el último elemento de la lista no ordenada y estílolo en azul */
4 /* 4. Selecciona el párrafo con la clase "resaltado" dentro del primer <section> y estílolo en verde */
5 /* 5. Selecciona el párrafo con la clase "resaltado" que está dentro del primer <div> y estílolo en naranja */
6 /* 6. Selecciona el primer título h2 en el primer <section> y estílolo en azul */
7 /* 7. Selecciona el segundo título h2 en el primer <section> y estílolo en verde */
8 /* 8. Selecciona el tercer título h2 en el primer <section> y estílolo en naranja */
9 /* 9. Selecciona todos los párrafos dentro del segundo <section> y estílolos en rojo */
10 /* 10. Selecciona los elementos de lista dentro de un <ul> y estílolos en azul */
11 /* 11. Selecciona los elementos de lista dentro de un <ul> y estílolos en verde */
12 /* 12. Selecciona todos los elementos de lista que contienen el texto "Elemento" y estílolos en naranja */
13 /* 13. Selecciona el primer elemento de lista en el segundo <section> y estílolo en azul */
14 /* 14. Selecciona el último elemento de lista en el segundo <section> y estílolo en verde */
15 /* 15. Selecciona el párrafo con la clase "importante" que está dentro del primer <div> y estílolo en rojo */
16 /* 16. Selecciona el primer párrafo dentro de un <div> que contiene el texto "importante" y estílolo en naranja */
17 /* 17. Selecciona el último párrafo dentro de un <div> que contiene el texto "importante" y estílolo en verde */
18 /* 18. Selecciona el segundo párrafo dentro del segundo <section> y estílolo en azul */
19 /* 19. Selecciona el párrafo con la clase "resaltado" que está dentro del primer <section> y estílolo en verde */
20 /* 20. Selecciona el párrafo con la clase "resaltado" que está dentro del segundo <section> y estílolo en naranja */

```



2. CSS3

6. UNIDADES DE MEDIDA

En CSS, hay varias unidades de medida que puedes utilizar para **especificar tamaños y dimensiones** de elementos en tu página web. El tamaño de los elementos de una web se puede expresar en **unidades absolutas y relativas**

6.1 UNIDADES ABSOLUTAS

Las **unidades absolutas** mantienen su aspecto y se visualizan siempre igual independientemente de las características del dispositivo. Uso para **medios impresos principalmente**.

in	Pulgadas (1 pulgada = 2.54 cm)
cm	Centímetros
mm	Milímetros
pt	Puntos (1 pt = 1/72 pulgadas)
pc	Picas (1 pica = 12 puntos)

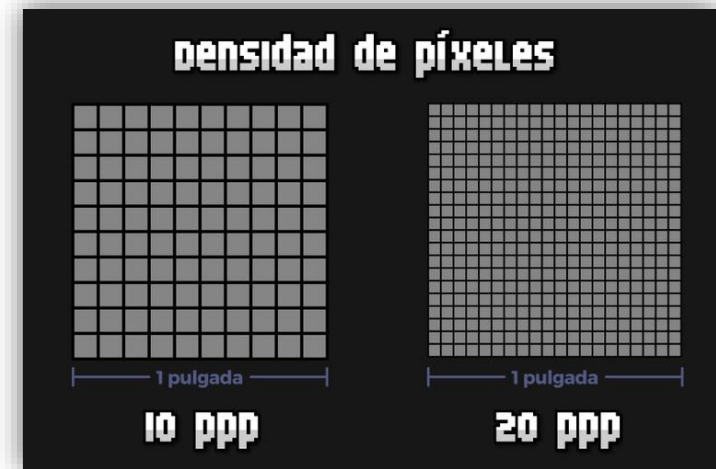


2. CSS3

6.2 UNIDADES RELATIVAS

Las **unidades relativas** se ajustan a cada tipo de dispositivo ya que dependen de la resolución de cada pantalla.

px	Píxeles (relativo al dispositivo)
rem	Relativo al tamaño de la fuente raíz (2rem significa 2veces el tamaño de la fuente raíz que suele ser 16px)
em	Relativo al tamaño de la fuente del elemento padre (2em significa 2veces el tamaño de la fuente del padre)
%	Porcentaje (relativo al elemento padre)
vh y vw	Medidas relativas de acuerdo al viewport 1vh = 1% de la altura del viewport 100vh = altura del viewport
fr	Flexible Grid Units (fr) Se utiliza en Grid Layout y representa una fracción del espacio disponible en un contenedor



Ten en cuenta que aunque **el tamaño en píxeles siempre sea el mismo**, puede observarse diferente en distintos dispositivos, ya que estos dispositivos no suelen tener la misma densidad de píxeles. Y para ello hay que entender lo que significa las siglas **ppp (píxeles por pulgada)** o **ppi en inglés** (points per inch)

[Código Ejemplo](#)



2. CSS3

6.3 USOS DE UNIDADES

Normalmente es recomendable **usar unidades relativas** en la medida de lo posible, ya que mejora la accesibilidad de la página web y permite que los documentos se adapten fácilmente a cualquier medio.

Por tanto, para la creación de una página web, el uso de **medidas absolutas queda descartado sobre todo a las medidas horizontal.**





2. CSS3

6.4 UNIDAD REM

Es especialmente útil para establecer tamaños proporcionales **al tamaño de fuente raíz**. Aunque no hay un criterio definido, el organismo W3C, recomienda el uso de la unidad rem para indicar el tamaño del texto. **El tamaño de los rem se establece en base al tamaño que tenga definido el navegador.**

Usualmente el tamaño de una fuente por defecto en los navegadores es de **16px**. Por tanto, tendríamos que **16px = 1rem** y podríamos definir la siguiente conversión entre unidades.

px	rem	%
12	0,750	75
14	0,875	87,5
16	1,000	100
18	1,125	112,5
20	1,250	125

Hay que tener en cuenta que el tamaño base definido en los navegadores puede ser modificado por los usuarios.



2. CSS3

6.5 PÍXELES

Es la unidad más utilizada y representa un **punto en la pantalla**. Se usa para tamaños fijos y proporciona **control preciso sobre el diseño**

6.6 PORCENTAJE (%)

Representa una **proporción del tamaño del elemento padre**. Es útil para hacer diseños fluidos y responsivos teniendo en cuenta la relación de los elementos con su contenedor padre.

6.7 VIEWPORT WIDTH (VW) Y VIEWPORT HEIGHT (VH)

Representan un porcentaje del ancho y alto de la ventana del navegador, respectivamente. Son útiles para crear **diseños responsive basados en el tamaño de la pantalla**.

6.7 FLEXIBLE GRID UNITS (fr)

Se utiliza en **Grid Layout** y representa una fracción del espacio disponible en un contenedor. Es útil para distribuir el espacio disponible entre elementos flexibles.



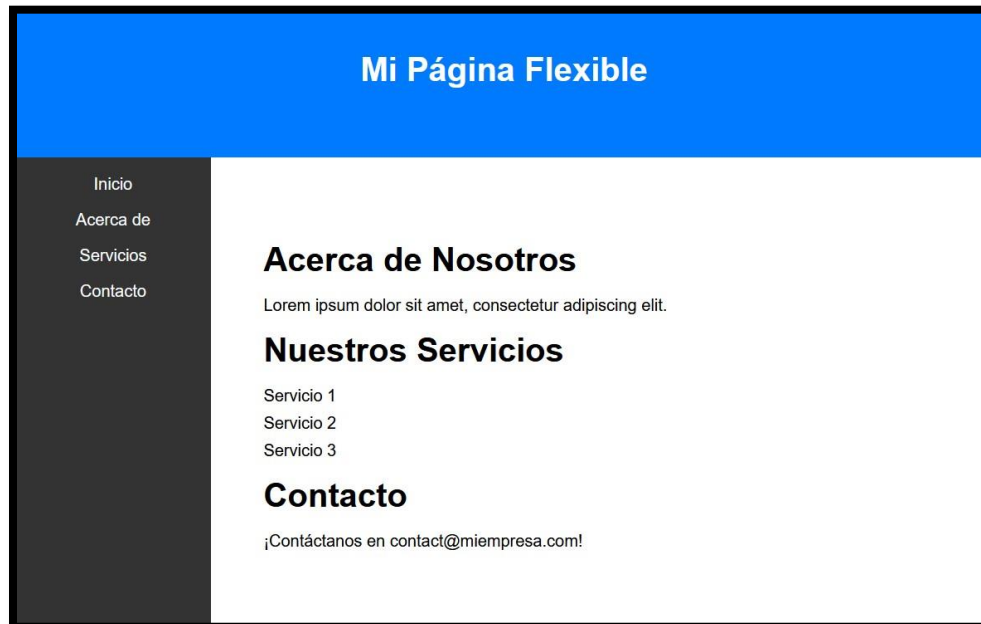
2. CSS3

ACTIVIDAD



Diseño de una Página Flexible con Unidades Relativas

Crear una página web que utilice unidades relativas de "porcentaje", "rem" y "vh" entre otras para lograr un diseño flexible.



Tarea en la Moodle



2. CSS3

7. COLORES Y FONDO

Las propiedades más utilizadas son:

Propiedad	Descripción	Valores
color	Color del texto	RGB HSL HEX nombre del color RGBA HSLA
background-color	Color de fondo	RGB HSL HEX nombre del color RGBA HSLA
