



Laboratorio I - 2024



Clase Teórica 02

Docente: Myriam Ruiz

Licenciada en Informática
Profesora en Computación y Matemática
Programadora Universitaria



Metatags (Meta etiquetas)

- Los Metadatos son datos que describen datos, y HTML tiene una forma "oficial" de agregar metadatos a un documento... en la etiqueta `<meta>`

Metatags

- `<meta name="" content="">`
- Varios `<meta ...>` permiten que los buscadores encuentren más fácilmente nuestro sitio (Rankea mejor)

Metatags

- Dos meta name más usados
 - `<meta name="description" content="Explicamos aquí de que trata el sitio">`
 - `<meta name="author" content="Nombre del alumno">`
- Existen más atributos para las etiquetas meta, que por el momento no veremos...

Metatags



```
<meta name="description" content="Sitio web oficial de la UNT, Universidad Nacional de Tucumán - Tucumán - Argentina" />
```

Más tipos de Enlaces

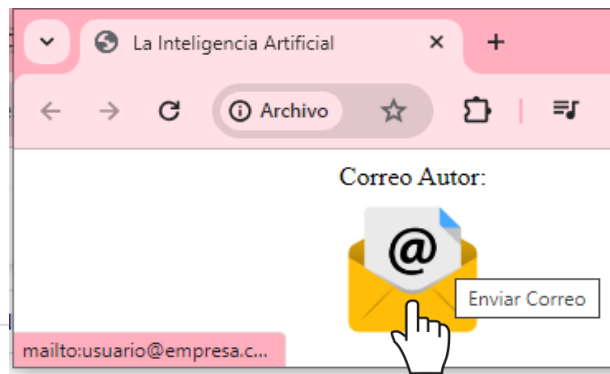
- A correo:
- `usuario@empresa.com`
- A Números Telefónicos:
- `381 4364093`
- A páginas externas al sitio
- `Facet Virtual`
-

Forzar descarga de archivos

- Utilizar el atributo download
- `<a download title="Teoría 01" href="../../pdf/clase01.pdf" >Clase 01`
- Se puede cambiar el nombre el nombre del archivo a descargar escribiendo `download="nuevoNombre"`

Enlace gráfico

Consiste en un enlace que usa una imagen como medio para que el usuario interactúe con este.



```
<footer>
  <p>Correo Autor:
    <a href="mailto:usuario@empresa.com">
      <figure>
        
      </figure>
    </a>
  </p>
</footer>
```


Class e Id

- Cuando existe más de una etiqueta que queremos personalizar de manera diferente desde CSS, por ejemplo, más de un article, distintas filas de una tabla, más de un menú, se requiere de una forma que permita individualizar los elementos por algo que les sea propio, para esto se utiliza **class** o **id**.

Diferencia entre Class e Id

- **Id:**
 - Es para un ÚNICO elemento en la página, un Menú Principal (suponiendo que haya varios menús), un encabezado, etc.
- **Class:**
 - Se puede REUTILIZAR en varios elementos

Diferencia entre Class e Id

- En HTML

Class	Id
<code><td class="fila1">Texto</td></code>	<code><h1 id="titulo">Título</h1></code>

- En CSS

Class	Id
<code>.fila1 { Background-color:#777777; }</code>	<code>#titulo { Font-size: 3em; }</code>

Punto

Hashtag o Numeral

Uso de Id

- Ej.: supongamos que tenemos dos zonas con menú de navegación, pero solo uno será el menú principal. Usaremos un id para ese.

```
<nav id="menu">
  <ul>
    <li><a href="html/nosotros.html">Nosotros</a></li>
    <li><a href="html/eventos.html">Eventos</a></li>
    <li><a href="html/contacto.html">Contacto</a></li>
  </ul>
</nav>
```

Escribir en el
archivo css:

```
#menu{
  background-color: navy;
}
```

Recomendaciones para nombres

1. Usar nombres de clases o id descriptivos y no abreviar excesivamente:

elegir `.productos` en vez de `.tabla`

`.usuario` en vez de `.u`

2. Escribir nombres compuestos separados con guión medio.

Ej: `.menu-principal`, `.imagen-destacada`

Enlaces a parte de una página

- Mediante el uso de id, se puede enlazar a una parte específica dentro de una página

Enlaces a parte de una página

- En el origen del enlace:

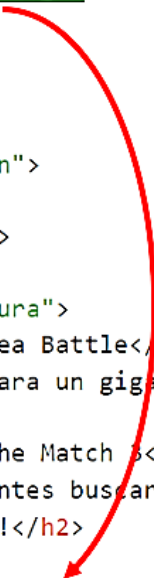
`Enlace`

- En el destino del enlace:

`<h2 id="nombreId">Argentina será sede de la Copa América 2020</h2>`

Enlaces a parte de una página

```
<nav>
  <ul>
    <li><a href="#accion"></a></li>
    <li><a href="#arte"></a></li>
    <li><a href="#aventura"></a></li>
    <li><a href="#estrategia"></a></li>
  </ul>
</nav>
<main>
  <section>
    <article id="accion">
    </article>
    <article id="arte">
    </article>
    <article id="aventura">
      <h2>Colonial Sea Battle</h2>
      <p>Preparate para un gigantesca batalla en el mar</p>
      <hr>
      <h2>Pirates! The Match </h2>
      <p>Estos valientes buscarenos zarparon a una aventura épica</p>
      <h2>Sea Bubble!</h2>
    </article>
    <article id="estrategia">
      <h2>Platas vs. Zombies</h2>
    </article>
  </section>
</main>
```



Enlaces a parte de una página

- También se puede acceder a una parte específica de una página diferente:
- `Enlace`

Enlaces que aportan Accesibilidad

- Saltar a Contenido (o bloque de contenido)
- Saltar a Navegación (Al menú)



https://www.once.es

Salto a contenido

Salto a navegación

Español

English

Catalá

Galego

Euskera

Buscar...



Servicios sociales

Déjanos Ayudarte

Conócenos

Comprometidos



Enlaces que aporta Accesibilidad

- Saltar a Contenido



Enlaces que aporta Accesibilidad

- Saltar a Contenido

```
<body>
  <nav>
    <ul>
      <li><a href="#contenido">Saltar a Contenido</a></li>
    </ul>
  </nav>
  <header>
    <h1>La FACET Diario</h1>
  </header>
  <nav>
    <ul>
      <li><a href="html/nosotros.html">Nosotros</a></li>
      <li><a href="html/eventos.html">Eventos</a></li>
      <li><a href="html/contacto.html">Contacto</a></li>
    </ul>
  </nav>
  <main>
    <section>
      <article>
        <h2 id="contenido">Argentina será sede de la copa América 2020</h2>
        <figure>
          
        </figure>
      </article>
    </section>
  </main>
</body>
```



Enlaces que aporta Accesibilidad

- La finalidad de este tipo de enlaces es que las personas que naveguen nuestro sitio con teclado (no videntes, con problemas motrices que no les permitan usar el mouse, etc.), puedan evitar recorrer todos los enlaces que haya en el sitio, e ir directo a lo que les interesa

Enlaces con **Target** (objetivo)

- Se usa cuando se desea que un enlace se abra en otra pestaña (o ventana), para que el usuario no "pierda" la página actual, ya que quedarán ambas abiertas
- `Enlace`

Unidades de Medida CSS

- Existen 2 tipos de Medidas CSS

1. Absolutas

- Su tamaño es igual, sin importar el tamaño del elemento que lo contiene

2. Relativas

- Su tamaño depende del tamaño del elemento que lo contiene

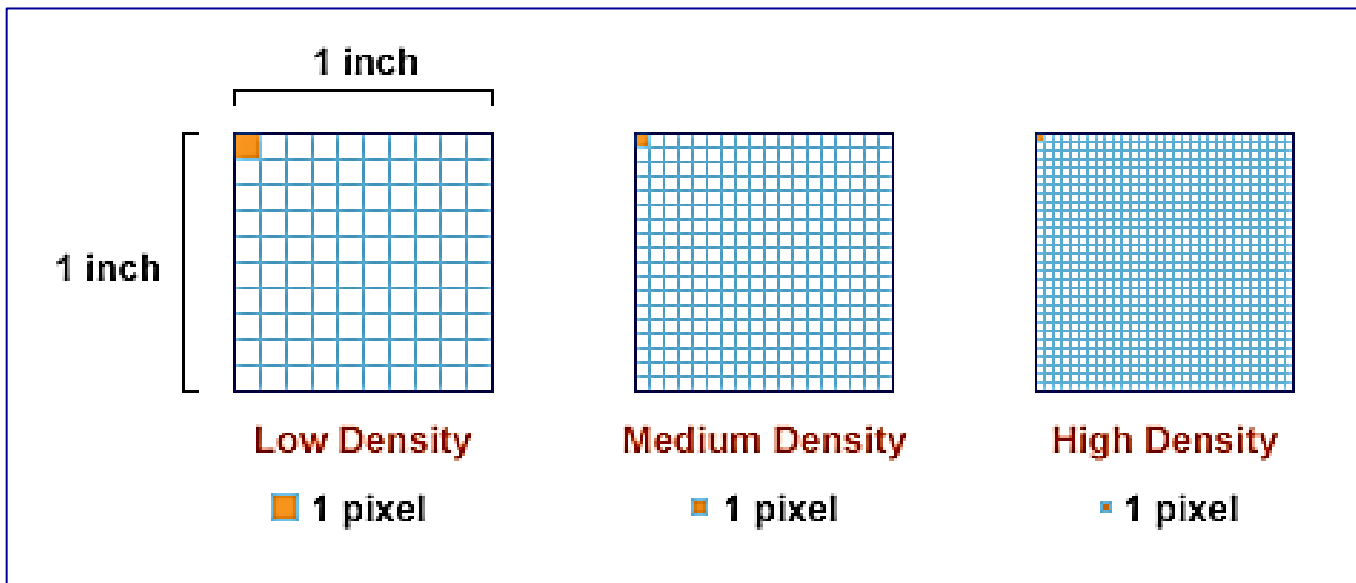
Unidades de Medida CSS

1. Absolutas

- La más utilizada es **px** (pixel), que representa un punto de la pantalla
 - Otras son cm, mm, in (pulgada), pt (punto), pc (pica)
- Problema de los **px**: no todas las pantallas muestran la misma cantidad de pixeles.

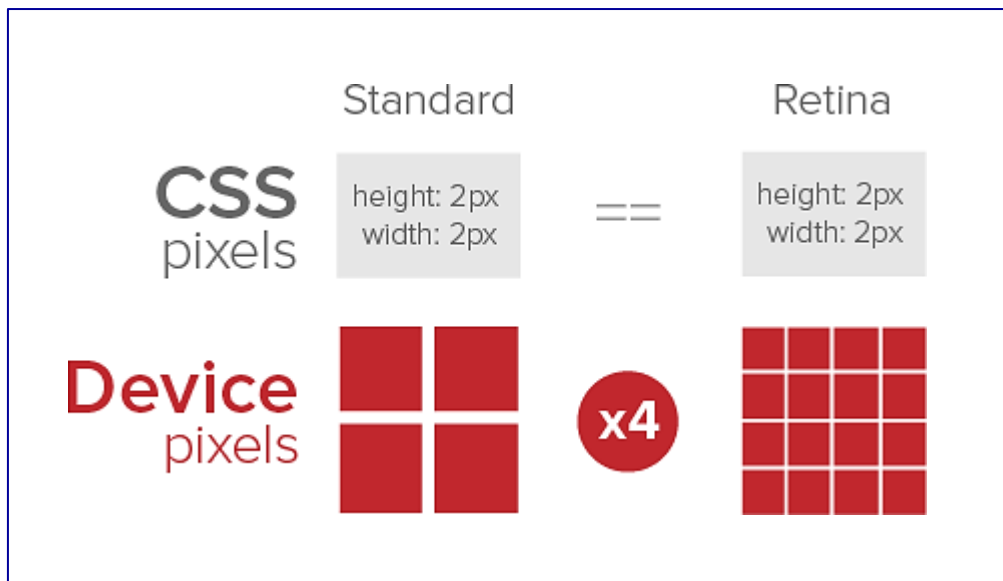
Unidades de Medida CSS

- Problema 1 de los **px**: observe que para el mismo tamaño físico de 1 pulgada de pantalla, 1 pixel se ve muy distinto, según sea la densidad de pixeles.



