




Interacción con Dispositivos Móviles

 Prof. Jonathan Cruz

 jcruz@emae.edu.ar



Clase 02

- Hoja de estilos CSS
- Cascada
- Especificidad
- Modelo de Cajas
- Overflow
- Unidades de medidas

”

Estilos CSS





Incluir css en el HTML

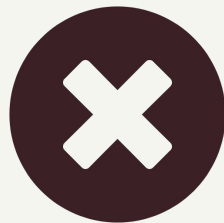
Incluir CSS en el HTML

Existen tres formas de incluir css en nuestro proyecto, cada una son recomendables para ciertos casos

1

En la etiqueta con la atributo style

```
<h1 style="color: blue;">Programación Web</h1>
```



Incluir CSS en el HTML

2

En el head del HTML

```
<head>
  <meta charset="UTF-8">
  <title>Programación Web</title>
  <style>
    h1 {
      color: red;
    }
  </style>
</head>
```



Esta forma es un poco mejor que en la etiqueta del HTML

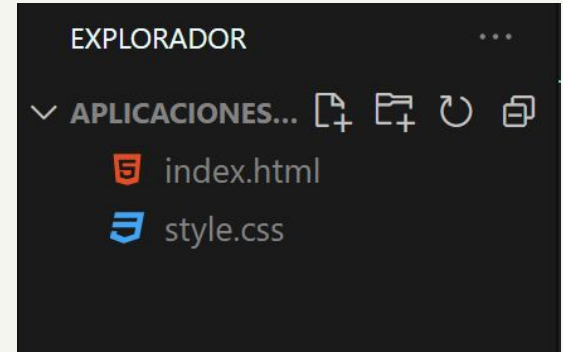
Incluir CSS en el HTML

3

Como un archivo externo vinculando al html con la etiqueta link



```
<head>
  <meta charset="UTF-8">
  <title>Programación Web</title>
  <link rel="stylesheet" href="style.css">
</head>
```




Esta forma es la más recomendada y es la que vamos a usar en la materia

Incluyendo CSS

En este curso vamos a utilizar la tercera forma de incluir css en nuestro proyecto. De esta manera separamos más las capas de lenguajes web

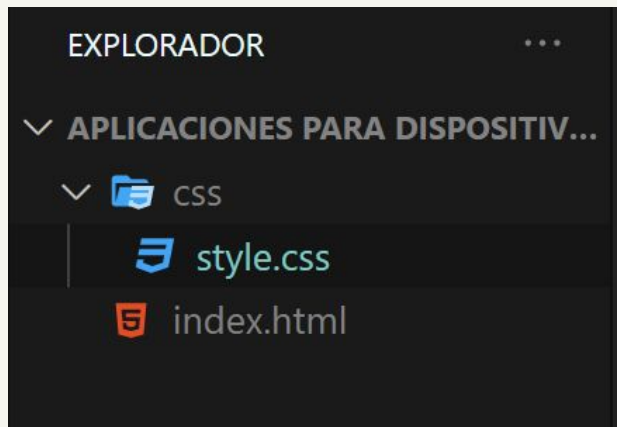
```
<link rel="stylesheet" href="css/estilos.css">
```



Ruta: Relativa o absoluta

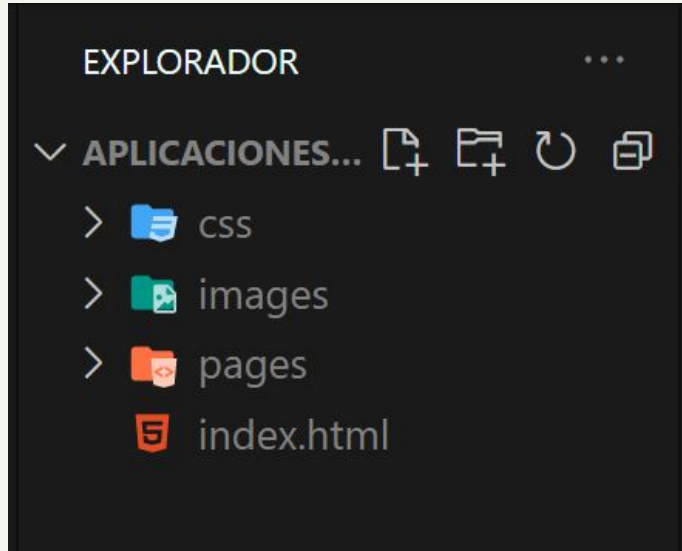
Es importante prestar especial atención a la ruta del archivo **css**