background-image	Imagen de fondo	url(«...») none
background-repeat	Repetición de la imagen de fondo	repeat repeat-x repeat-y no-repeat
background-attachment	Desplazamiento de la imagen de fondo	scroll fixed
background-position	Posición de la imagen de fondo	percentage length left center right
background-size	Tamaño de la imagen de fondo	valor
opacity	Transparencia de un elemento	[0 – 1] (0 → totalmente transparente)



2. CSS3

8. PROPIEDADES DE TEXTO

[Códigos Ejemplo](#)

Propiedad	Descripción	Valores
text-indent	Desplazamiento de la primera línea del texto	longitud porcentaje
text-align	Alineamiento del texto	left right center justify
text-decoration	Efectos de subrayado y tachado	none underline overline line-through
letter-spacing	Espacio entre caracteres	normal longitud
word-spacing	Espacio entre palabras	normal longitud
text-transform	Transformación a mayúsculas / minúsculas	capitalize uppercase lowercase none
line-height	Tamaño del espacio entre líneas	longitud porcentaje
vertical-align	Alineación vertical	top middle bottom baseline sub super valor



2. CSS3

9. PROPIEDADES FUENTES

[Códigos Ejemplo](#)

Propiedad	Descripción	Valores
font-family	Familias de fuentes	nombre-familia *
font-style	Estilo de la fuente	normal italic oblique
font-variant	Convierte a mayúsculas manteniendo todas las letras en un tamaño inferior a la primera	normal small-caps
font-weight	Anchura de los caracteres. Normal = 400, Negrita = 700	normal bold bolder lighter 100 200 300 400 500 600 700 800 900
font-size	Tamaño de la fuente	xx-small x-small small medium large x-large xx-large larger smaller longitud porcentaje



2. CSS3

10. PROPIEDADES DE LAS LISTAS

[Códigos Ejemplo](#)

Propiedad	Descripción	Valores
list-style-type	Estilo aplicable a los marcadores visuales de las listas	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none
list-style-image	Imagen aplicable a los elementos de las listas	url() none
list-style-position	Posición del marcador dentro de la lista	inside outside
list-style	Permite establecer varios estilos de la lista en una sola propiedad	list-style-type list-style-position list-style-image



2. CSS3

11. PROPIEDADES DE LAS TABLAS

[Códigos Ejemplo](#)

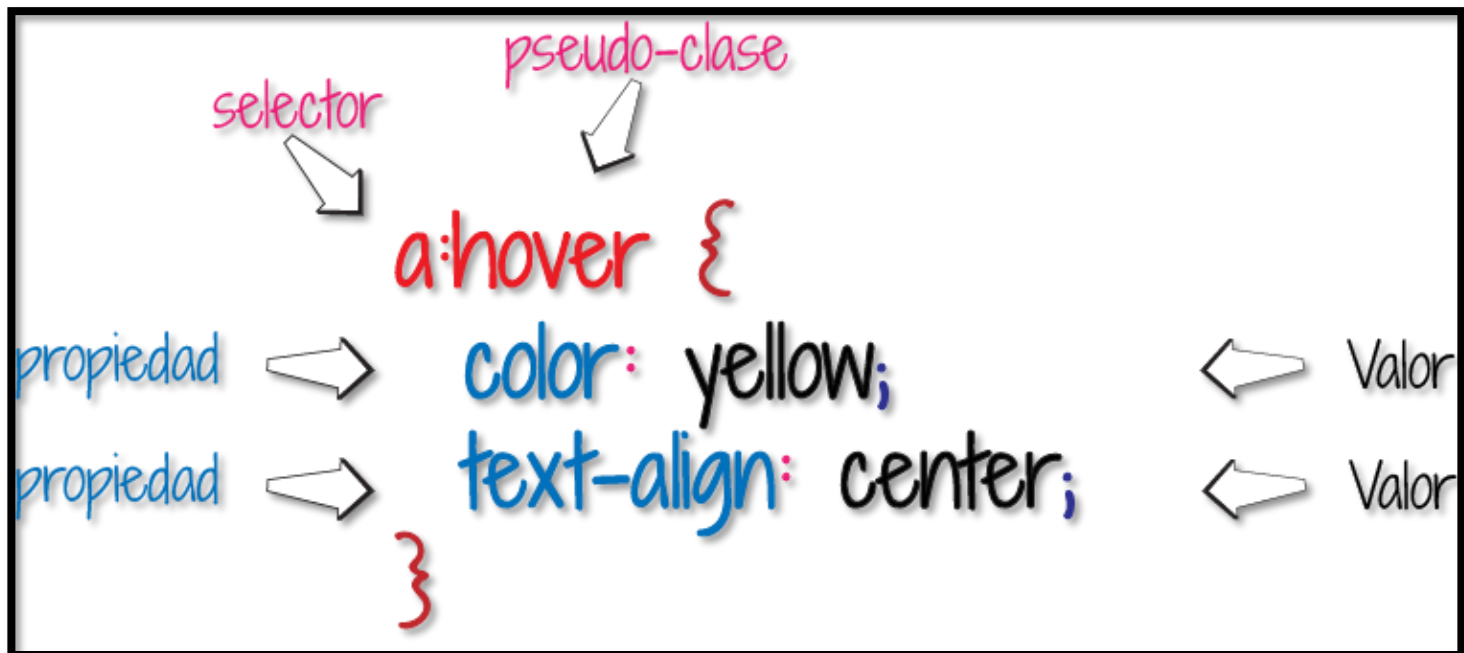
Propiedad	Descripción	Valores
caption-side	Posición del título respecto la tabla	top bottom
table-layout	Formato de las celdas, filas y columnas	auto fixed
border-collapse	Selección del modelo de los bordes	collapse separate
border-spacing	Espaciado entre los bordes de celdas adyacentes	longitud
empty-cells	Visibilidad de los bordes de celdas sin contenido	show hide



2. CSS3

13. PSEUDO-CLASES Y PSEUDO-ELEMENTOS

Gracias a las **pseudo-clases** y los **pseudo-elementos de CSS** podemos realizar una selección más específica de los elementos a los que queremos aplicar un cierto estilo sin necesidad de crear una clase concreta.





2. CSS3

13.2 PSEUDO-CLASES PARA LOS ESTADOS DE UN ELEMENTO

Podemos utilizar diferentes pseudo-clases para definir las propiedades de ciertos elementos con diferentes estados. Uno de los usos más comunes es en los **enlaces <a>** aunque se pueden aplicar en otros elementos.

[Códigos Ejemplo](#)

Pseudo-clase	Descripción
:link	No visitado por el usuario
:visited	Visitado por el usuario
:hover	Modifica el estilo cuando un elemento apuntador pasa por encima
:active	Se activa cuando el usuario pulsa el elemento
:focus	Se activa cuando tiene el foco sobre el elemento



2. CSS3

13.3 PSEUDO-ELEMENTOS

A diferencia de las pseudo-clases, los **pseudo-elementos** no describen un estado sino que permiten **añadir estilos y contenido a una parte específica del documento**.

Pseudo-elemento	Descripción
::first-line	Primera línea de texto de un elemento
::first-letter	Primera letra de la primera línea de texto de un elemento
::before	Añade contenido al principio del documento
::after	Añade contenido al final del documento
::selection	Coge la porción del texto que se está seleccionando por el usuario

[Códigos Ejemplo](#)



2. CSS3

ACTIVIDAD

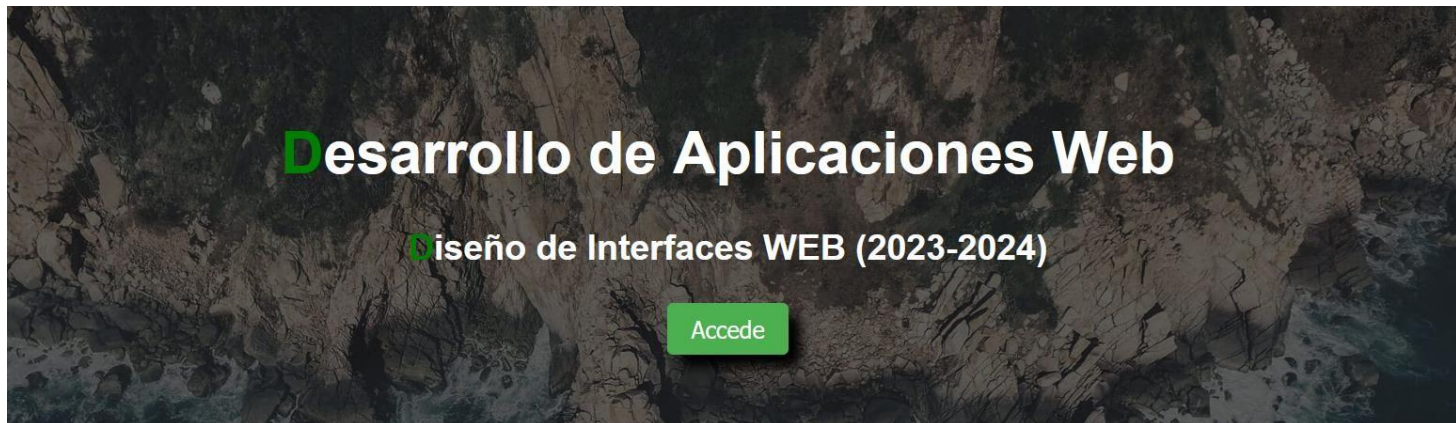


Uso de diferentes propiedades de fondo para crear una **imagen de tipo Hero** (**Imagen de gran tamaño** que suele aparecer al inicio de un sitio web y que sirve para destacar el mensaje, los productos o la esencia de una compañía).

Diseña la siguiente pantalla usando unidades relativas, para que se adapte cuando la pantalla sea más pequeña.

Al pasar el cursor por encima del botón debe cambiar de color.

Tanto el color de la primera letra como el año debe añadirse con pseudoelementos.



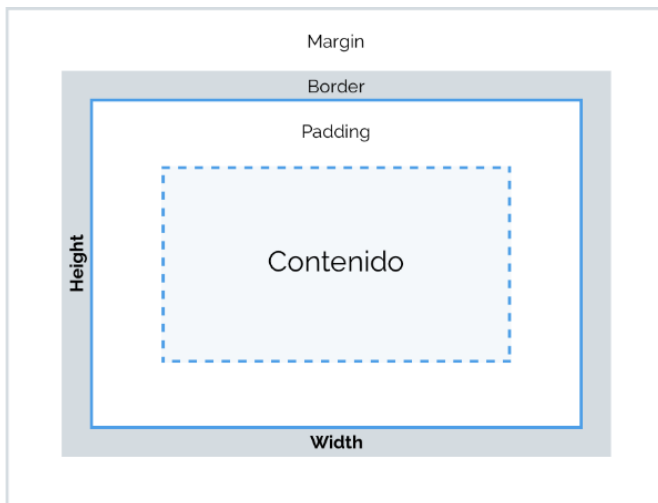


2. CSS3

14. MODELO DE CAJAS

Cualquier elemento incluido en un documento HTML dispone de una estructura tipo caja que se puede modificar usando las propiedades CSS. Las propiedades más importantes de las cajas o contenedores son las siguientes: **margin (margen externo), border (borde) y padding (margen interno).**

Las propiedades operan en el siguiente orden: **superior, derecha, inferior e izquierda.**



- **Un valor:** se aplica el estilo a los 4 lados.
- **Dos valores:** el primer valor se aplica a arriba y abajo, el segundo valor se aplica a derecha e izquierda.
- **Tres valores:** el primer valor se aplica a arriba, el segundo valor a derecha e izquierda y el tercer valor se aplica a abajo del elemento.
- **Cuatro valores:** el primer valor se aplica a arriba, el segundo valor se aplica a la derecha, el tercer valor se aplica a abajo y el cuarto valor se aplica a la izquierda.



2. CSS3

15. TAMAÑO DE LOS ELEMENTOS

[Códigos Ejemplo](#)

En CSS podemos controlar el tamaño de los elementos.

Nombre propiedad	Descripción	Valores
width	Establece el ancho del área de contenido de un elemento	Unidad de longitud (px, em, %, etc.), auto, initial, inherit