Unidades de Medida CSS

- Problema 2 de los **px**: existen tipos de pantallas (como la Retina), que multiplica los pixeles CSS



Unidades de Medida CSS

- Problema 2 de los **px**: Para qué multiplica los pixeles?

13-inch MacBook Pro



13-inch MacBook Pro with Retina display



Unidades de Medida CSS

2. **Relativas:** Las más utilizadas son:

- em
- rem: root em
- % (Porcentaje)
- vw: ancho del visor
- vh: alto del visor
- vmin: dimensión menor entre vw y vh
- vmax: dimensión mayor entre vw y vh

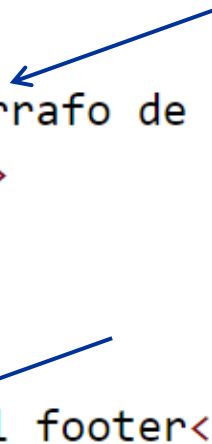
Unidades de Medida CSS

- **em**: la unidad depende del tamaño de fuente de la etiqueta contenedora más próxima.
- Por defecto los navegadores le dan a las etiquetas un tamaño de fuente de 16px.

Unidades de Medida CSS

- Observemos los dos <p>

```
<body>
  <section> <!-- contenedor principal-->
    <main>
      <section>
        <article>
          <p>Párrafo de Contenido Principal</p>
        </article>
      </section>
    </main>
  </section>
  <footer><p>Párrafo del footer</p></footer>
</body>
</html>
```

The diagram illustrates the HTML structure within a <body> tag. It shows a nested structure: a <section> containing a <main> tag, which in turn contains another <section> with an <article> tag. Inside the <article> tag is a paragraph <p>Párrafo de Contenido Principal</p>. Below the <main> tag is another <section> containing a <footer> tag with a paragraph <p>Párrafo del footer</p>. Two yellow boxes highlight the text 'Párrafo de Contenido Principal' and 'Párrafo del footer'. Two blue arrows point from the right towards these highlighted paragraphs, indicating they are the focus of the observation.

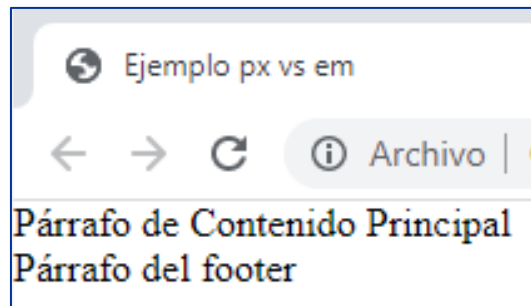
Unidades de Medida CSS

- Ahora probaremos las siguientes reglas CSS, una aplicando 16px y otra con 1em, para ver si tienen el mismo tamaño o no. Resultado: Tienen el mismo tamaño

```
/*--- Section main ---*/
section main section article p {
  font-size: 16px;
}

/*--- Section aside ---*/

/*--- Footer -----*/
footer p {
  font-size: 1em;
}
```



Unidades de Medida CSS

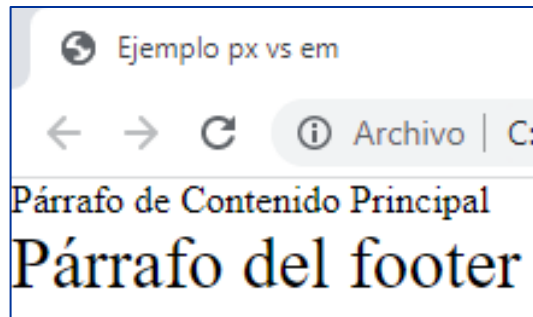
- Ahora agregaremos un tamaño de fuente al footer, de modo que veamos si se altera el tamaño de la p del footer.

```
/*--- Section main ---*/
section main section article p {
  font-size: 16px;
}

/*--- Footer -----*/
footer {
  font-size: 30px;
}

footer p {
  font-size: 1em;
}
```

Resultado: la regla con **em** ve modificado el tamaño de su fuente, ya que **depende del tamaño de la etiqueta que la contiene**



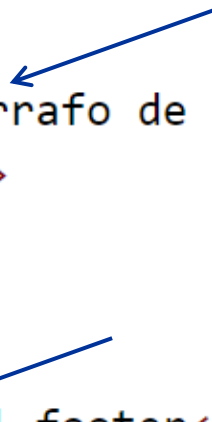
Unidades de Medida CSS

- **rem**: la unidad depende del tamaño de la fuente de la etiqueta `<html>`.
- Por defecto los navegadores le dan un tamaño de 16px.

Unidades de Medida CSS

- Sea el mismo html del ejemplo anterior

```
<body>
  <section> <!-- contenedor principal-->
    <main>
      <section>
        <article>
          <p>Párrafo de Contenido Principal</p>
        </article>
      </section>
    </main>
  </section>
  <footer><p>Párrafo del footer</p></footer>
</body>
</html>
```

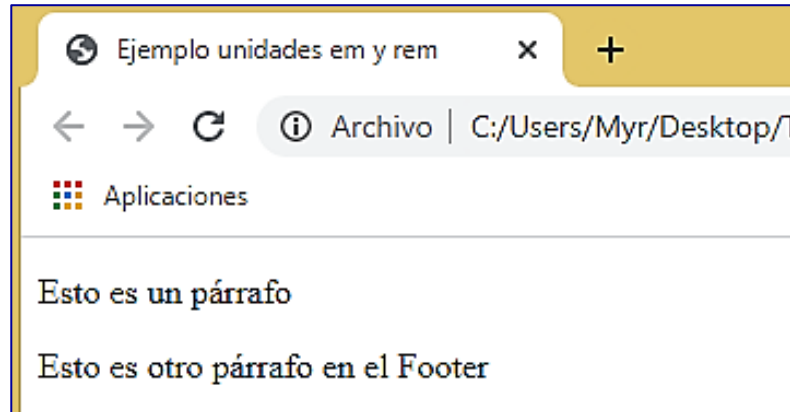
The diagram consists of two blue arrows. The first arrow originates from the right side of the slide and points to the paragraph element '<p>Párrafo de Contenido Principal</p>' within the nested HTML structure. The second arrow also originates from the right side and points to the paragraph element '<p>Párrafo del footer</p>' in the footer section of the HTML code. Both paragraph elements are highlighted with a yellow box in the original image.

Unidades de Medida CSS

- Ahora probaremos las siguientes reglas CSS, una aplicando 16px y otra con 1rem, para ver si tienen el mismo tamaño o no.

```
article p {  
    font-size: 16px;  
}  
  
footer p {  
    font-size: 1rem;  
}
```

Resultado: Tienen el mismo tamaño

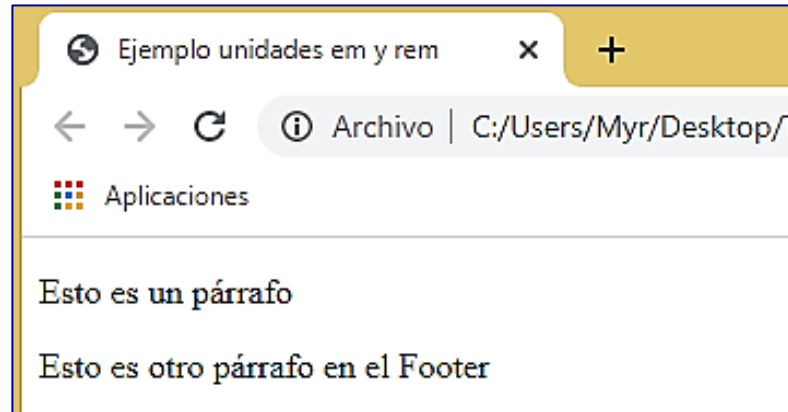


Unidades de Medida CSS

- Ahora agregaremos un tamaño de fuente al footer, de modo que veamos si se altera el tamaño de la p del footer.

```
article p {  
  font-size: 16px;  
}  
  
footer {  
  font-size: 30px;  
}  
  
footer p {  
  font-size: 1rem;  
}
```

Resultado: la regla con rem no cambia su tamaño, ya que no depende del tamaño de fuente de la etiqueta que la contiene

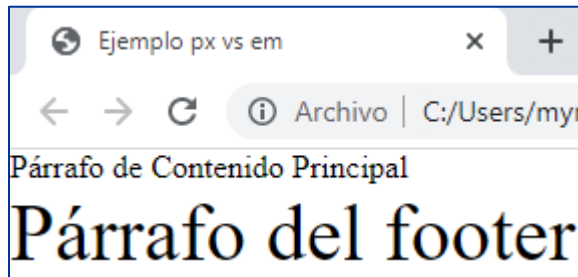


Unidades de Medida CSS

- Quitemos la regla de footer y modifiquemos el tamaño de fuente de la etiqueta html

```
html {  
  font-size: 40px;  
}  
  
section main section article p {  
  font-size: 16px;  
}  
  
/*--- Footer -----*/  
footer p {  
  font-size: 1rem;  
}
```

Resultado: la regla con **rem** cambia su tamaño, porque **siempre hace referencia al tamaño de la fuente de la etiqueta html**



Unidades de Medida CSS

- Recomendaciones de uso de em y rem
 - **em**: usar para definir los tamaños de fuente, las alturas y también para elementos que requieran una medida que tenga relación con el tamaño del texto, como por ejemplo el margen entre párrafos, el relleno interior de una etiqueta, etc.

Unidades de Medida CSS

- Recomendaciones de uso de em y rem
 - **rem**: aplicar a elementos de diseño que requieran medidas "fijas" y eventualmente también para textos que deseemos que tengan un tamaño de fuente que no dependa de su elemento padre.

Depende fuertemente de las preferencias del navegador del usuario, a menos que nosotros cambiemos el tamaño de fuente de la etiqueta html.

En general se debe tender a usar rem en vez de em

Unidades de Medida CSS

- % (Porcentaje): funciona similar a em, se refiere a un porcentual de la etiqueta que lo contiene.