Incluyendo CSS



- Crear archivo un archivo **index.html**
- Crear una carpeta **css**
- Crear un archivo **estilos.css**
- Vincular el **estilos.css** en el **index.html**

Estructura de un Proyecto Web



- Estructura típica de un proyecto web
- Permite una organización más legible de los archivos

Sintaxis CSS

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

La sintaxis de **css** es: Selector luego { } dentro cada propiedad, al finalizar cada una con un ;

```
p {  
    color: blue;  
    font-size: 20px;  
}
```




Esta porción de código **css** selecciona las etiquetas **p** y aplica estilos de color azul y tamaño de fuente de 20 pixeles.

"Alt" + 123: signo de apertura, {.
"Alt" + 125: signo de cierre, }.



Selectores CSS

Selectores

- Por etiqueta 
- Por clase
Anteponiendo un punto . 
- Por id
Anteponiendo el # 

```
h1 {  
  color: red;  
}  
  
.clase {  
  color: red;  
}  
  
#id {  
  color: red;  
}
```

Seleccionar elementos HTML

p a



```
<p>  
    Texto y un <a href="pagina.html">enlace</a>  
</p>
```

p.rojo



```
<p class="rojo">Texto</p>
```

p a.azul



```
<p> Texto y un  
    <a href="pagina.html" class="azul">enlace</a>  
</p>
```

Nombres de las Clases e IDS

Con **camelCase** es más fácil leer las palabras compuestas

claseEmpleado

textoRojo

imagenPrincipalGrande



No se admiten caracteres de espacios en los nombres
No se admiten caracteres especiales

Nombres de las Clases e IDS

Otra forma de de nombrar las clases es con
`snake_case`

`titulo_principal`

`texto_rojo`

`imagen_principal_grande`



No se admiten caracteres de espacios en los nombres



Cascada y Especificidad

Cascada

¿Cuál será el color del texto mostrado en el navegador?

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

Los estilos se aplican en cascada

Precedencia

Cuando se aplican reglas distintas al mismo objeto pueden ocurrir dos cosas:

- Si son propiedades distintas, se suman (se combinan).
- Si tienen alguna propiedad repetida, sólo una queda.

```
#texto{  
    color: purple;  
}  
p {  
    color: red;  
    font-weight: 700;  
}  
.verde {  
    color: green;  
}
```

Precedencia

```
#texto{  
  color: purple;  
}  
p {  
  color: red;  
  font-weight: 700;  
}  
.verde {  
  color: green;  
}
```

- Id sobrescribe cualquier regla
- Class sobrescribe las reglas de las etiquetas
- Etiqueta es de menor precedencia
- Estilos en línea sobre escribe los estilos de los archivos css. ¡No recomendado!

```
<p style="color: blue;"> texto </p>
```

Id > class > etiqueta

Precedencia

!important

Esta declaración corta la precedencia. Se escribe luego de la propiedad CSS.

No es recomendable su uso ya que esto genera un código desorganizado y difícil de mantener

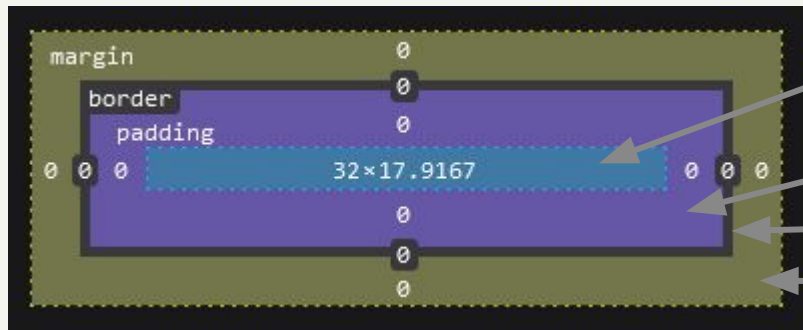
```
#texto{  
    color: purple;  
}  
  
p {  
    color: red;  
    font-weight: 700;  
}  
  
.verde {  
    color: green !important;  
}
```