height	Establece el alto del área de contenido de un elemento	Unidad de longitud (px, em, %, etc.), auto, initial, inherit
max-width	Establece el ancho máximo que puede tener un elemento	Unidad de longitud (px, em, %, etc.), none, initial, inherit
min-width	Establece el ancho mínimo que debe tener un elemento	Unidad de longitud (px, em, %, etc.), 0, initial, inherit
max-height	Establecer el alto máximo que debe tener un elemento	Unidad de longitud (px, em, %, etc.), 0, initial, inherit
min-height	Establecer el alto mínimo que debe tener un elemento	Unidad de longitud (px, em, %, etc.), 0, initial, inherit



2. CSS3

16. POSICIÓN Y COMPORTAMIENTO DE CONTENEDORES EN CSS

Propiedad	Descripción	Valores
<u>display</u>	Comportamiento del contenedor	inline block inline-block none
<u>position</u>	Esquema de posicionamiento	static relative absolute fixed sticky
top right bottom left	Desplazamiento de la caja respecto al borde superior, derecho, inferior o izquierdo	longitud porcentaje auto
<u>float</u>	Posicionamiento flotante	left right none
<u>clear</u>	Control de cajas adyacentes a las float	none left right both
<u>z-index</u>	Nivel de la capa	auto número entero
<u>box-sizing</u>	Control de bordes y relleno en el comportamiento del contenedor	content-box border-box
<u>visibility</u>	Muestra u oculta un elemento ocupando el espacio	visible hidden
<u>overflow</u>	Controla cuando el contenido es más grande que su contenedor	visible hidden scroll auto



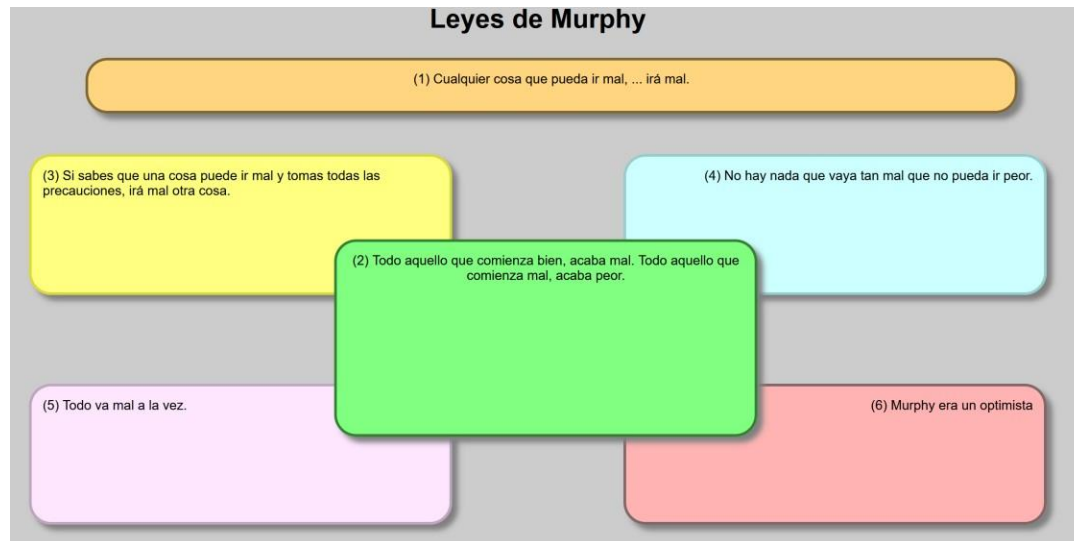
2. CSS3

ACTIVIDAD



Realiza la siguiente página web teniendo en cuenta estas cuestiones:

- Al pasar el cursor por el título debe desaparecer.
- Al pasar el cursor por encima de cada una de las cajas debe ubicarse por encima.
- Los números deben colocarse con pseudoelementos
- Las cajas deben ser flexibles y adaptarse al tamaño de la pantalla.





2. CSS3

17. CREAR MENU HORIZONTAL

Los menús son una parte muy importante dentro de nuestro diseño web.

15. Cómo crear un menú horizontal en HTML y CSS

Paso 1: Estructura HTML

Paso 2: Estilos CSS

Paso 3: Añadir colores y estilos personalizados al menú horizontal

Ejemplos de menú horizontal

[Ejemplo 1](#)

[Ejemplo 2](#)



2. CSS3

18. BUENAS PRÁCTICAS CSS

1. Organizar la estructura de arriba hacia abajo
2. Nombrar correctamente los selectores
3. Separar las palabras mediante guiones o mediante mayúsculas
4. Legibilidad
5. Combinar elementos
6. Utilizar selectores descendientes
7. Utilizar propiedades abreviadas
8. Utilizar nombres descriptivos en los selectores
9. Evitar utilizar como nombre de un selector una característica visual
10. Probar el diseño en los diferentes navegadores
11. Validar el código CSS
12. Agregar los prefijos de los navegadores en propiedades que no sean estables
13. No usar la regla !important



2. CSS3

19. HERRAMIENTAS ÚTILES

Herramientas útiles y tests de verificación CSS

1. CSS Validation Service, W3C
2. Browserling
3. Caniuse
4. Extensión Autoprefixer para Visual Studio Code
5. Repositorio de fuentes de Google Fonts
6. Repositorio de iconos de Font Awesome
7. Convertidores de fuentes online
8. Fuentes web online



2. HTML5

ACTIVIDAD



Realiza la siguiente página web. Los requisitos que tienen que cumplir están en la tarea de Moodle.

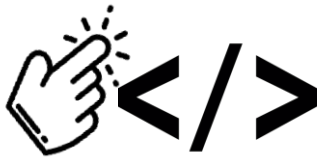
Inicio	Servicios	Nosotros	Contacto
<div> </div> <div> <h2>Beautiful Norway</h2> <p>Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.</p> <p>Norway has beautiful mountains, glaciers and stunning fjords. Oslo, the capital, is a city of green spaces and museums. Bergen, with colorful wooden houses, is the starting point for cruises to the dramatic Sognefjord. Norway is also known for fishing, hiking and skiing.</p> </div> <div> <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Vero magnam mollitia aperiam, animi sequi consequatur ad odio aliquam dolore quod dolor laboriosam numquam totam voluptatem rerum laudantium quibusdam repellendus voluptatum qui aliquid. Consequuntur voluptatibus nesciunt at reprehenderit tenetur suscipit accusantium aperiam, id assumenda dignissimos cumque sequi animi qui. In obcaecati earum ex sequi, eaque accusantium unde magnam doloremque minus</p> <p>beatae consequatur laborum quam, quia impedit doloribus odio quo perspiciatis natus deserunt? Culpa accusantium quasi dolorem at aperiam error, consequuntur iste perferendis corporis quidem laborum, voluptatibus porro aliquam sint commodi iusto ipsum rerum nam sed illum veniam deleniti. Atque quam consectetur dignissimos sint alias adipisci eaque labore nemo exercitationem dolor ea eius illum debitis recusandae asperiores cupiditate aperiam explicabo excepturi</p> <p>impedit, eligendi cumque temporibus reprehenderit quo aliquam. Illo, quisquam ex tempora vitae sed molestiae laborum? Possimus, rem! Ratione ex dolorem tenetur repellendus maiores minus veniam itaque officiis rerum enim delectus beatae, sunt ullam quam facilis obcaecati asperiores repudiandae, cumque saepe libero, ad laboriosam in! Neque est inventore quibusdam dolore temporibus quo illo iure, in voluptatem earum necessitatibus eligendi tempore totam quia a</p> <p>molestias, ipsam tenetur odit voluptate iusto illum expedita sed similique! Distinctio iure incidunt earum? Tempore molestiae est quod dolorem ipsam reprehenderit beatae iusto, autem quia ipsum possimus sunt molestias ex! Placeat dolorum atque, consequatur iusto amet eligendi ratione, rerum, facilis expedita obcaecati adipisci deleniti facere incidunt id fuga? At autem quos quaerat blanditiis deserunt aperiam.</p> </div>			



2. CSS3

21. CURSORES EN CSS

En CSS, los **cursores** se utilizan para cambiar la apariencia del cursor del ratón cuando se encuentra sobre un elemento interactivo. Esto permite personalizar la forma del cursor y proporcionar retroalimentación visual al usuario.



[Código Ejemplo](#)