Unidades de Medida CSS

- Ejemplo de porcentaje: supongamos que queremos que una imagen ocupe la mitad del ancho de una página (50%)

```
<main>
  <section>
    <article>
      <figure>
        
      </figure>
    </article>
  </section>
</main>
```

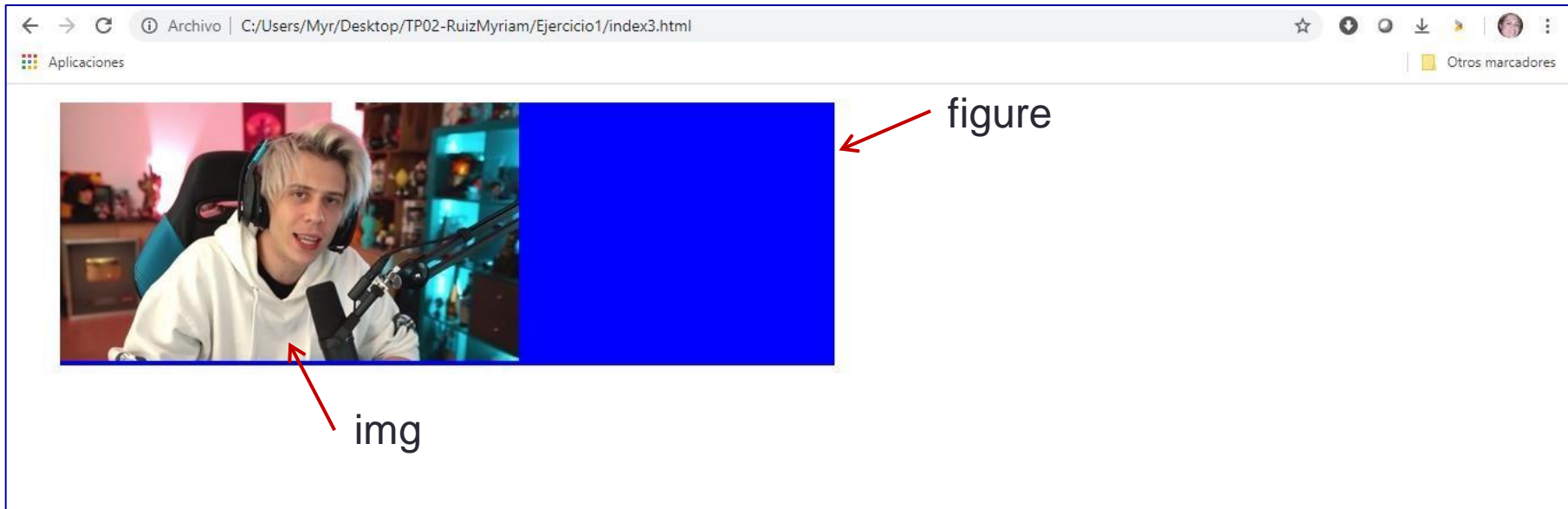
Unidades de Medida CSS

- Haremos que el figure ocupe el 50% del ancho, en este caso será del ancho de su contenedor que no tiene tamaño fijo, por lo que tomará el ancho de la página.

```
figure {  
  width: 50%;  
  background-color:  blue;  
}
```


Unidades de Medida CSS

- Observe que la imagen (img) no ocupa todo el espacio establecido para el figure (con fondo azul):



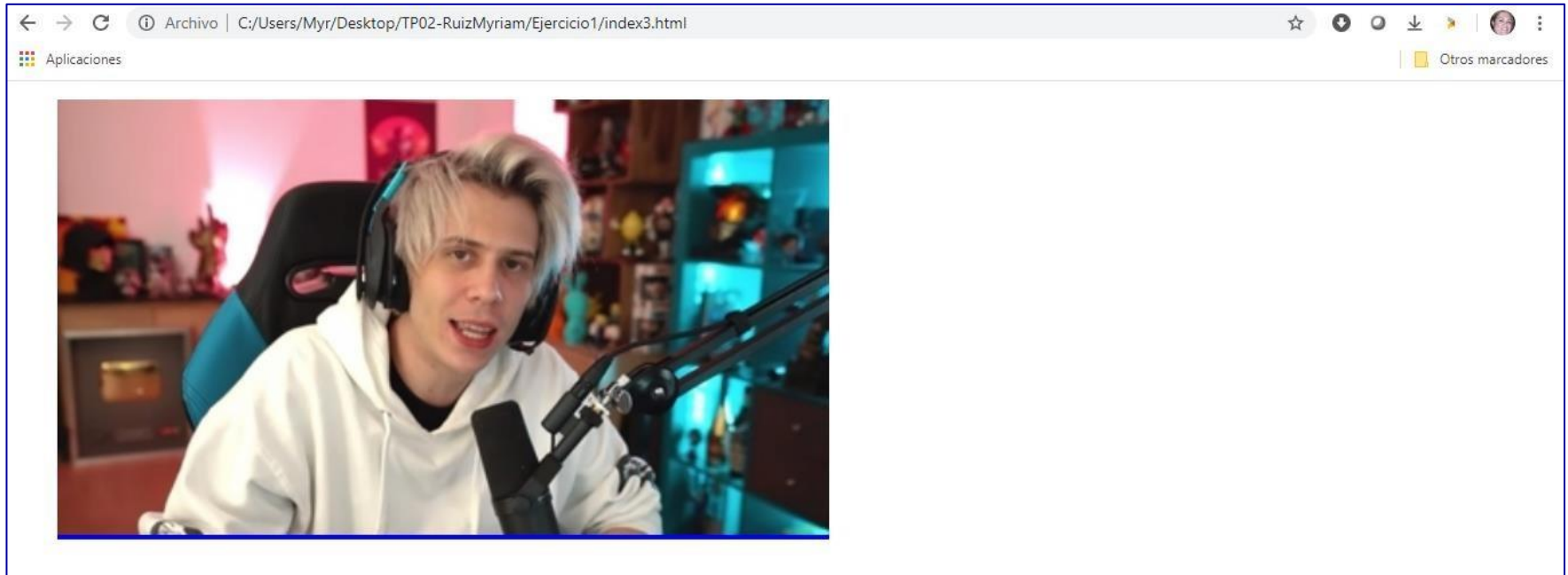
Unidades de Medida CSS

- Si establecemos una regla para la img dentro de figure, de modo que ocupe todo el ancho del figure:

```
figure {  
  width: 50%;  
  background-color:  blue;  
}  
  
figure img {  
  width: 100%;  
}
```

Unidades de Medida CSS

- Resultado: La imagen se adapta a todo el ancho del figure

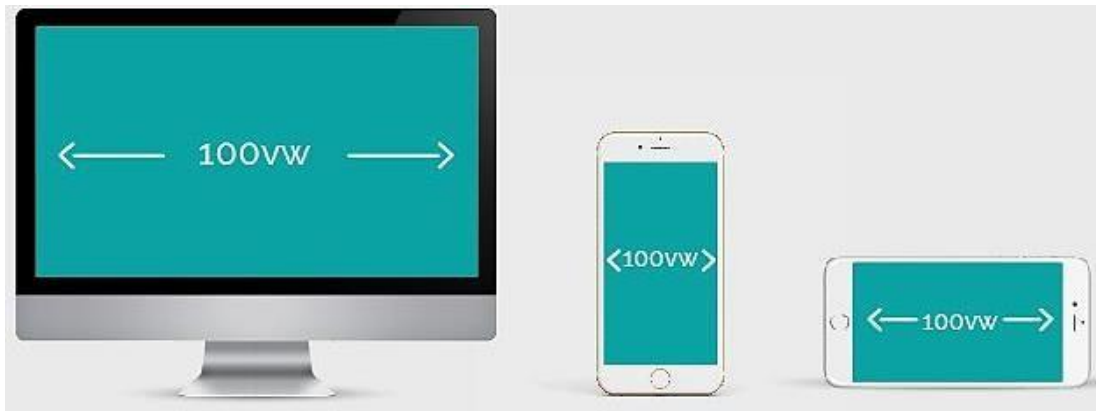


Unidades de Medida CSS

- **vw**: está basada en el ancho de vista de la ventana (navegador o visor).
- El valor de 1vw es equivalente al 1% del ancho del área de visualización.
- El 100vw equivale al 100% del ancho de área de visualización.

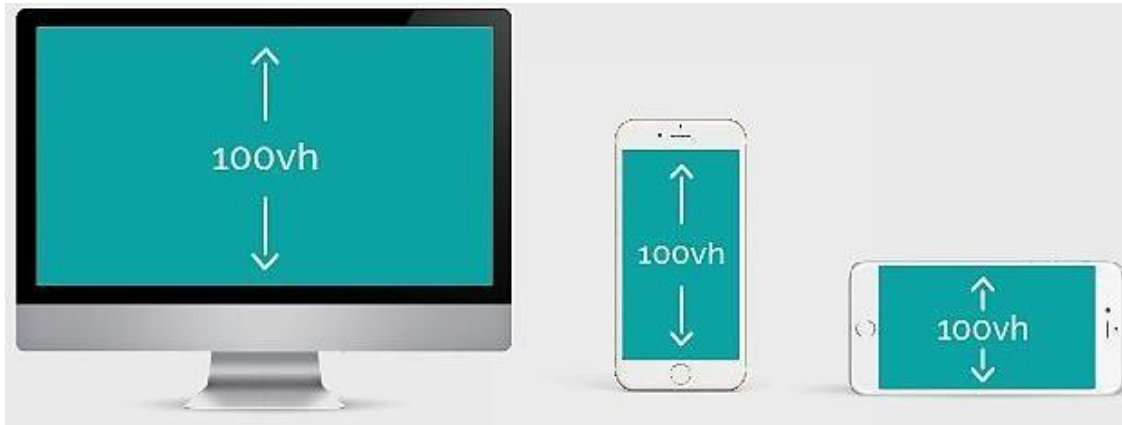
Unidades de Medida CSS

- **vw**: Sin importar el tamaño de pantalla, siempre hace referencia al ancho del área que se visualiza



Unidades de Medida CSS

- **vh**: similar al vw, se base en la altura del área de visualización



Unidades de Medida CSS

- Cabe preguntarse cuál es la diferencia entre vh, vw y %?
- 1- En el caso de porcentajes, el ancho o alto del elemento hijo es determinado con respecto a su padre. Con vw no existe esa dependencia



Unidades de Medida CSS

- Cabe preguntarse cuál es la diferencia entre vh, wh y %?
- 2- La propiedad que le da altura a los elementos (height) no puede expresarse en porcentajes, en vez de eso puede usar la unidad de medida vh.
- Ejemplo:

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

Equivalencia de px y rem

- Suponiendo que el tamaño de fuente por defecto de la etiqueta `<html>` es 16px, tenemos la siguiente equivalencia:

px	rem
16px	1rem
8px	0.5rem
4px	0.25rem
2px	0.12rem
1px	0.06rem

Recomendación para organizar el CSS

1. Modificar las propiedades del sitio desde arriba hacia abajo y de izquierda hacia derecha (Hacer coincidir el orden del CSS con el del HTML)
2. Delimitar el CSS mediante comentarios
3. Escribir todo en minúsculas

Recomendación para organizar el CSS

```
* {  
    margin: 0; /* elimina márgenes */  
    padding: 0; /* elimina espacios internos */  
    border: 0; /* elimina bordes */  
}  
  
/*--- Encabezado ---*/  
  
/*--- Menú -----*/  
  
/*--- Section main -*/  
  
/*--- Section aside -*/  
  
/*--- Footer -----*/
```

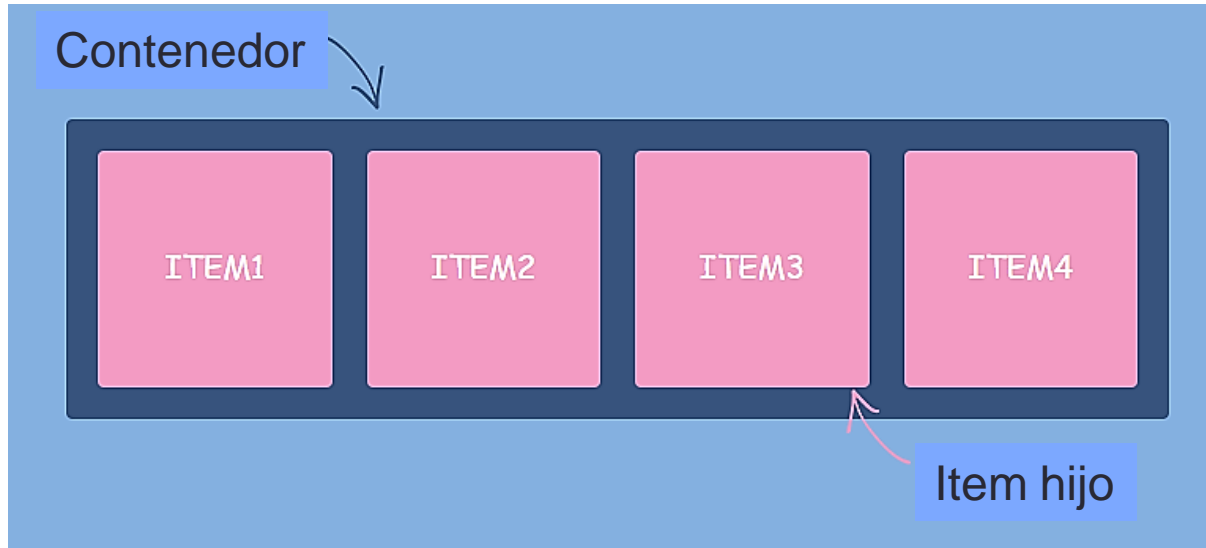
display: flex (básico)