BOX MODEL

Box Model

Podríamos ver todos los **elemento** HTML como una **caja**, cada elemento de “la caja” tendrán propiedades comunes



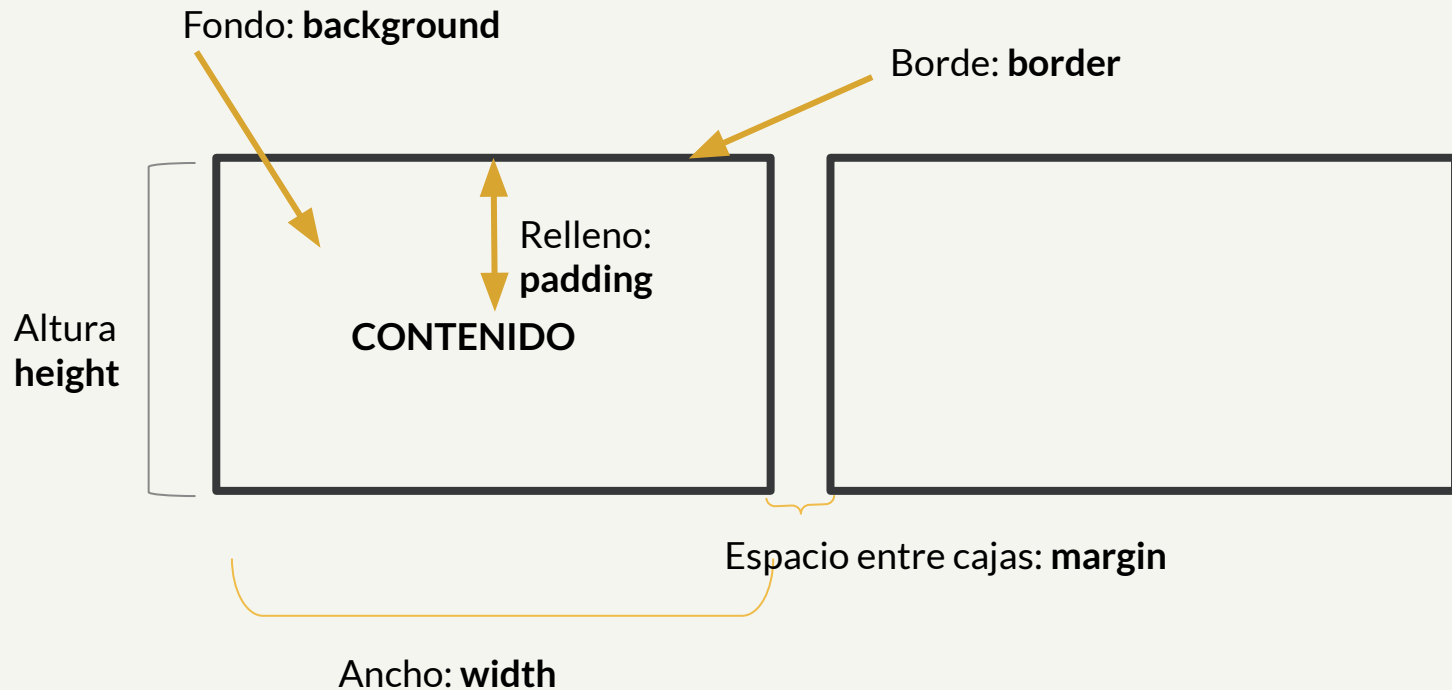
CONTENT: Es el contenido del elemento.