```
HTML
1 <div class="container">
2   <div class="caja caja1">crosshair</div>
3   <div class="caja caja2">help</div>
4   <div class="caja caja3">move</div>
5   <div class="caja caja4">pointer</div>
6   <div class="caja caja5">progress</div>
7   <div class="caja caja6">text</div>

CSS
15 .caja1{
16   background-color: #C0392B;
17   cursor: crosshair;
18 }
19 .caja2{
20   background-color: #9B59B6;
21   cursor: help;
22 }
```

crosshair

help

move

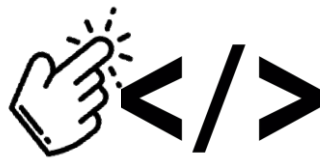
pointer



2. CSS3

22. BOTONES CSS

En CSS, podemos crear nuestros **propios botones**:



[Código Ejemplo](#)

The screenshot shows a code editor with two panels: HTML and CSS. The HTML panel contains five button elements with different classes. The CSS panel shows the corresponding styles for these classes. Below the code editor, a visual preview of the buttons is shown.

```
HTML
1 <button class="btn primary">Primary</button>
2 <button class="btn
   secondary">Secondary</button>
3 <button class="btn success">Success</button>
4 <button class="btn warning">Warning</button>
5 <button class="btn danger">Danger</button>

CSS
1 .btn {
2   borde
3   color
4   paddi
5   curso
6   borde
7 }
8
9 .primar
10 .primar
11
12 .second
   color:
13 .second
14
```

Primary Secondary Success Warning Danger



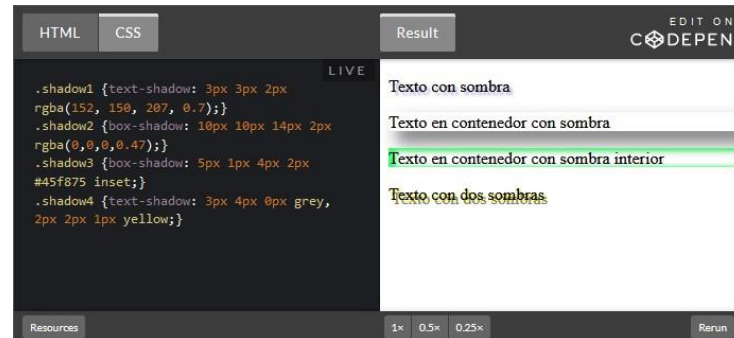
2. CSS3

23. SOMBRAS

Gracias a las propiedades de CSS3 «**text-shadow**» y «**box-shadow**» podemos aplicar sombras en los textos y en los contenedores para así enriquecer nuestras interfaces.

El formato es prácticamente el mismo para las dos propiedades.

- **text-shadow:** posx posy desenfoque color | none;
- **box-shadow:** posx posy desenfoque tamaño color inset | none;



Generador de sombras en textos y otros efectos: <https://cssgenerator.org/>



2. CSS3

24. GRADIENTES

El gradiente lineal se define mediante la función `linear-gradient()` y permite crear fondos degradados en una dirección específica. El formato es el siguiente:

```
background: linear-gradient(dirección|ángulo, color, color, color...);  
background: linear-gradient(dirección|ángulo, color posición, color  
posición, color posición...);  
background: linear-gradient(color, color, color...);
```

[Código Ejemplo](#)

```
background: radial-gradient(forma, color, color, color...);  
background: radial-gradient(forma posición, color, color, color...);  
background: radial-gradient(forma, color posición, color posición, color  
posición...);  
background: radial-gradient(color, color, color...);
```

[Código Ejemplo](#)

Generador de sombras en textos y otros efectos: <https://cssgenerator.org/>



2. CSS3

25. FILTROS EN IMAGENES

Los **filtros CSS** permiten aplicar efectos visuales a las imágenes. Para aplicar un filtro CSS a una imagen se utiliza la propiedad **filter**. Veamos un ejemplo de cómo aplicar un filtro de desenfoque a una imagen:

HTML:

```
1. 
```

CSS:

```
1. .filtro-desenfoque {  
2.   filter: blur(5px);  
3. }
```

[Código Ejemplo](#)

Generador de sombras en textos y otros efectos: <https://cssgenerator.org/>



2. CSS3

26. TRANSICIONES

Las **transiciones en CSS** nos permiten aplicar un cambio de **estilo gradual a los elementos del documento HTML**. Además, nos ofrecen la ventaja de poder especificar el tiempo para que se produzca la transformación entre estilos, de esta forma podríamos utilizarlas para dar un **efecto de animación**.

Formato:

transition: [propiedad a modificar] [Duración] [Tipo de animación] [Retardo];

Ejemplo:

```
.caja1{
  background-color: #C0392B;
  transition: background-color 1s linear;
}
.caja1:hover{
  background-color: #3F51B5;
}
```

[Código Ejemplo](#)



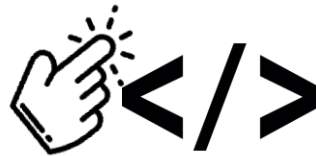
2. CSS3

27. TRANSFORMACIONES

Las transformaciones CSS nos permiten **rotar, torcer, escalar o desplazar** los elementos de nuestra página web. Este tipo de propiedades de CSS3 son muy interesantes para convertir el lenguaje de hojas de estilo en un sistema capaz de realizar **todo tipo de efectos visuales**.

Ejemplos:

- **transform:** scale(0.5); /* Escala el elemento a la mitad */
- **transform:** translate(10px); /* Traslada el elemento 10px hacia la derecha */
- **transform:** rotate(45deg); /* Rota el elemento 45 grados */
- **transform:** skew(45deg); /* Distorsiona el elemento 45 grados en el eje x */



[Código Ejemplo](#)



2. CSS3

28. ANIMACIONES

En este apartado veremos cómo **encadenar diferentes animaciones** utilizando la propiedad `animation` y sus subpropiedades. Para ello, definiremos la propiedad **`animation`** sobre nuestros selectores y después definiremos nuestra secuencia de animación mediante `@keyframes`.

The animation-iteration-count Property

Play the animation two times:



[Código Ejemplo](#)



2. CSS3

29. TOOLTIP

El contenido de **un tooltip** puede ser texto, imágenes u otros elementos HTML

The screenshot shows a CodePen editor with the following code:

```
HTML
1 <div class="tooltip-container">
2   <span class="tooltip-text">Mensaje del
   tooltip</span>
3   Este elemento tiene un tooltip
4 </div>
5

CSS
1 .tooltip-container {
2   position: relative;
3   display: inline-block;
4 }
5
6 .tooltip-text {
7   visibility: hidden;
8   width: 120px;
9   background-color: #000;
10  color: #fff;
11  text-align: center;
12  border-radius: 6px;
13  padding: 5px;
14  position: absolute;
15  z-index: 1;
```

The live preview at the bottom shows the text "Este elemento tiene un tooltip" with a black tooltip box containing the text "Mensaje del tooltip" appearing below it.





2. CSS3

30. PROPIEDADES PERSONALIZADAS

Las **CSS Custom Properties** o propiedades personalizadas (también llamadas variables CSS) nos permiten definir nombres para valores determinados de las propiedades. De esta forma, se puede utilizar ese nombre como si **fuese una variable**.

Para definir una propiedad personalizada pondremos **dos guiones -- delante del nombre** que queramos utilizar. Para definir la propiedad personalizada para **todo el documento HTML** la incluiremos en el pseudoelemento **:root**.

```
1. :root {  
2.   --principal-color: black; /* Valor personalizado */  
3.   --font: "Arial";  
4. }
```

Para utilizar una **propiedad personalizada** hay que insertar su nombre dentro de la **expresión var()**:

```
1. .clase {  
2.   background-color: var(--principal-color);  
3.   font-family: var(--font);  
4. }
```




2. HTML5

ACTIVIDAD



Realiza la siguiente página web. Los requisitos que tienen que cumplir están en la tarea de Moodle.





2. CSS3

31. TIPOS DE DISEÑOS WEB

A la hora de crear una **interfaz web** existen diferentes tipos de diseño según su adaptabilidad a los distintos dispositivos

- **Diseño Fijo:** Uso de unidades **fijas**. No cumple normas de diseño responsive.
- **Diseño elástico:** Se definen las dimensiones con unidades "**rem**". El diseño se adapta cuando **cambia el tamaño del texto**, pero no en función de la ventana del navegador.
- **Diseño líquido o fluido:** El diseño líquido se adapta a la ventana del dispositivo, normalmente definiendo los tamaños **mediante porcentajes**. Sin embargo, este método perjudica la experiencia de usuario, porque **varía constantemente** según el tamaño del dispositivo.



2. CSS3

32. TIPOS DE DISEÑOS WEB

A la hora de crear una interfaz web existen diferentes tipos de diseño según su adaptabilidad a los distintos dispositivos

- **Diseño web responsive o adaptable:** Basado en el uso de **media queries**. Se definen **puntos de ruptura** que nos permiten cambiar el diseño de la página.
- **Diseño dentro de unos mínimos y máximos:** Consiste en utilizar las propiedades CSS «**min-width**» y «**max-width**» para que las anchuras de los bloques puedan adaptarse dentro de unos mínimos y máximos.

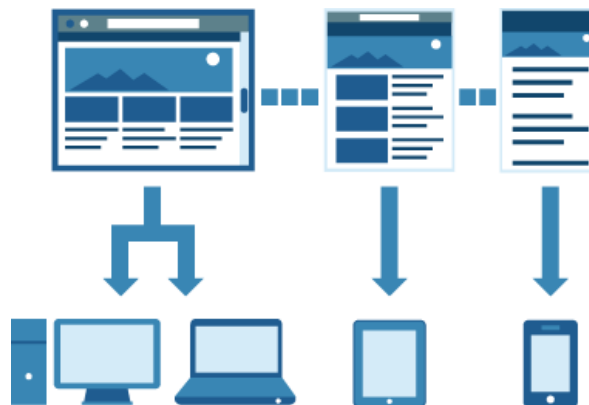


2. CSS3

33. ¿QUÉ TIPO DE DISEÑO WEB UTILIZAR?

Crear un sitio web adaptable exige, en muchos casos, **combinar algunas de las técnicas anteriores**. Lo más habitual es declarar **más de una media query** para adaptar el diseño a múltiples dispositivos (Móvil, Tablet, Escritorio)

Lo ideal es utilizar un **diseño web responsivo**, que puede mezclar diseño **líquido o fluido (porcentajes)** según el contenedor. De este modo ofreceremos la mejor experiencia de usuario y podremos considerar las distintas medidas de los dispositivos utilizando tan solo **HTML y CSS**.



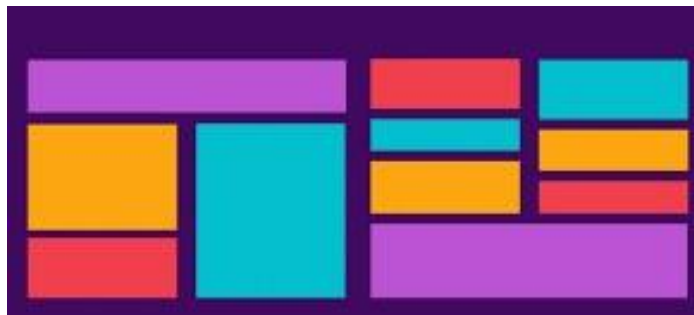


2. CSS3

34. TIPOS DE DISEÑO WEB SEGÚN LAS PROPIEDADES CSS UTILIZADAS

Según las propiedades CSS tenemos las siguientes opciones:

- **Diseño convencional con bloques y floats:** Uso de float, position y display. Todavía se utiliza pero es difícil de mantener.
- **Diseño con Flexbox:** Flexbox es un modelo de **diseño unidimensional** que permite organizar los elementos en una sola dirección (fila o columna).
- **Diseño con CSS Grid:** CSS Grid es un modelo de **diseño bidimensional** que permite organizar los elementos en filas y columnas, creando así diseños más complejos y estructurados.





2. CSS3

35. MEDIA QUERIES

Las **media queries** se introdujeron en CSS3 para dar respuesta a las necesidades del **diseño web responsive**. Mediante ellas podemos definir **estilos condicionales**, aplicables únicamente en determinadas situaciones. El uso más extendido de las media queries es para establecer estilos diferentes para cada **ancho de pantalla**.

- **Media query indicando el ancho mínimo de pantalla:**

```
1. @media (min-width: 768px) {  
2.     /* Estilos para pantallas con un ancho mínimo de 768px */  
3. }
```

- **Media query indicando el ancho máximo de pantalla:**

```
1. @media (max-width: 767px) {  
2.     /* Estilos para pantallas con un ancho máximo de 768px */  
3. }
```



2. CSS3

35. MEDIA QUERIES

Es posible combinar **múltiples condiciones en una media query** para aplicar estilos solo cuando se cumplan todas las condiciones. Todos los estilos que se incluyan en el interior de la siguiente media query serán utilizados en pantallas que tengan un ancho entre 768px y 1024px (puntos de ruptura de 768px y 1024px)

```
1. @media (min-width: 768px) and (max-width: 1024px) {  
2.     /* Estilos para pantallas con un ancho entre 768px y  
   1024px */  
3. }
```



2. CSS3

36. MEDIA QUERIE PARA MOBILE FIRST

En un enfoque **"Mobile First"**, los estilos base están diseñados para dispositivos móviles y luego se aplican ajustes mediante media queries para tabletas y escritorios.

Ejemplo

Prácticas comunes

```
1.  /* Estilos base para dispositivos móviles */
2.  body {
3.      background-color: red;
4.  }
5.  p:after {
6.      color: white;
7.      content: "Móvil";
8.  }
9.  /* Media query para tabletas */
10. @media (min-width: 600px) {
11.     body {
12.         background-color: green;
13.     }
14.     p:after {
15.         color: white;
16.         content: "Tablet";
17.     }
18. }
19. /* Media query para desktop */
20. @media (min-width: 1024px) {
21.     body {
22.         background-color: blue;
23.     }
24.     p:after {
25.         color: white;
26.         content: "Escritorio";
27.     }
28. }
```




2. CSS3

37. FLEXBOX

Flexbox, también conocido como modelo de caja flexible, es un **diseño unidimensional** que permite crear un diseño más eficiente y predecible en situaciones donde el tamaño de los elementos es desconocido o dinámico. Los elementos se organizan usando **contenedores flexibles**.



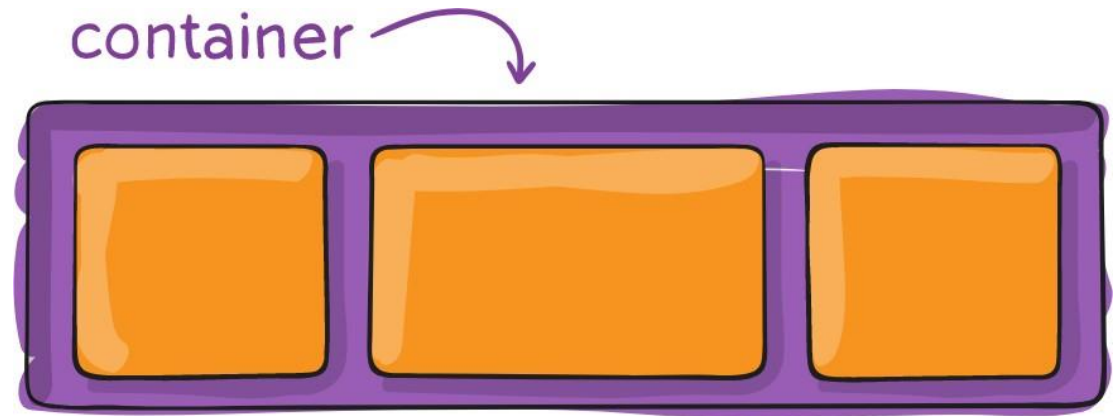


2. CSS3

37. PROPIEDADES DEL CONTAINER

Las propiedades del **flex container** son:

- **display**
- **flex-direction**
- **flex-wrap**
- **flex-flow**
- **justify-content**
- **align-items**
- **align-content**





2. CSS3

37.1 display:flex | inline-flex

La propiedad display en CSS determina cómo se muestra un elemento en el diseño de la página. En el contexto de flexbox, estos dos valores, **flex** e **inline-flex**, se utilizan para establecer un contenedor como un contenedor flexible.

- **flex:** Cuando aplicas display: flex a un contenedor, este se convierte en un contenedor flexible que distribuye su contenido de **manera flexible** a lo largo del eje principal.

```
1. .contenedor-flex {  
2.   display: flex;  
3. }
```

- **inline-flex:** Este valor es similar a display: flex, pero **el contenedor flexible** se comporta como un elemento en línea en lugar de un bloque. Esto significa que ocupará solo el ancho necesario y permitirá que otros elementos en línea coexistan en la misma línea.