- Permite
 - Alinear elementos
 - Distribuir el espacio entre los elementos de un contenedor, incluso cuando su tamaño es desconocido y/o dinámico (es decir, que puede cambiar)
 - Lograr alturas fijas de manera sencilla

Conceptos Principales de Flexbox

- Siempre habrá un **contenedor** (padre) y los elementos **item** que están dentro de él (**hijos inmediatos**). Ej: section (padre) y articles (hijos).
- Se trabaja siempre en **una única dimensión por vez** (horizontal o vertical)

Esquema básico de Flexbox



Propiedades del contenedor

- **display:** flex | inline-flex;
- **flex-direction:** row | column; // row = fila
- **justify-content:** flex-start | flex-end | center | space-between | space-around | space-evenly;
- **align-items:** flex-start | flex-end | center | baseline | stretch;

Propiedades del Contenedor

- **El contenedor debe mostrarse flexible**, para permitir que sus hijos se orienten de manera horizontal o vertical. Para ellos se escribe:

```
contenedor {  
    display: flex;  
}
```

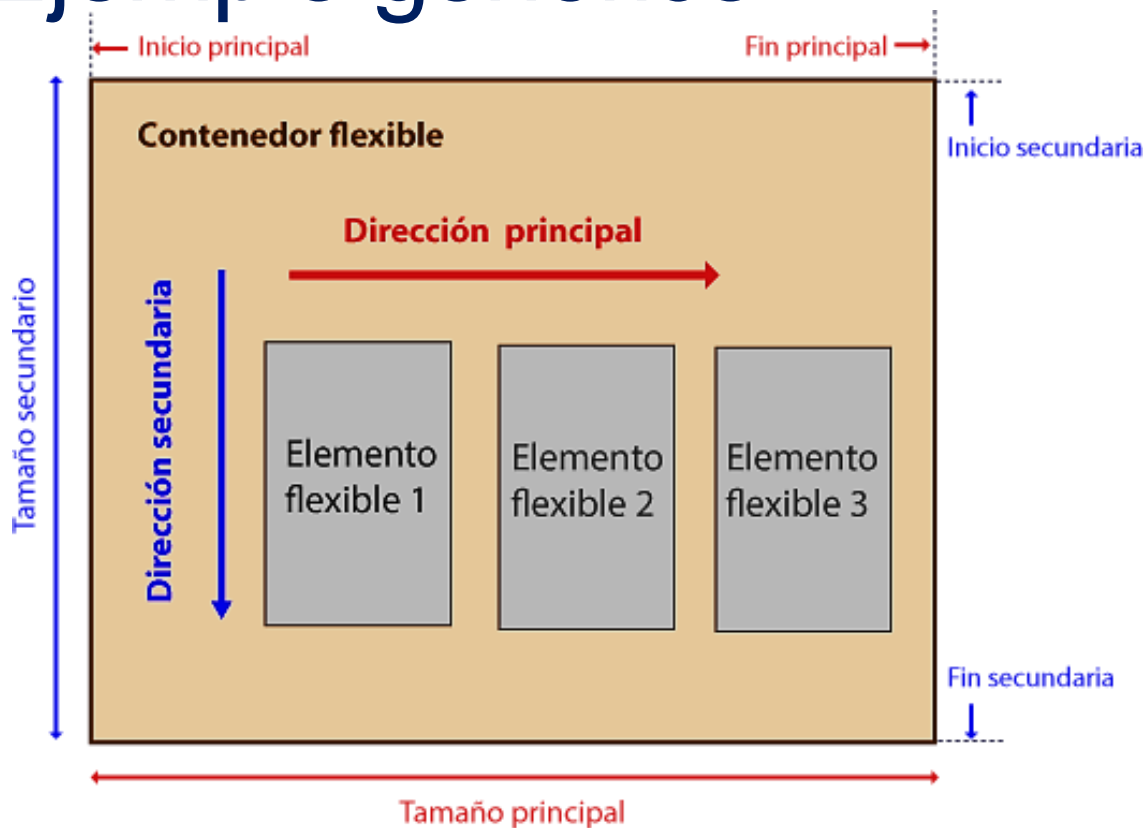
- Donde contenedor puede ser cualquier etiqueta o selector

Propiedades del Contenedor: Dirección

- Se debe definir una dirección para la distribución de los items. Luego de eso se generan unos ejes imaginarios con la dirección correspondiente a la elección.
- La propiedad del contenedor para indicar la dirección se llama **flex-direction**, que puede tomar los valores **row** (fila) o **column** (columna)

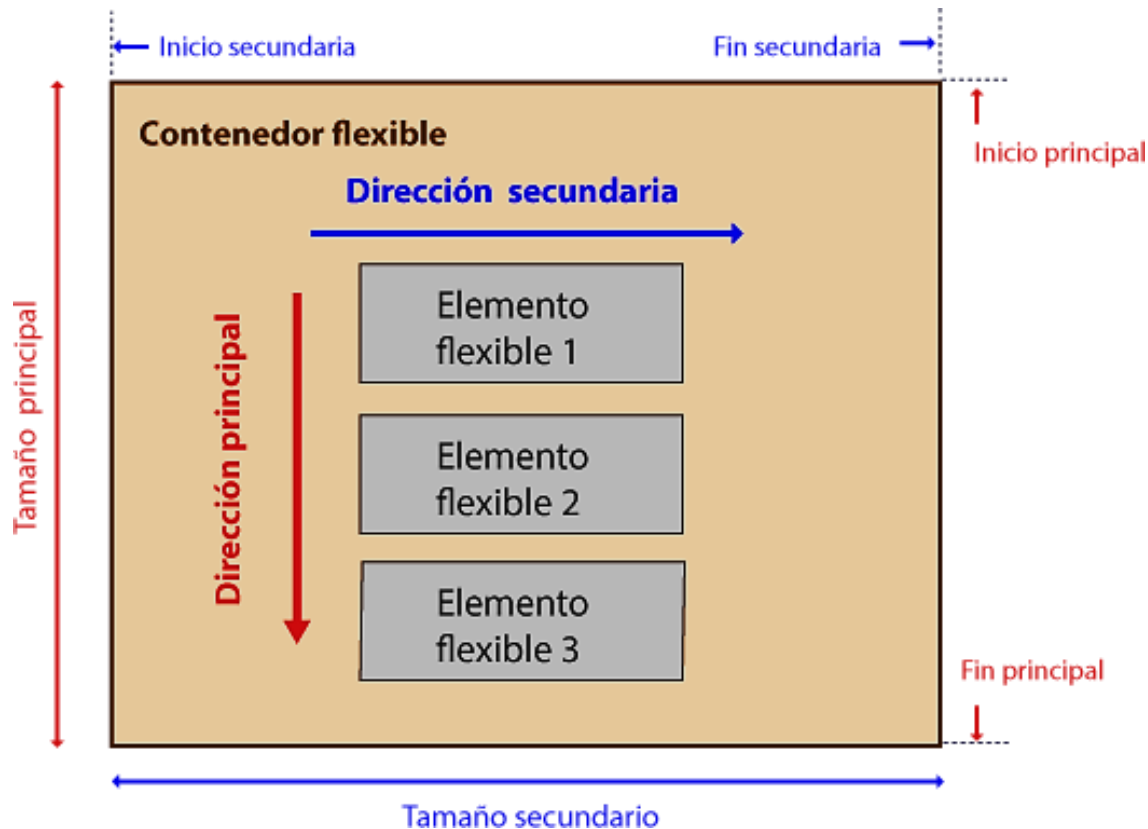
Dirección – Ejemplo genérico

```
contenedor {  
  display: flex;  
  flex-direction: row;  
}
```



Dirección – Ejemplo genérico

```
contenedor {  
  display: flex;  
  flex-direction: column;  
}
```



Ejemplo de display: flex con flex-direction

```
<main>
  <section>
    <article>
      <h2>Título Caja 1</h2>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Sequi nemo, veniam ducimus laborum
et eligendi facilis. Cupiditate, vel dolorum in et quam, dignissimos, doloribus culpa
distinctio quae suscipit facere quisquam?</p>
    </article>
    <article>
      <h2>Título Caja 2</h2>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Officia deserunt, nulla tempora
voluptates labore ratione est odio eos voluptatem, aspernatur voluptatum a in ea error nam,
earum quia doloribus ut.</p>
    </article>
    <article>
      <h2>Título Caja 3</h2>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quis alias quas vel sed tempore
cupiditate libero qui expedita, laudantium nisi et reiciendis, deleniti assumenda ipsam quia
facere, iure explicabo. Placeat!</p>
    </article>
  </section>
```

Ejemplo de display: flex con flex-direction

```
header {  
  height: 10vh;  
}  
  
section article {  
  background-color: #ffff80;  
  padding: 1em;  
  margin: 2%;  
}  
  
section {  
  height: 90vh;  
  background-color: #ffa8ff;  
}  
  
main section {  
  display: flex;  
  flex-direction: row;  
}
```



Ejemplo de display: flex con flex-direction

Ejemplo de Cajas Flexibles

Título Caja 1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Sequi nemo, veniam ducimus laborum et eligendi facilis. Cupiditate, vel dolorum in et quam, dignissimos, doloribus culpa distinctio quae suscipit facere quisquam?

Título Caja 2

Lorem ipsum dolor sit amet consectetur adipisicing elit. Officia deserunt, nulla tempora voluptates labore ratione est odio eos voluptatem, aspernatur voluptatum a in ea error nam, earum quia doloribus ut.