PADDING: Espacio interior

BORDER:

MARGIN: Separación entre el borde y los demás elementos

Propiedades de la caja



Propiedades de la caja

```
div {  
  background-color: blueviolet;  
  color: white;  
  text-align: center;  
  height: 200px;  
  width: 200px;  
}
```

```
<div>Contenido</div>
```





Bloques y de líneas

Box Model

Podríamos ver todos los **elementos** HTML como una **caja** sea un **<h1>**, ****, etc. Cada elemento puede ser de **bloque** o de **línea**



Bloque: Se visualiza uno debajo de otro.

Línea: Se visualiza uno al lado del otro

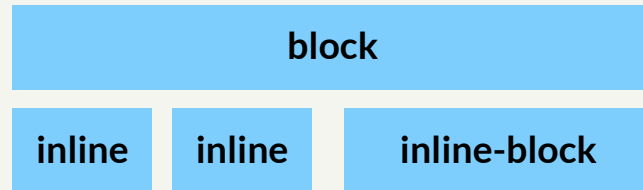


Tipos de elementos

El estándar HTML clasifica a todos sus elementos en dos grandes grupos: elementos **en línea (inline)** y **de bloque (block)**.

Los **elementos de bloque** siempre empiezan en una nueva línea, y ocupan todo el espacio disponible hasta el final de la misma (100%).

Por otra parte, **los elementos en línea** no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.



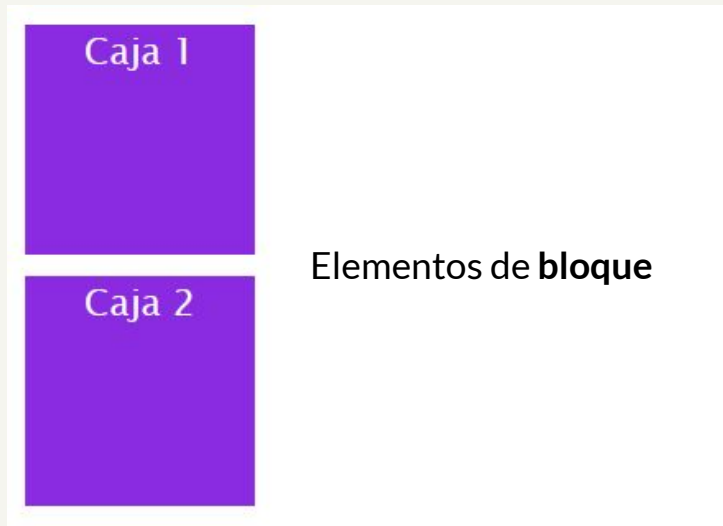
Propiedades de las cajas

Los elementos creados con la etiqueta **div** son elementos de bloque. Si creo dos **div** se visualizan uno debajo de otro

```
<div> Caja 1 </div>
```

```
<div> Caja 2 </div>
```

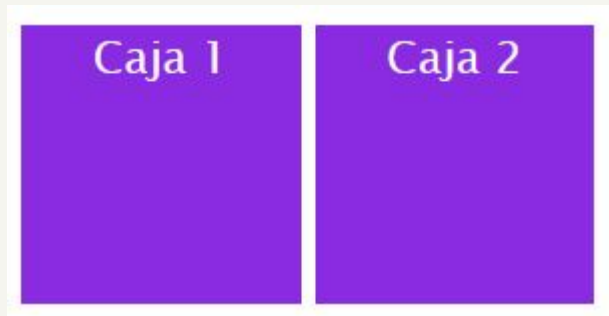
```
div {  
  background-color: blueviolet;  
  color: white;  
  text-align: center;  
  height: 200px;  
  width: 200px;  
  margin-top: 10px;  
}
```



Propiedades de las cajas

Con la propiedad **display** cambiamos la ubicación de los elementos

```
div {  
  background-color: blueviolet;  
  color: white;  
  text-align: center;  
  height: 200px;  
  width: 200px;  
  margin-top: 10px;  
  display: inline-block;  
}
```