```
1. .contenedor-inline-flex {  
2.   display: inline-flex;  
3. }
```

Ejemplo



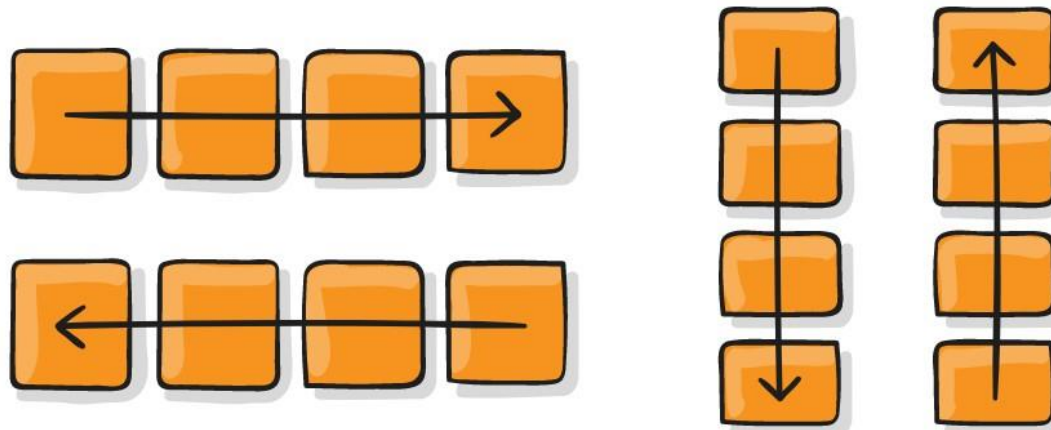
2. CSS3

37.2 flex-direction: row | row-reverse | column | column-reverse

La propiedad **flex-direction** determina la dirección en la que los elementos flexibles se colocan dentro de un **contenedor flex**.

La propiedad **flex-direction** acepta cuatro valores:

- **row (valor por defecto):** Coloca los elementos en una fila horizontal. **row-reverse:** Coloca los elementos en una fila horizontal, pero en orden inverso al valor por defecto.
- **column:** Coloca los elementos en una columna vertical.
- **column-reverse:** Coloca los elementos en una columna vertical, pero en orden inverso al valor por defecto.



[Ejemplo](#)

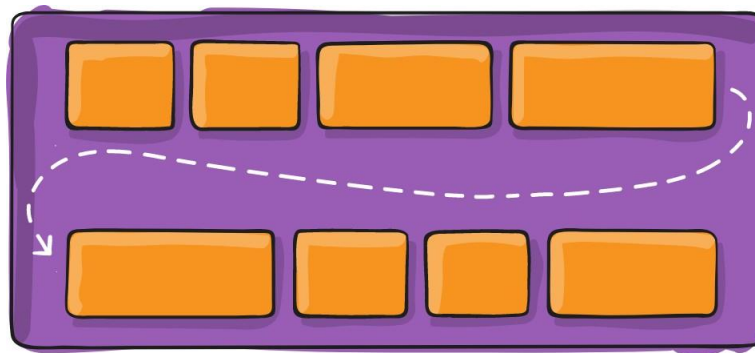


2. CSS3

37.3 flex-wrap: nowrap | wrap | wrap-reverse

Podemos indicar qué **elementos flexibles se deben trasladar** cuando **no hay suficiente espacio** en el contenedor mediante la propiedad flex-wrap. Sus valores son los siguientes:

- **nowrap:** los elementos no pasan a la siguiente fila y se reduce su anchura para mostrarlos. Este es el valor por defecto.
- **wrap:** los elementos pasan a la siguiente fila y conservan su anchura.
- **wrap-reverse:** los elementos pasan a la siguiente fila, pero en sentido inverso al de su declaración.



Ejemplo



2. CSS3

37.4 justify-content: flex-start | flex-end | center | space-between | space

La propiedad **justify-content** es la que define la justificación horizontal de los elementos hijos de un contenedor flexible. Esta propiedad acepta cinco valores:

- **flex-start:** posiciona los elementos a la izquierda del contenedor. Valor por defecto.
- **flex-end:** posiciona los elementos a la derecha del contenedor.
- **center:** centra los elementos en el contenedor
- **space-between:** añade un espacio idéntico entre los elementos. El primer elemento está alineado a la izquierda del contenedor, y el último, a la derecha.
- **space-around:** espacia de forma regular los elementos que no están alineados a la izquierda y a la derecha del contenedor.

flex-start



flex-end



center



space-between



space-around



Ejemplo



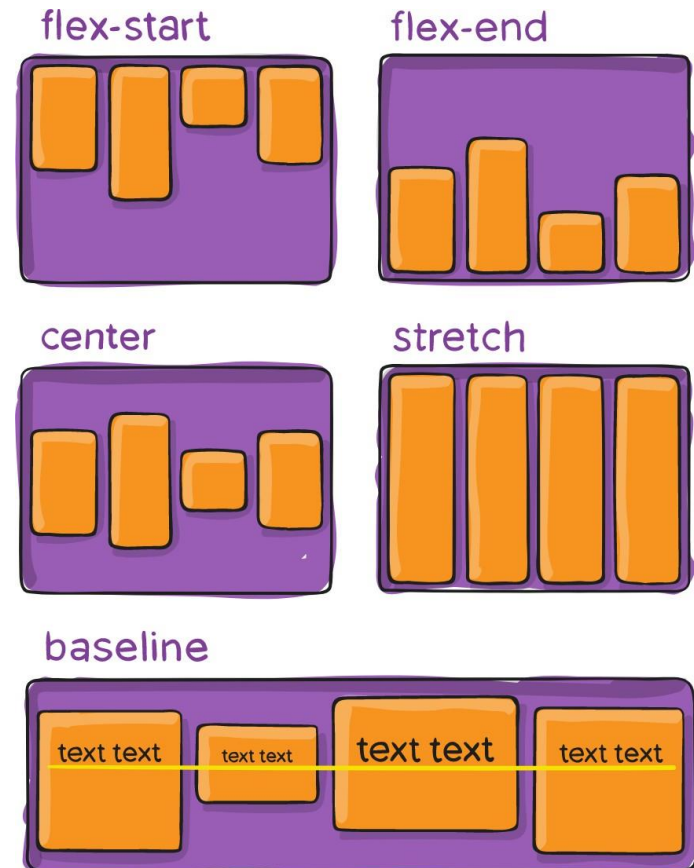
2. CSS3

37.5 align-items: stretch | flex-start | flex-end | center | baseline

La propiedad **align-items** actúa sobre el eje vertical. Para utilizar esta propiedad primero se debe definir la propiedad **flex-direction: row**

La propiedad **align-items** acepta cinco valores:

- **stretch**: los elementos se amplían verticalmente para ocupar toda la altura disponible en el contenedor. Este es el valor por defecto.
- **flex-start**: los elementos se colocan en la parte superior del contenedor.
- **flex-end**: los elementos se colocan en la parte inferior del contenedor.
- **center**: los elementos se colocan en el centro del contenedor.
- **baseline**: los elementos se alinean sobre la línea de base del texto.



Ejemplo

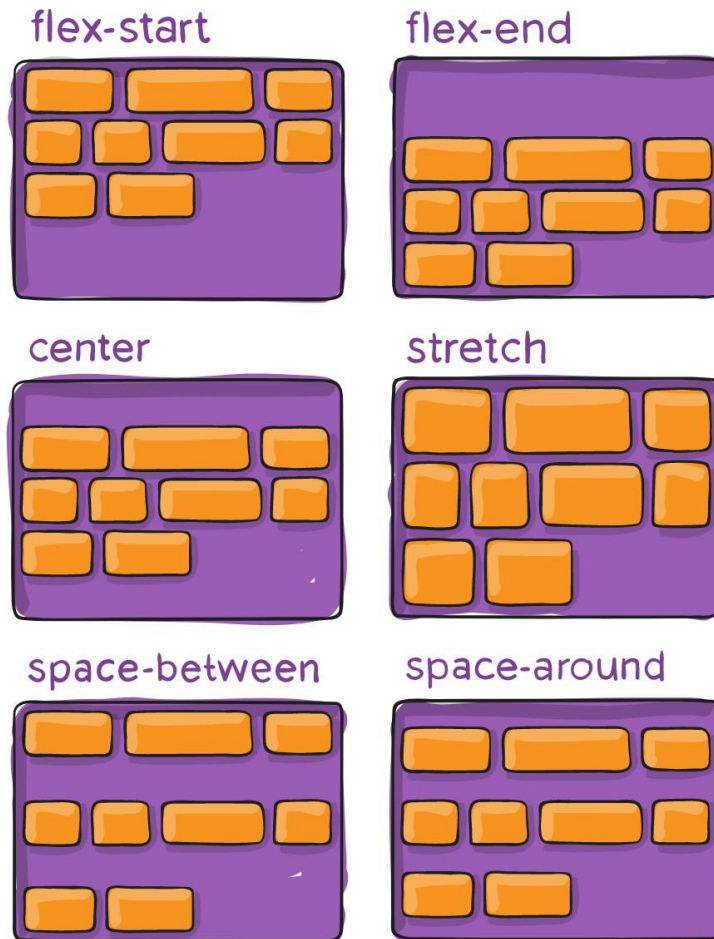


2. CSS3

37.6 align-content: flex-start | flex-end | center | space-between | space-around | stretch

Cuando tenemos **varias líneas** con esta propiedad se organizan las líneas dentro de un **flex container** cuando hay espacio adicional en **el eje transversal**, similar a cómo justify-content alinea ítems individuales dentro del eje principal.

Importante: Esta propiedad no tiene efecto **cuando sólo hay una línea de flex ítems** en el contenedor.



[Ejemplo](#)



2. CSS3

37.7 gap

La propiedad gap define el ancho del hueco entre filas y columnas en flexbox.

The gap Property

Use the *gap* property to specify the size of the gap between the rows and the columns.

This grid has a 50px gap between the rows and between the columns:

1	2	3
4	5	6
7	8	



2. CSS3