Título Caja 3

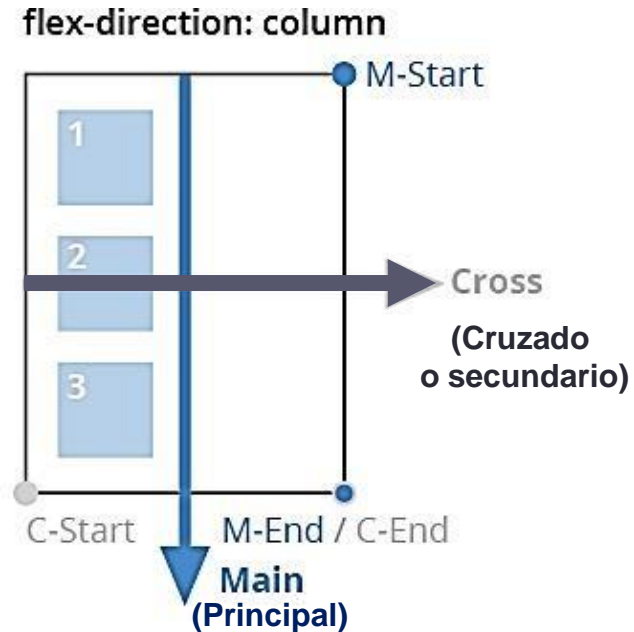
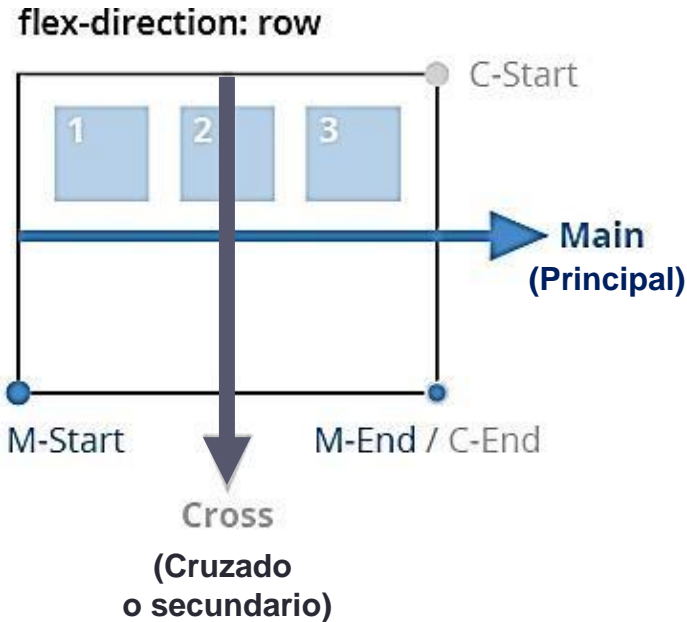
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quis alias quas vel sed tempore cupiditate libero qui expedita, laudantium nisi et reiciendis, deleniti assumenda ipsam quia facere, iure explicabo. Placeat!

Propiedades del Contenedor: **Justificación**

- Cuando elegimos la dirección con flex-direction, se genera un eje principal y un eje secundario, ambos imaginarios.
- A partir de ahí se puede **distribuir los items hijos a lo largo del eje principal**
- Para ello el contenedor debe tener establecida la propiedad **justify-content**

Propiedades del Contenedor: **Justificación**

- Detalle de los ejes



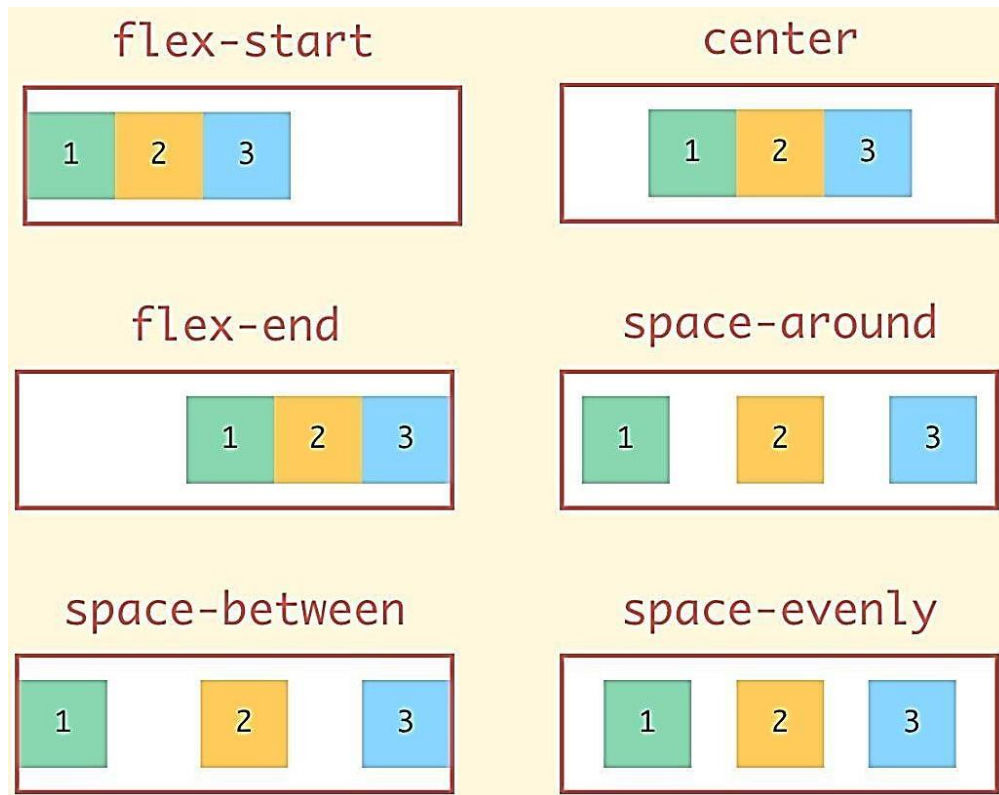
Propiedades del Contenedor: **Justificación**

- Valores para la propiedad **justify-content**:
 - **flex-start**: ubica los items al **inicio** del eje principal
 - **flex-end**: ubica los items al **final** del eje principal
 - **center**: ubica los items al **centro** del eje principal
 - **space-between**: deja el mismo espacio entre los items
 - **space-around**: deja el mismo espacio a ambos lados de los items, por lo que en medio tiene espacio doble
 - **space-evenly**: distribuye el espacio de manera que sean iguales

Propiedades del Contenedor: Justificación

Sea flex-direction: row;

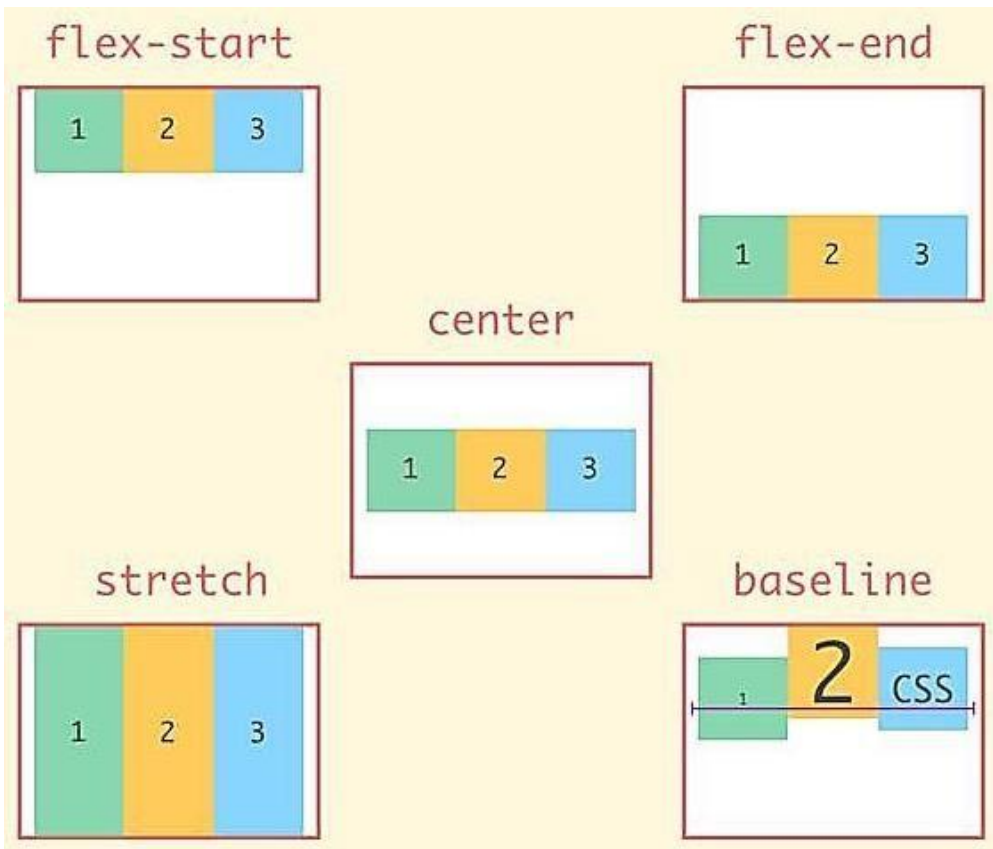
justify-content:



Propiedades del Contenedor: **Alineación**

Sea **flex-direction: row;**

align-items:



Propiedades del Contenedor: **Alineación**

- Ejemplo: alinear logo y h1 en el header



Sitio de Laboratorio I

```
<header>
  
  <h1>Sitio de Laboratorio I</h1>
</header>
```

```
header {
  display: flex;
  flex-direction: row; /* acomoda uno al lado del otro */
  justify-content: center; /* centra a lo ancho */
  align-items: center; /* centrar verticalmente */
}
```

Propiedades del Contenedor: **Alineación**

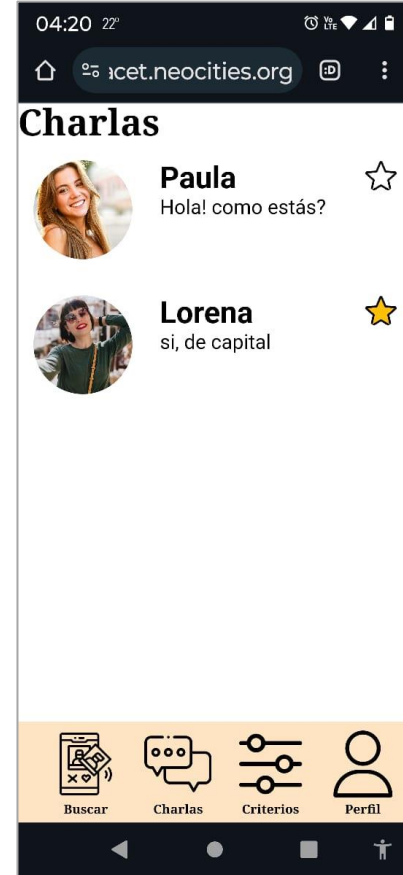
- Ejemplo: alinear logo y h1 en el header



```
header {  
  display: flex;  
  flex-direction: row; /* acomoda uno al lado del otro */  
  justify-content: center; /* centra a lo ancho */  
  align-items: flex-start; /* ubica los elementos arriba */  
}
```

Ejemplo

- Alinear los botones del menú
- Alinear imagen, con texto y otra imagen



Ejemplo

- Alinear los botones del menú: vamos a acomodar los items

```
<nav>
  <ul id="contenedor-enlaces">
    <li>
      <a href="#">
        <figure>
          
          <figcaption>Buscar</figcaption>
        </figure>
      </a>
    </li>
    <li><a href="html/charlas.html">
      <figure>
        
```

Ejemplo

- Alinear los botones del menú: vamos a acomodar los items

```
#contenedor-enlaces {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-evenly;  
  list-style-type: none;  
}
```

Vuelve al **ul** una caja flexible

Muestra a sus hijos (**li**) en una fila

Reparte el espacio equitativamente

```
#contenedor-enlaces figure {  
  width: 17vw;  
  color: ■ black;  
}
```

Le da un ancho a las imágenes

```
#contenedor-enlaces figure img {  
  width: 100%;  
}
```

Ejemplo

- Alinear los botones del menú: vamos a acomodar los items

```
#contenedor-enlaces figure img {  
    width: 100%;  
}  
  
#contenedor-enlaces figure figcaption {  
    text-align: center;  
    font-weight: bold;  
    font-size: 0.7em;  
}
```

Descubrir



Buscar



Charles



Criterios



Perfil

Ejemplo

- Dejar el menú fijo en la parte inferior de la pantalla, para que sea más sencillo utilizar los enlaces

```
nav {  
  position: fixed;  
  top: 79vh;  
  width: 100vw;  
  background-color: #e6b89c; /* bisque */  
  padding: 0.5em;  
}
```