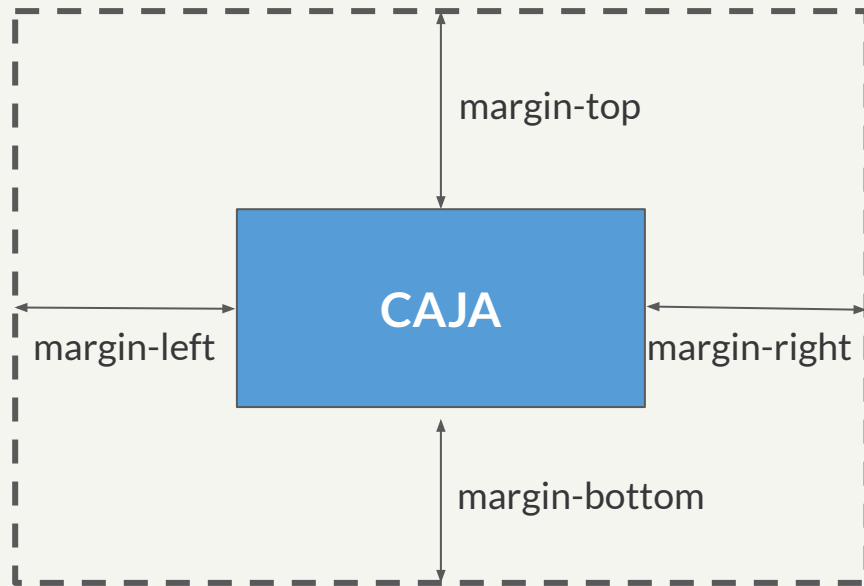
Elementos de línea



Márgenes

Márgenes

Por medio de las propiedades **margin-top**, **margin-right**, **margin-bottom** y **margin-left** podemos aplicar márgenes (**margin**) en cada de la caja



Márgenes

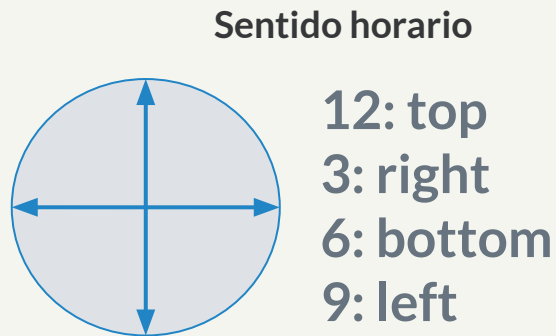
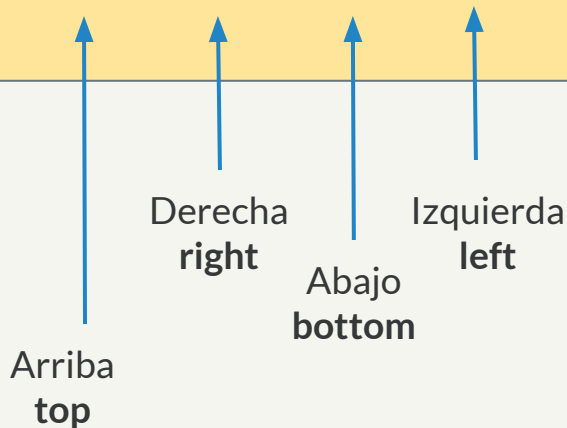
```
div {  
  background-color: blueviolet;  
  color: white;  
  text-align: center;  
  height: 80px;  
  width: 80px;  
  display: inline-block;  
  margin-top: 10px;  
  margin-left: 15px;  
  margin-bottom: 20px;  
  margin-right: 25px;  
}
```



Márgenes

Existe una forma abreviada de colocar los márgenes

```
selector {  
  margin: 10px 20px 15px 7px;  
}
```