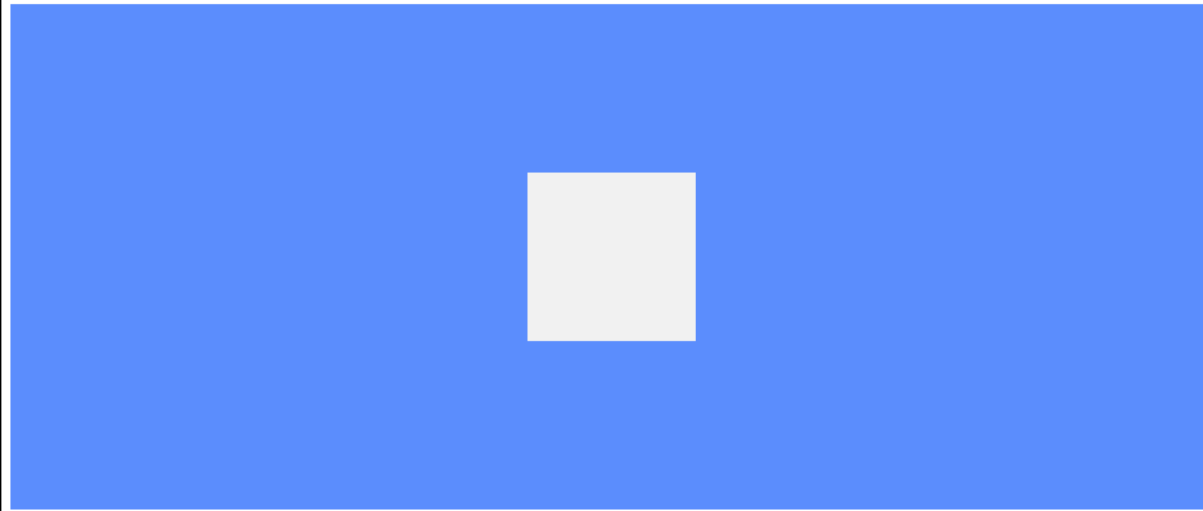
37.7 El centro perfecto

Cuando queremos poner un **elemento en el centro** de un container tanto en altura como en anchura podemos usar las dos propiedades:

- **align-items: center**
- **align-content: center**

Perfect Centering

A flex container with both the justify-content and the align-items properties set to *center* will align the item(s) in the center (in both axis).



[Ejemplo](#)

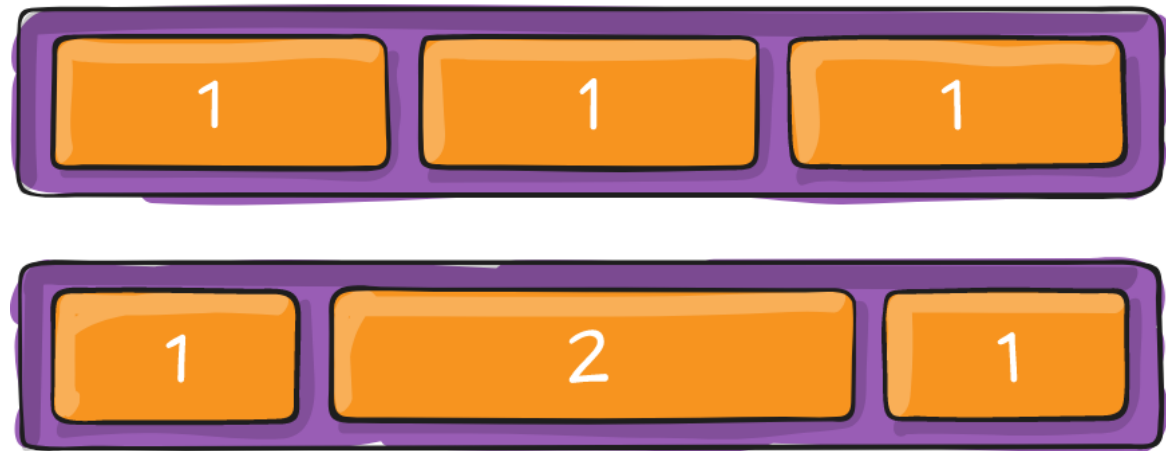


2. CSS3

37.8 Propiedades de los flex-ítems

Los elementos incluidos en un flex – container se convierten en flex – items con las siguientes propiedades:

- **order**
- **flex-grow**
- **flex-shrink**
- **flex-basis**
- **flex**
- **align-self**

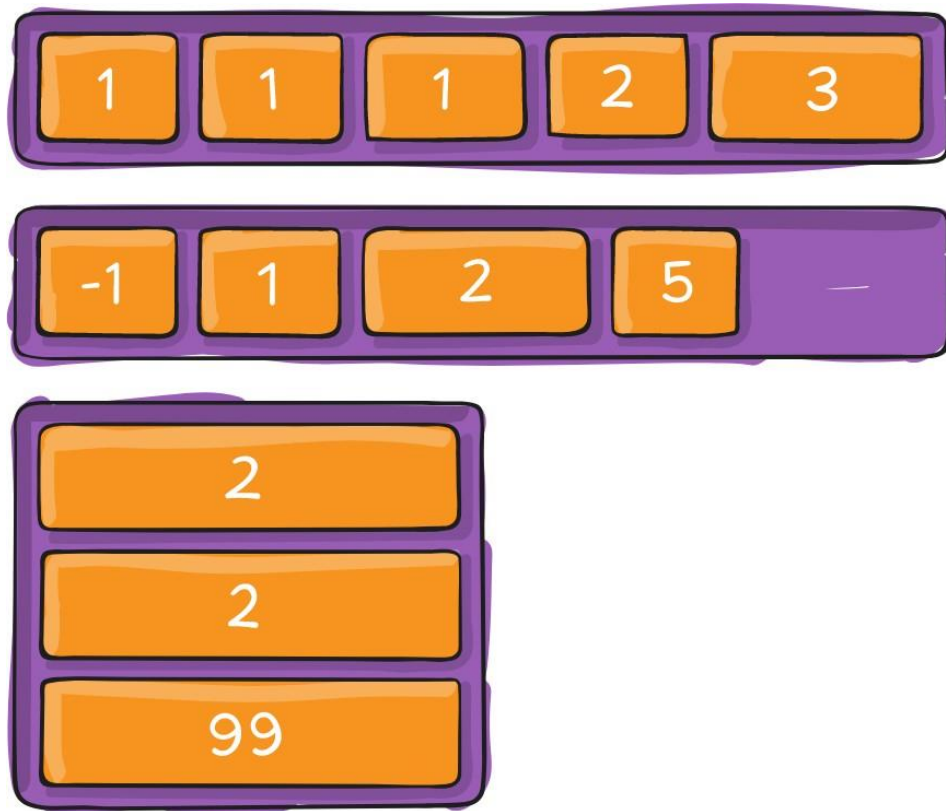




2. CSS3

37.9 Propiedades de los flex-ítems: order

- **order:** Cambia el orden en el que aparecen los elementos flexibles.



[Ejemplo](#)



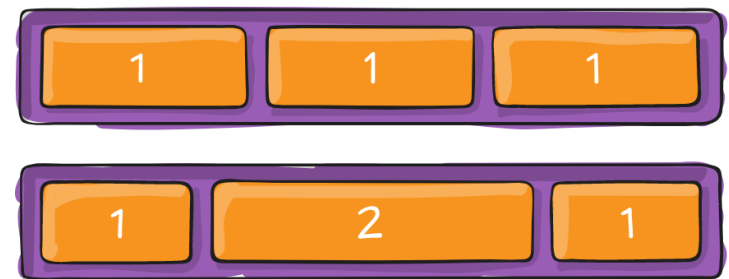
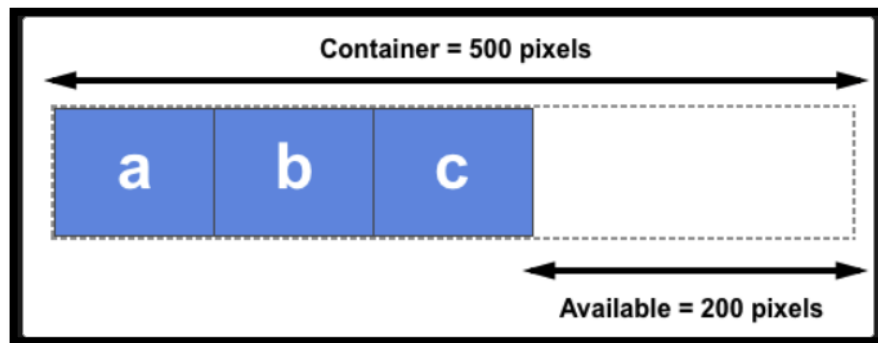
2. CSS3

37.10 Propiedades de los flex-ítems: flex-grow

- **flex-grow:** define cómo un elemento debe crecer en relación con el resto de los elementos flexibles en el contenedor Flexbox. **Valor por defecto: 0. No crece.**

Importante: El espacio a repartir será el **espacio inicial libre**.

Si todos los ítems tienen **flex-grow establecido en 1**, el espacio remanente en el contenedor se distribuirá de manera uniforme entre todos. Si uno de los ítems tiene el valor 2, ocupará el doble de espacio en el contenedor que los demás (o al menos intentará hacerlo).



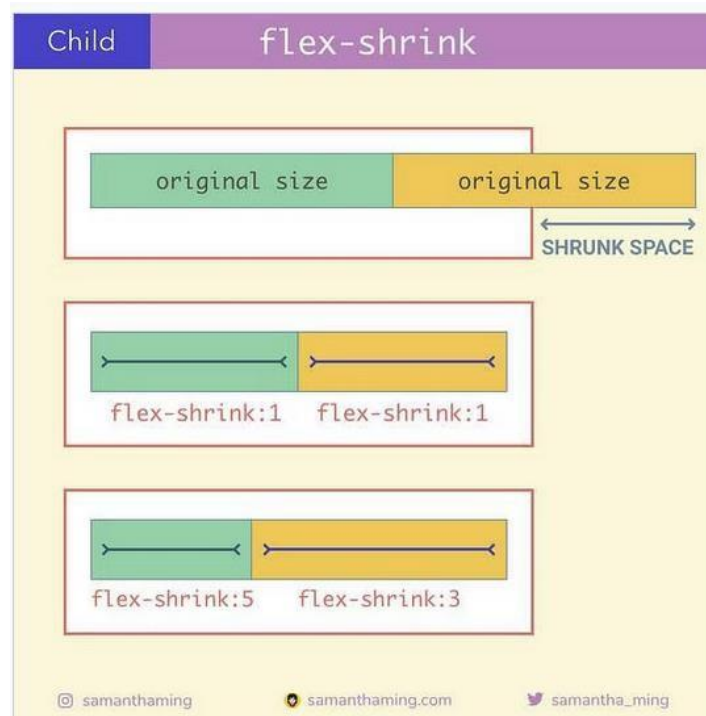
Ejemplo



2. CSS3

37.11 Propiedades de los flex-ítems: flex-shrink

- **flex-shrink:** define cómo los elementos se encogen en relación con los demás cuando no hay suficiente espacio en el contenedor. **Valor por defecto: 1. Encoge en la misma proporción que el resto.**





2. CSS3

37.12 Propiedades de los flex-ítems: flex-basis

- **flex-basis:** establece el **tamaño inicial** de un elemento **antes de que se distribuya** el espacio restante

The diagram illustrates the `flex-basis` property with three examples of flex containers. Each container is represented by a yellow rectangle with a red border. Above the containers is a header bar with a blue 'Child' label and a purple 'flex-basis' label.

- Example 1:** Two flex items, one green and one yellow, both labeled `flex-basis: auto`. They are side-by-side, filling the container.
- Example 2:** Two flex items, one green and one yellow, both labeled `flex-basis: 50%`. They are side-by-side, each taking up 50% of the container's width.
- Example 3:** Two flex items, one green and one yellow, labeled `flex-basis: auto` and `flex-basis: 500px` respectively. The green item is smaller, and the yellow item is larger, taking up 500px of the container's width.

At the bottom of the diagram, there are three social media links: [@samanthaming](#), [samanthaming.com](#), and [samantha_ming](#).

Ejemplo



2. CSS3

37.13 Propiedades de los flex-ítems: flex (Shorthand)

- **flex:** Esta es la **propiedad shorthand** para **flex-grow**, **flex-shrink** y **flex-basis**, combinadas. El segundo y tercer parámetros (flex-shrink y flex-basis) son opcionales.

El valor por defecto es **0 1 auto**:

flex-grow: no crece;

flex-shrink: encoge en la misma proporción que el resto.

flex-basis: se ajusta al contenido;

Se recomienda que utilices esta propiedad **shorthand en lugar de definir cada una de las propiedades por separado**. Shorthand establece los demás valores de forma inteligente.

Ejemplo



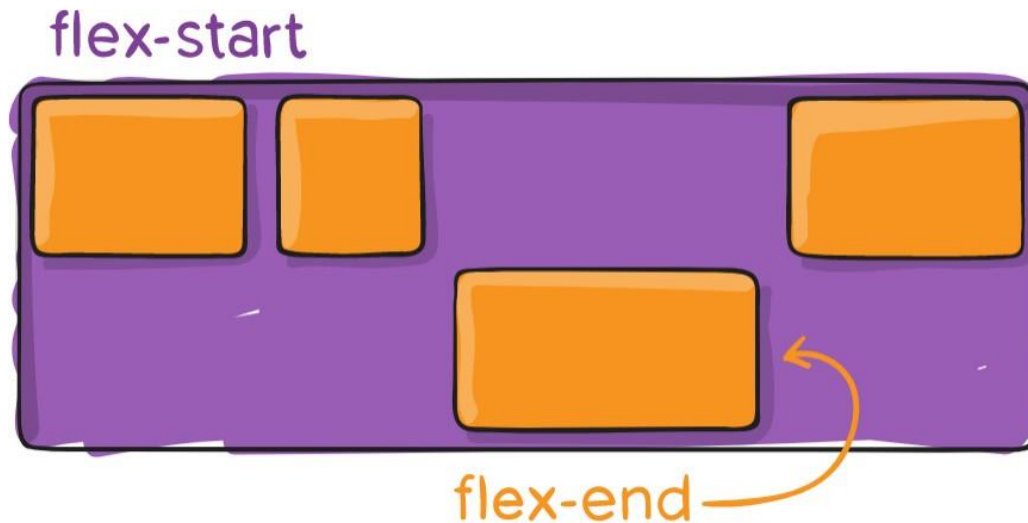
2. CSS3

37.14 Propiedades de los flex-ítems: align-self

Permite que la alineación estándar (o lo que esté definido por align-items) se **sobrescriba para ítems individuales**.

Los posibles valores.