Deja fija la
posición del nav
Separa el nav
desde la parte
superior

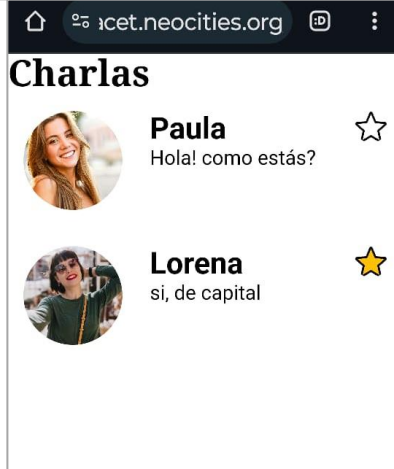
Descubrir



Ejemplo

- Alinear imagen, con texto y otra imagen

```
<main>
  <section>
    <article>
      <figure>
        
      </figure>
      <section class="charla">
        <h2>Paula</h2>
        <p>Hola! como estás?</p>
      </section>
      <figure>
        
      </figure>
    </article>
```



Ejemplo

- Alinear imagen, con texto y otra imagen

```
main article {
```

```
  display: flex;
```

```
  flex-direction: row;
```

```
  justify-content: space-between;
```

```
  padding: 4vw;
```

```
}
```

```
.charla {
```

```
  display: flex;
```

```
  flex-direction: column;
```

```
  width: 45vw;
```

```
  font-family: Verdana, Geneva, Tahoma, sans-serif;
```

```
  font-size: 1.1em;
```

```
}
```

Muestra cada artículo como fila

Le da espacio entre figure, section y figure

Agrega espacio interno al artículo para que no quede pegado al borde

Muestra p debajo de h2

Contenedores sugeridos:

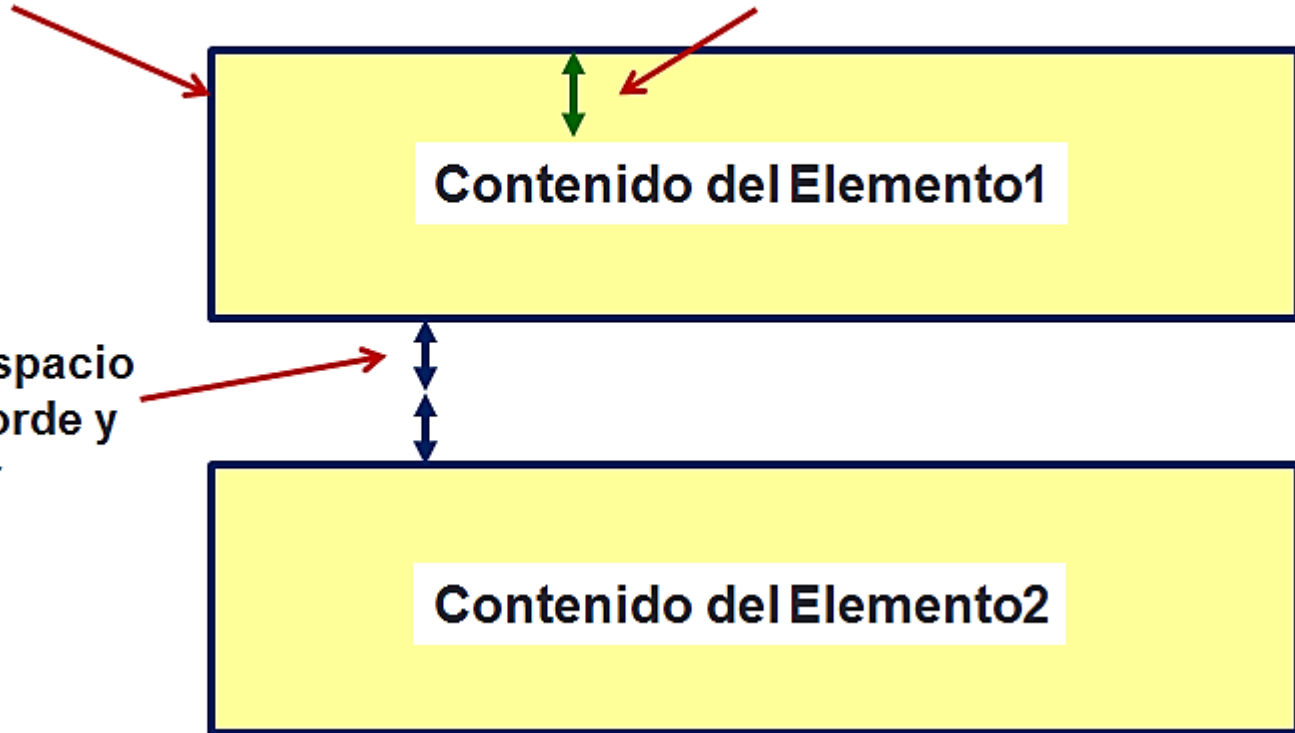
Contenedor	Item Hijo directo posible
body	header, main, nav, aside, footer
header	h1, figure, nav (si no está fuera)
ul	li
main	section
section	article, main , aside
figure	img, figcaption
form	label, input, textarea (tema que se verá más adelante)
footer/aside	p, h2, h3

Diferencia entre Margin, Padding y Border

Border: Contorno del elemento

Padding: Espacio entre el border y el contenido de un elemento

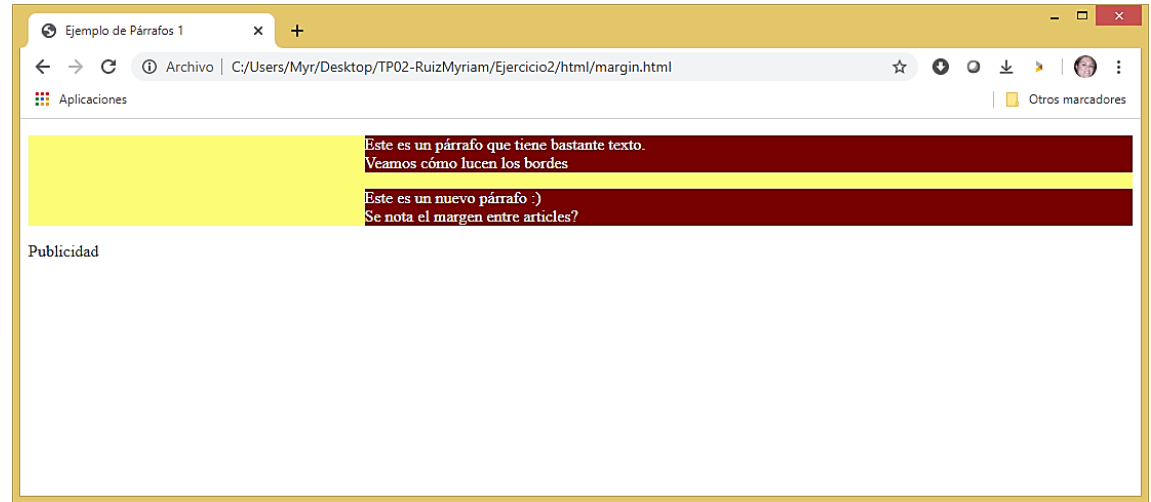
Margin: Espacio entre el borde y el exterior



CSS – margin (margen) individual

- Ejemplo: margen izquierdo e inferior:

```
article {  
    margin-left: 30vw;  
    margin-bottom: 1vh;  
}
```



CSS – **margin** (margen) para Centrar

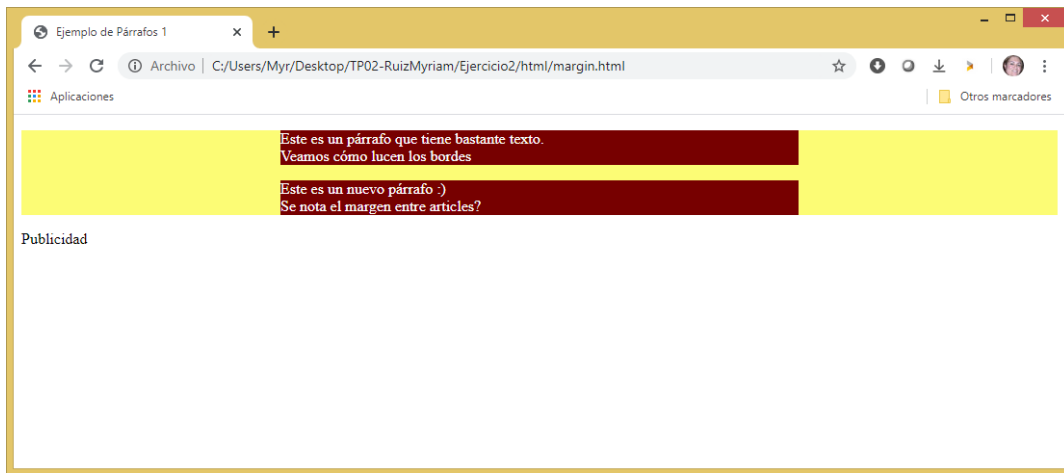
- Los márgenes se pueden usar para **centrar elementos horizontalmente**, que tengan establecido un ancho menor al de la página.

```
selector {  
    ancho: tamaño;  
    margin: superior-e-inferior izquierdo-y-derecho  
}
```

CSS – margin (margen) para Centrar

- Ejemplo 3: Centrar los article (de fondo oscuro)

```
article {  
    width: 50%;  
    margin: 0 auto;  
}
```

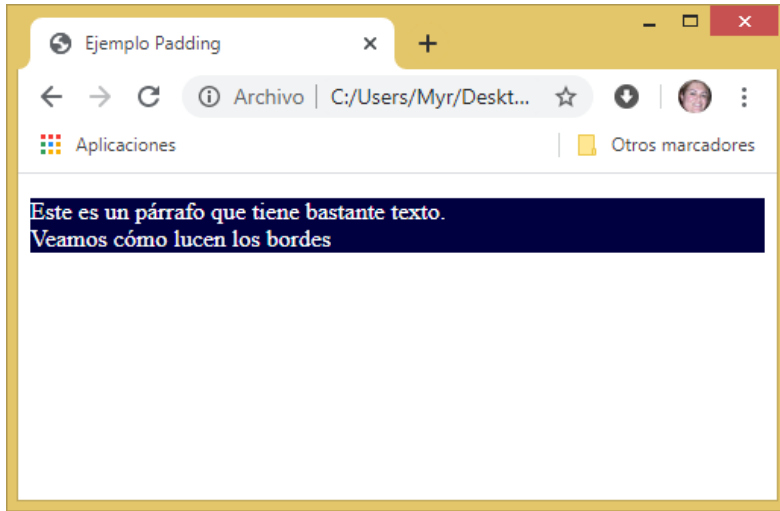


- **auto** indica que se calculan los márgenes de manera automática

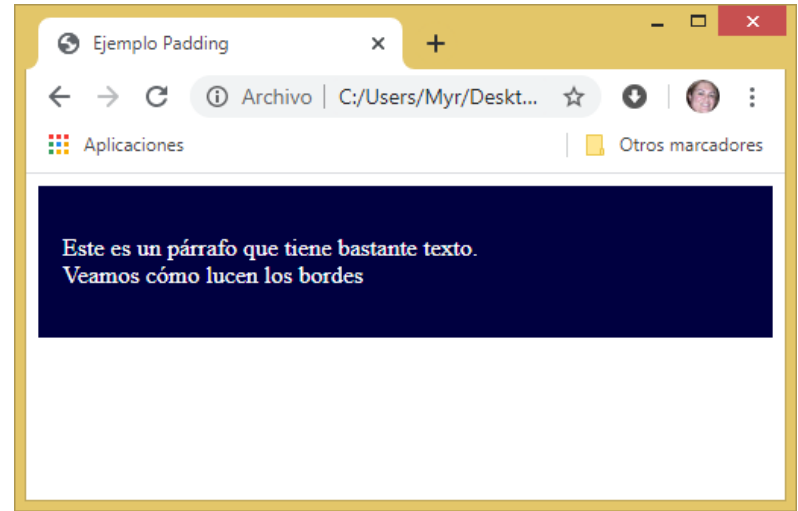
CSS – padding (espacio interno)