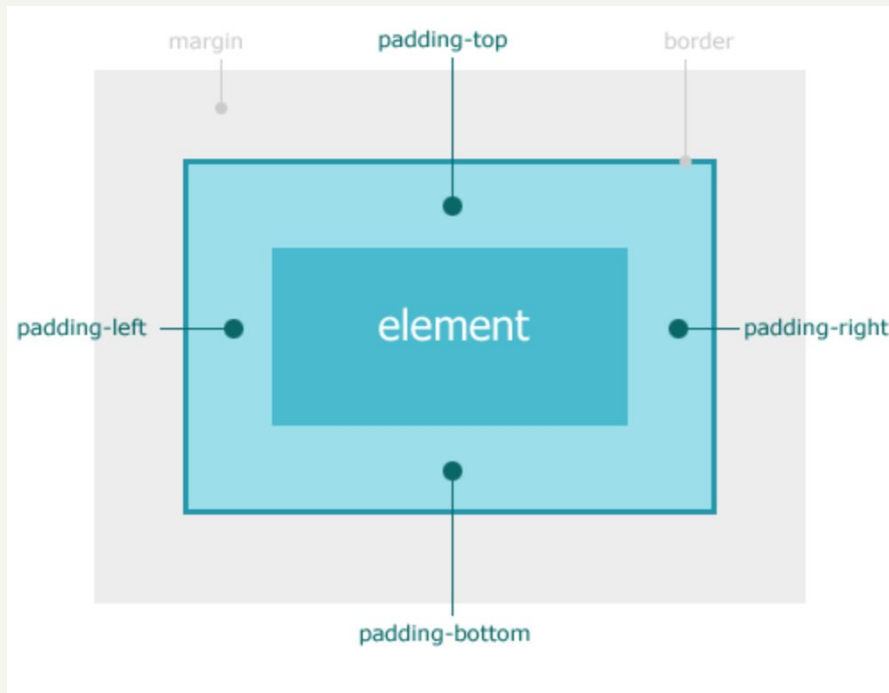




Relleno

Relleno (Padding)

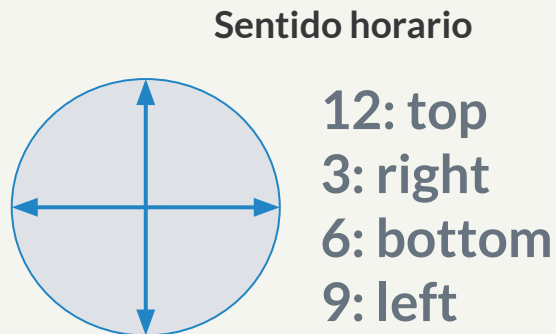
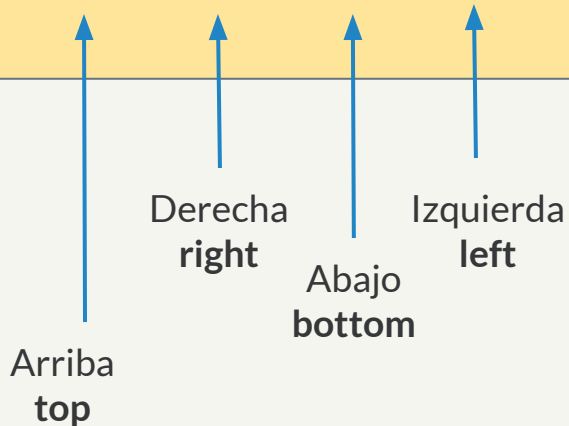
Por medio de las propiedades **padding-top**, **padding-right**, **padding-bottom** y **padding-left** podemos aplicar relleno (**padding**) en cada de la caja



Relleno (Padding)

Existe una forma abreviada de colocar relleno

```
selector {  
  padding: 10px 20px 15px 7px;  
}
```

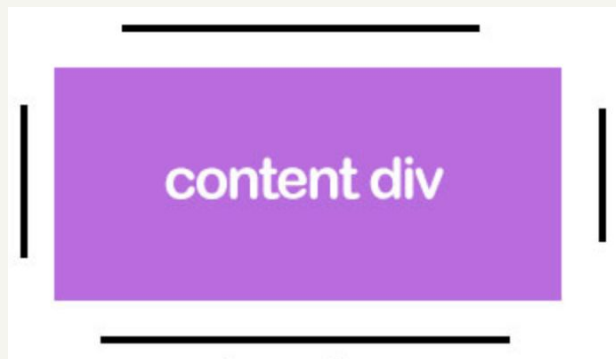




Bordes

Bordes (Border)