```
.item {  
    align-self: auto | flex-start | flex-end | center | baseline | stretch;  
}
```



Ejemplo



2. CSS3

37.15 PRACTICA FLEXBOX

FLEXBOX FROGGY

◀ Nivel 1 de 24 ▶

Bienvenido a Flexbox Froggy, un juego donde ayudarás a Froggy y a sus amigos escribiendo código CSS. Guía a esta rana hacia la hoja de lirio en la derecha, usando la propiedad `justify-content`, la cual alinea elementos horizontalmente y acepta los siguientes valores:

- `flex-start`: Alinea elementos al lado izquierdo del contenedor.
- `flex-end`: Alinea elementos al lado derecho del contenedor.
- `center`: Alinea elementos en el centro del contenedor.
- `space-between`: Muestra elementos con la misma distancia entre ellos.
- `space-around`: Muestra elementos con la misma separación alrededor de ellos.

Por ejemplo, `justify-content: flex-end;` moverá la rana a la derecha.



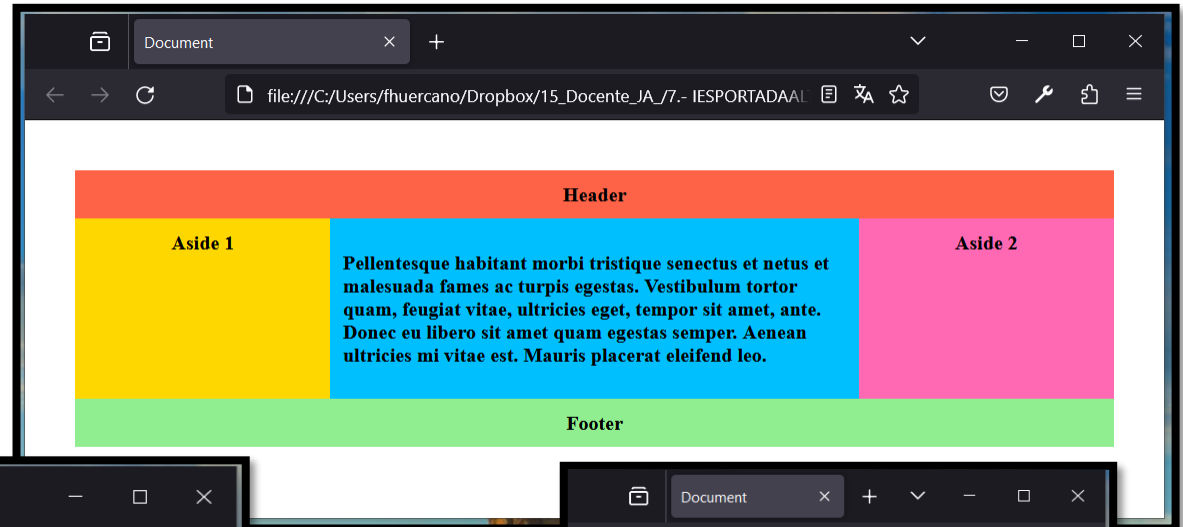
<https://flexboxfroggy.com/#es>



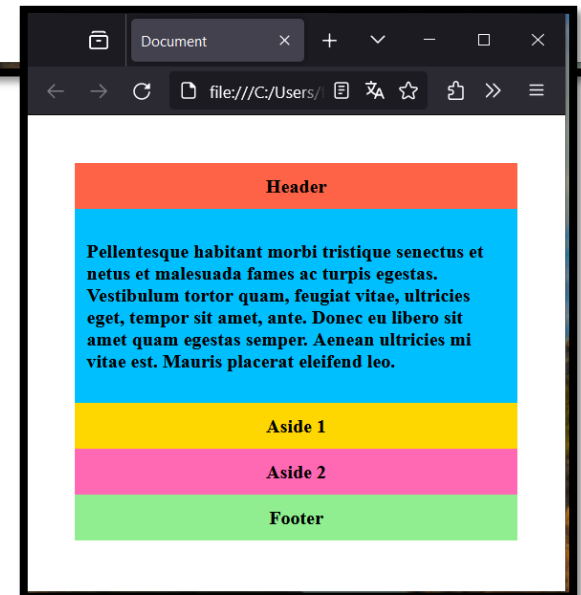
2. HTML5

ACTIVIDAD

Crea el siguiente diseño responsive para móvil, Tablet y escritorio usando la técnica "mobile first"



Escritorio



2. HTML5

ACTIVIDAD Tablet

Móvil





2. HTML5

ACTIVIDAD

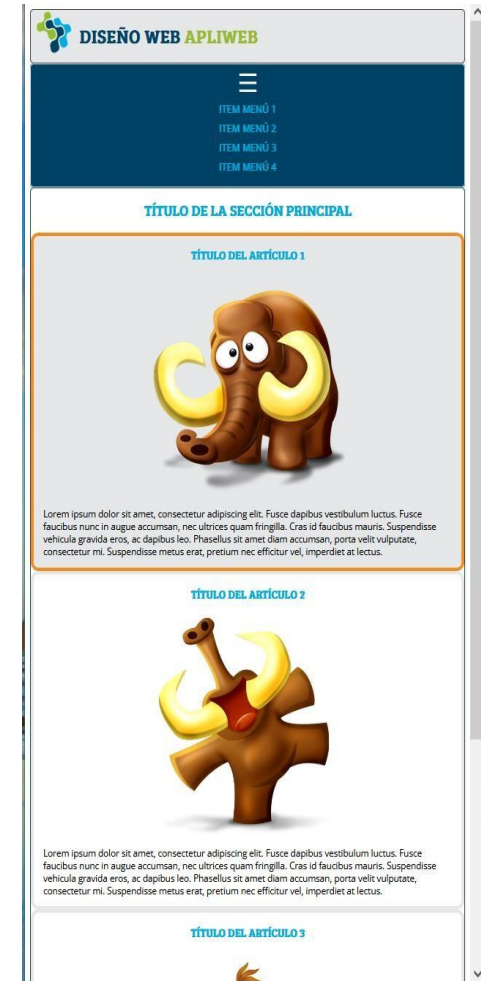
Crea el siguiente diseño responsive para móvil, Tablet y escritorio usando la técnica **"mobile first"**.



Tablet



Escritorio



Móvil