- Ejemplo: idéntico espacio en los cuadros lados

Sin padding



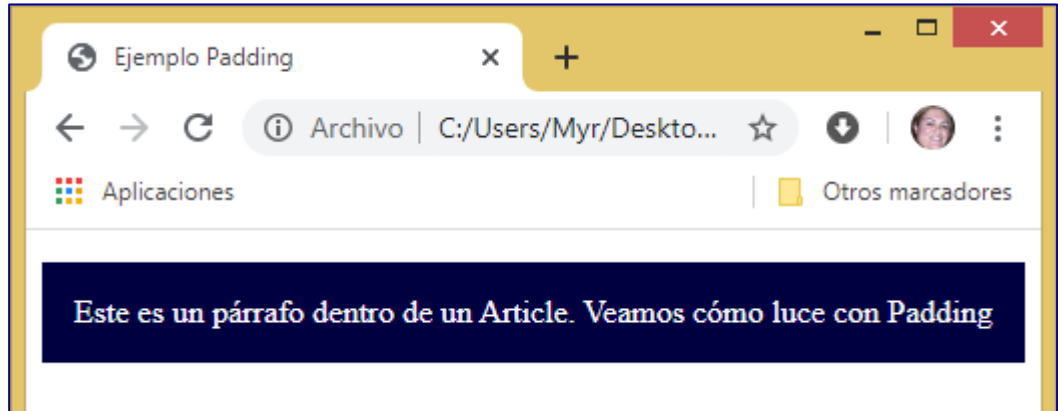
Con Padding



CSS – padding (espacio interno)

- Ejemplo: idéntico espacio en los cuadros lados

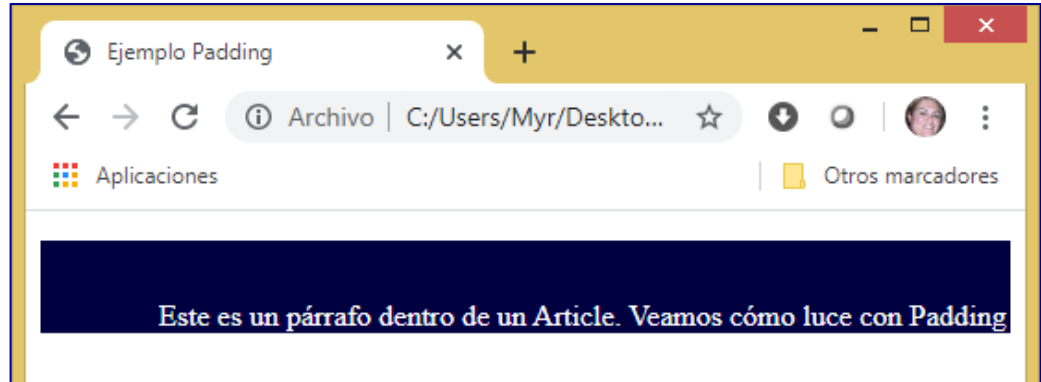
```
article p {  
    padding: 1rem;  
}
```



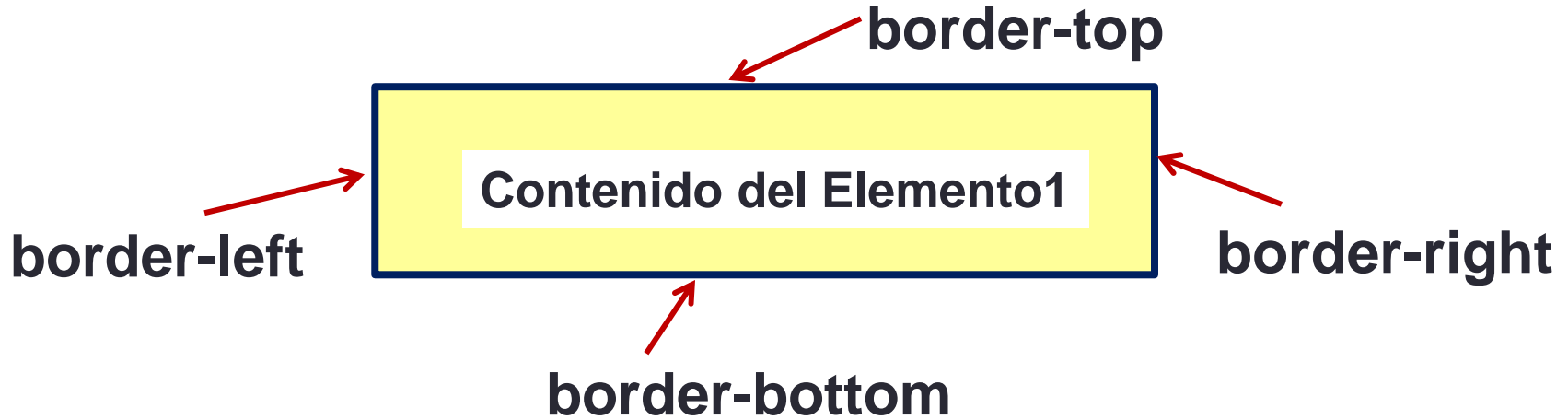
CSS – padding (espacio interno)

- Ejemplo de padding izquierdo y superior.

```
article p {  
    padding-top: 2rem;  
    padding-left: 4rem;  
}
```



CSS - Border



CSS - Border



CSS - Border

-width



Se mide en px, em, rem, o tamaños predefinidos thin, medium, thick

-style



solid, dotted, dashed, ridge, double, etc.

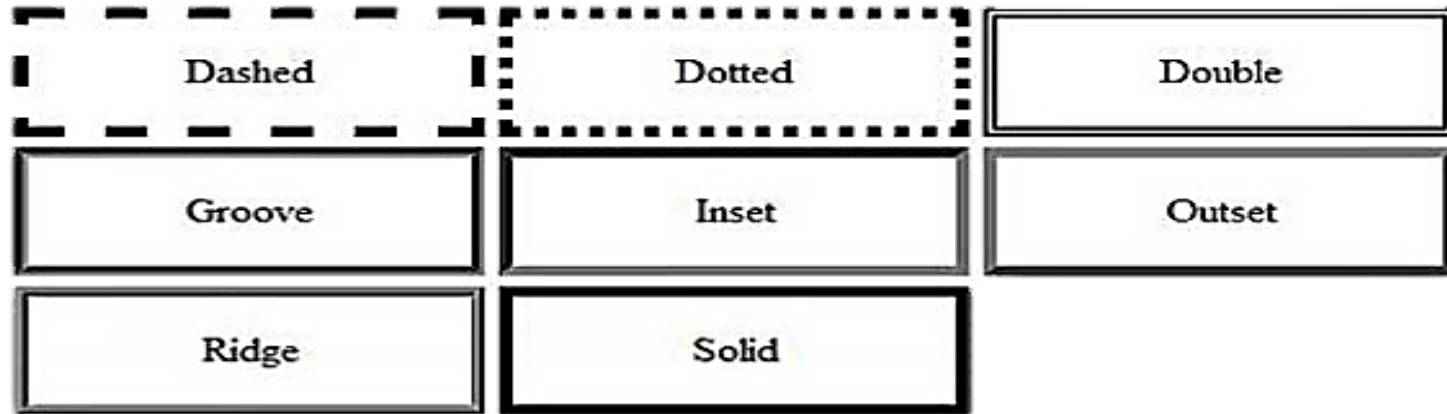
-color



RGB (#BB0000) o en inglés (Red)

CSS - Border

-style

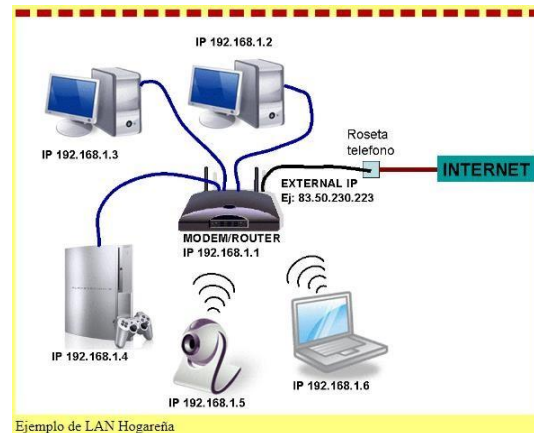


CSS - Border

- Tres formas de escribir los cuatro bordes:
 - Borde de **un lado**. Ejemplo para borde superior (top)

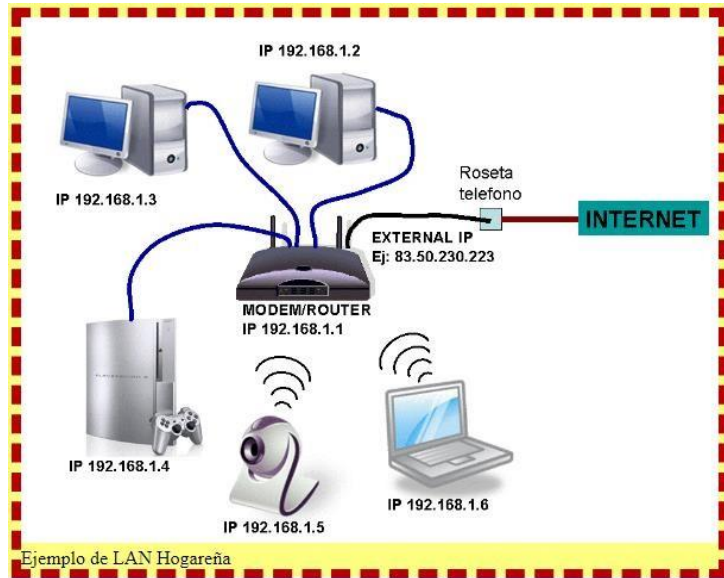
```
<figure class="ilustracion">  
    
  <figcaption>Ejemplo de LAN Hogareña</figcaption>  
</figure>
```

```
.ilustracion {  
  width: 60%;  
  margin: 0 auto;  
  border-top-width: 0.5em;  
  border-top-style: dashed;  
  border-top-color: ■ #BB0000;  
}
```



CSS - Border

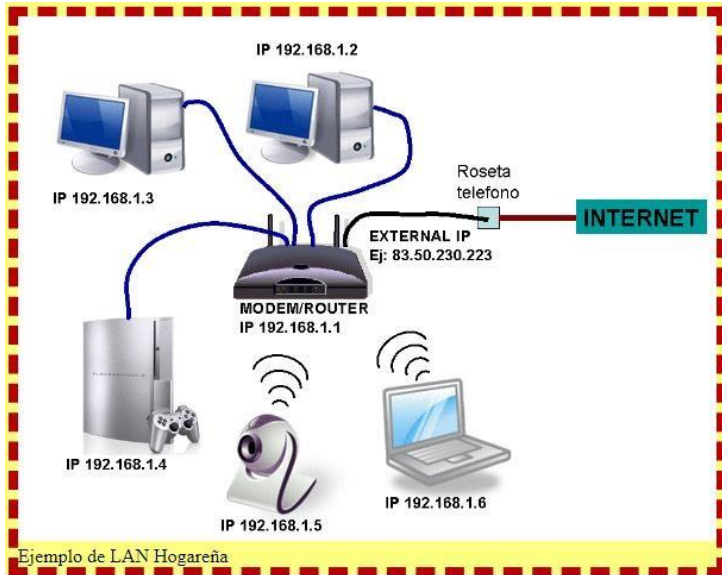
- Tres formas de escribir los cuatro bordes:
 - Los **cuatro bordes iguales**, con las propiedades separadas



```
.ilustracion {  
    width: 60%;  
    margin: 0 auto;  
    border-width: 0.5em;  
    border-style: dashed;  
    border-color: #BB0000;  
}
```