Por medio de las propiedades **border-top**, **border-right**, **border-bottom** y **border-left** podemos aplicar bordes (**border**) en cada lado de la caja



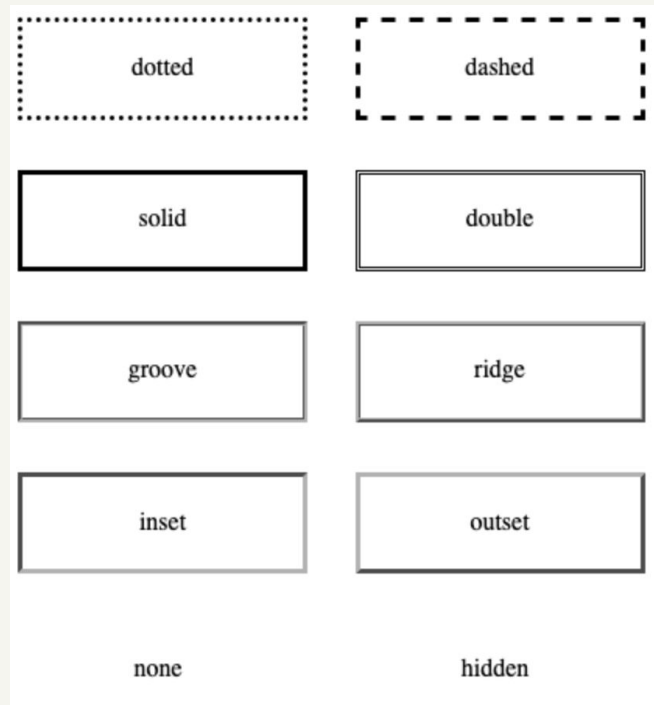
https://www.w3schools.com/css/css_border.asp

Bordes (Border)

Los bordes están formados por tres valores:

- Tipo de borde
- Grosor
- Color

https://www.w3schools.com/css/css_border.asp



Bordes (Border)

```
div {  
  border-top: 3px solid red;  
  border-right: 3px solid blue;  
  border-bottom: 3px solid yellow;  
  border-left: 3px solid green;  
}
```



Bordes (Border)

```
div {  
  border: 1px solid red;  
}
```



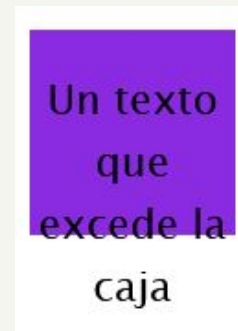


Overflow

Overflow

```
div {  
  background-color: blueviolet;  
  text-align: center;  
  height: 80px;  
  width: 80px;  
  margin-top: 10px;  
  display: inline-block;  
}
```

```
<div>  
  <p>  
    Un texto que excede la caja  
  </p>  
</div>
```



Si un elemento tiene un ancho o alto fijos, cualquier contenido que se sobrepase el contenedor de la caja será visible

Overflow

Tiene 4 valores posibles:

- **Visible:** Es el valor por defecto. El excedente es visible.
- **Hidden:** El excedente no se muestra (lo corta).
- **Scroll:** Genera una barra de scroll en los dos ejes (x/y) del elemento, aunque no se necesite.
- **Auto:** Genera el scroll solo en el eje necesario.





Unidades de medidas

Unidades de Medidas

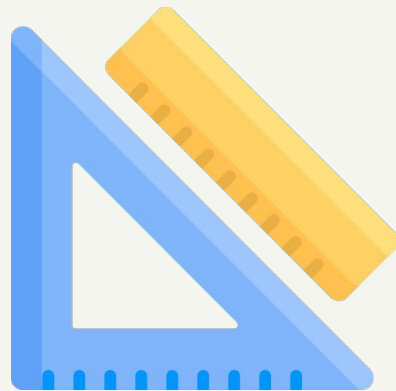
Principalmente se clasifican en **absolutas** y **relativas**

Absolutas

→ **Px** (Píxeles) es una unidad que usan las pantallas

Relativas

- **Rem** relativa a la configuración de tamaño de la raíz (etiqueta html)
- **Porcentaje** tomando en cuenta que 16px es 100%
- **Viewport** se utilizan para layouts responsivos



Responsive Web Design

Es el diseño de la web adaptable a la mayor cantidad de dispositivos:
Laptop, desktop, tablet, celulares, lectores de libros, etc.