CSS - Border

- Tres formas de escribir los cuatro bordes:
 - Los **cuatro bordes iguales**, con las propiedades **juntas**



```
.ilustracion {  
    width: 60%;  
    margin: 0 auto;  
    border: 0.5em dashed ■ #BB0000;  
}
```

Nota: Se requiere al menos grosor y estilo para ver un borde. Los valores pueden ir en cualquier orden

CSS - Border

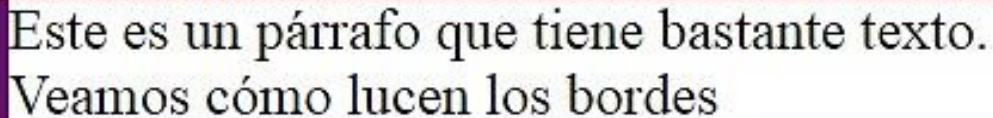
- Hacer los 4 bordes diferentes
 - El orden para declarar los borde es:

top	right	bottom	left
(arriba	derecha	abajo	izquierda)

CSS - Border

- Hacer los 4 bordes diferentes

```
.bordes-parrafo {  
    font-size: 1.5em;  
    /* arriba derecha abajo izquierda*/  
    border-width: thin medium thick 2em;  
    border-style: dashed dotted solid outset;  
    border-color: ■ #BB0000 ■ #157415 ■ #0000FF ■ #650F70;  
}
```



Este es un párrafo que tiene bastante texto.
Veamos cómo lucen los bordes

CSS – **border-radius** (borde redondeado)

- La propiedad CSS **border-radius** permite definir qué tan redondeadas serán las esquinas, de cualquier elemento.

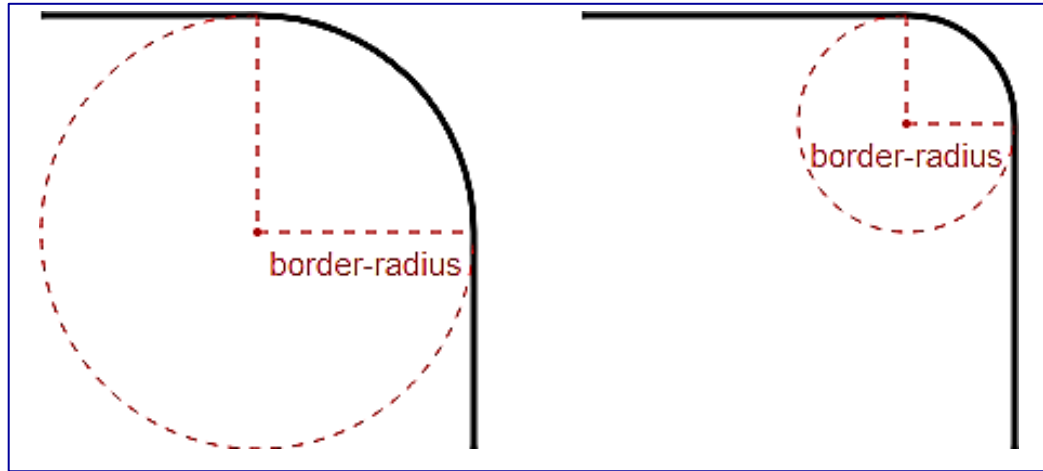
CSS – **border-radius** (borde redondeado)

- La redondez de cada esquina está definida usando un valor que representa un arco de circunferencia imaginario, cuyo tamaño está dado por el radio.
- Se "recorta" la esquina tomando el contorno del arco de circunferencia



CSS – **border-radius** (borde redondeado)

- A mayor tamaño del radio, mayor el redondeo del borde



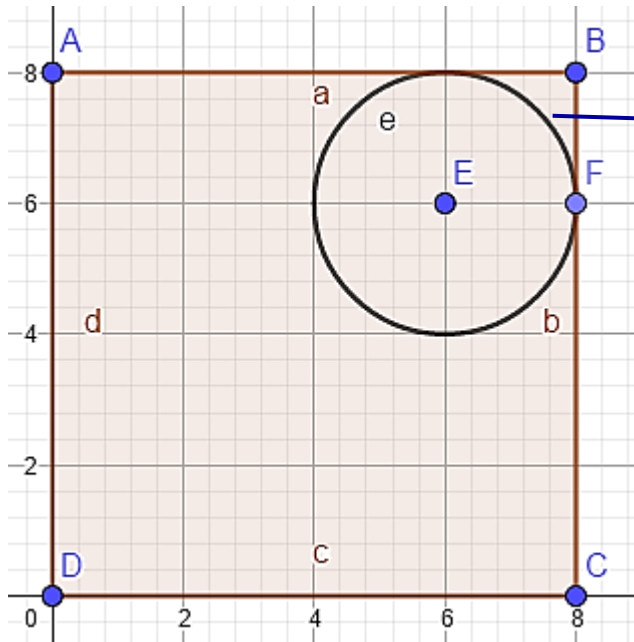
CSS – **border-radius** (borde redondeado)

- Ejemplo: le daremos bordes a un artículo del aside

```
<aside>
  <article class="logueo">
    <h2>Logueo</h2>
    <a href="html/logueo.html">Ingresar</a>
  </article>
</aside>
```

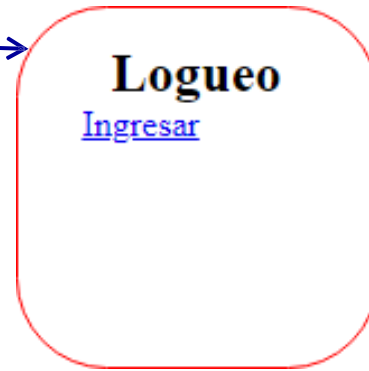
CSS – **border-radius** (borde redondeado)

- Tamaño esperado: Bosquejo en Geogebra de caja de 8rem de lado y una circunferencia con 2rem de radio



Resultado en la Web

25%



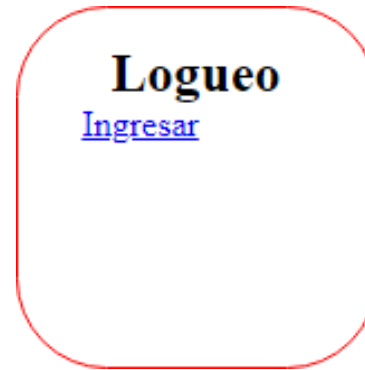
Le damos un radio de 2, es decir, el 25% de 8, que es el ancho del article

CSS – **border-radius** (borde redondeado)

- Tamaño logrado: CSS para caja de 8rem de lado, una circunferencia con 2rem de radio

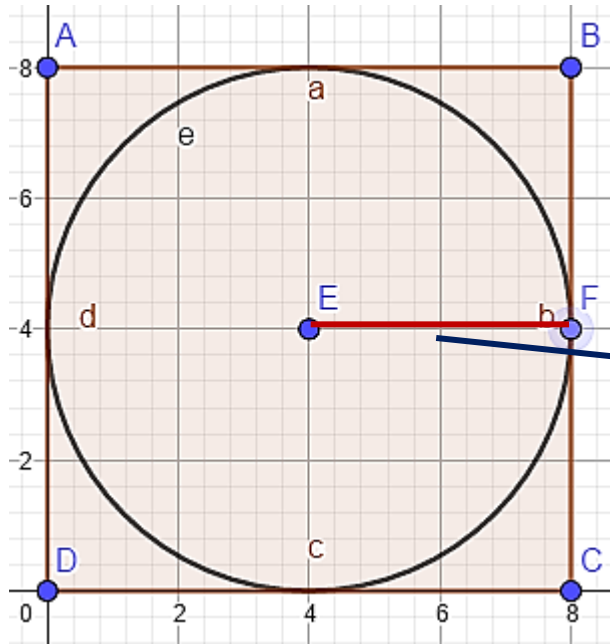
Resultado en la Web

```
.logueo {  
  border: thin solid ■ #FF0000;  
  width: 8rem;  
  height: 8rem;  
  border-radius: 25%;  
}
```



CSS – border-radius (borde redondeado)

- Ejemplo 2: cómo hacer un contorno circular?



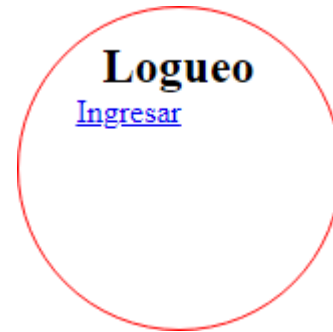
El radio debiera ser de 4rem, el 50%

```
.logueo {  
  border: thin solid ■ #FF0000;  
  width: 8rem;  
  height: 8rem;  
  border-radius: 50%;  
}
```

CSS – **border-radius** (borde redondeado)

- Ejemplo 2: cómo hacer un contorno circular? Resultado final

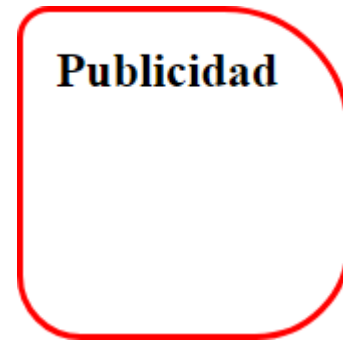
```
.logueo {  
  border: thin solid ■ #FF0000;  
  width: 8rem;  
  height: 8rem;  
  border-radius: 50%;  
}
```



CSS – **border-radius** (borde redondeado)

- Cómo el border tradicional, se puede indicar que cada borde-radius sea diferente:
- **border-radius:** sup-izquierdo sup-derecho inf-derecho inf-izquierdo

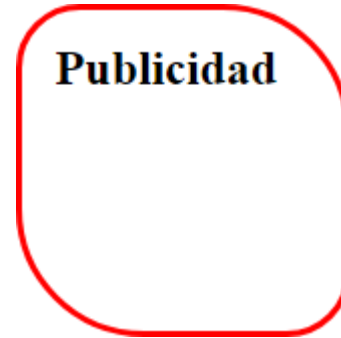
```
.publicidad {  
  border: 0.2em solid ■ #FF0000;  
  width: 8rem;  
  height: 8rem;  
  border-radius: 1rem 4rem 3rem 2rem;  
}
```



CSS – **border-radius** (borde redondeado)

- Se pueden indicar de a pares:
- `border-radius: sup-izq-e-inf-der sup-der-e-inf-izq`

```
.publicidad {  
  border: 0.2em solid ■ #FF0000;  
  width: 8rem;  
  height: 8rem;  
  border-radius: 2rem 4rem;  
}
```



CSS – **border-radius** (borde redondeado)

- Redondear imágenes que no son cuadradas:

Charlas



Paula

Hola! como
estás?



Lorena

si, de capital



Charlas



Paula

Hola! como estás?



Lorena

si, de capital



CSS – **border-radius** (borde redondeado)

- Redondear imágenes que no son cuadradas:

Charlas



Paula

Hola! como estás?



Lorena

si, de capital



```
main figure img {  
  width: 25vw;  
  height: 25vw;  
  object-fit: cover;  
  border-radius: 50%;  
}  
  
.vacía, .llena {  
  width: 8vw;  
  height: 8vw;  
}
```

Hacer cuadrada la imagen

Indica cómo rellenar su etiqueta contenedora (se **recorta** para encajar)

Redondea la imagen

Herramienta para seleccionar color

- <https://color.adobe.com/es/create/color-wheel>