Cada dispositivo presenta distintas resoluciones de pantallas
1400x900, 1028x768, 800x600, 320x480, etc.



Responsive Web Design

Básicamente se puede usar medidas relativas

- %: Relativas al elemento padre.
- **em**: Relativas al elemento padre.
- **rem**: Relativas al elemento raíz.
- **vw**: Ancho de la ventana gráfica.
- **vh**: Altura de la ventana gráfica.

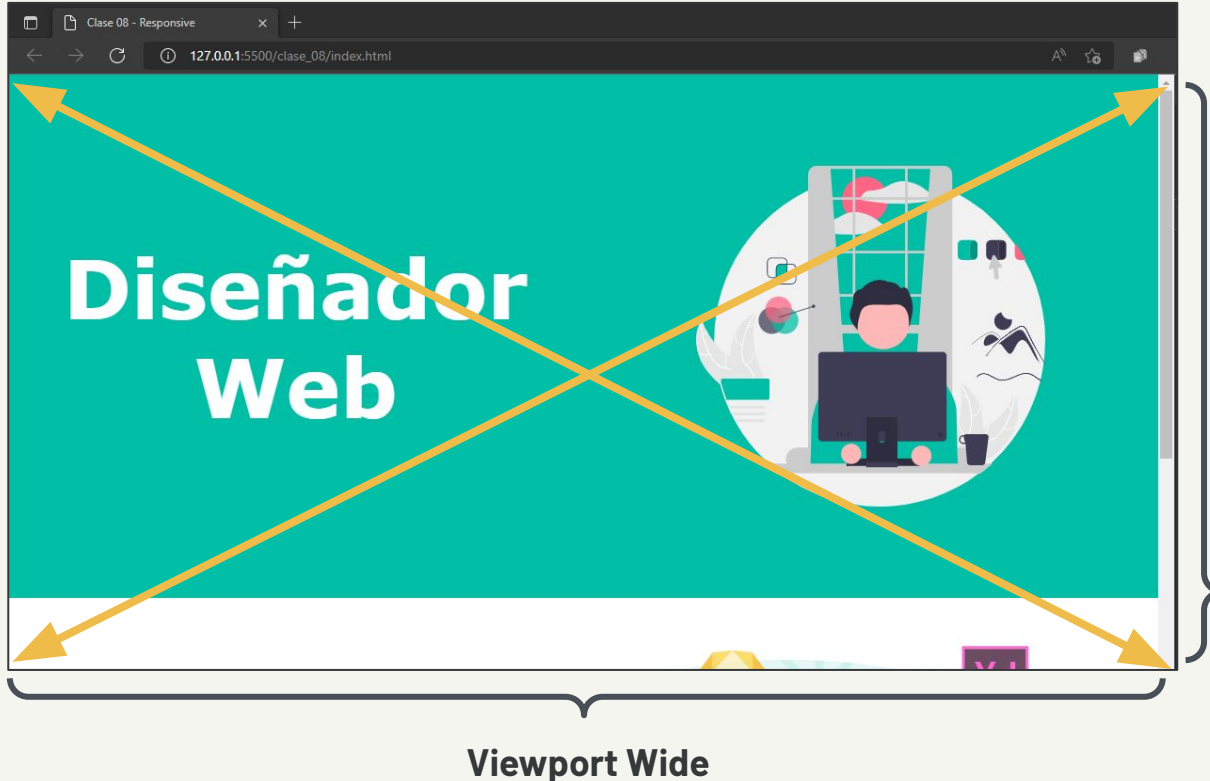


Responsive Web Design

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

Imágenes con medidas
máximas en porcentajes

Viewport



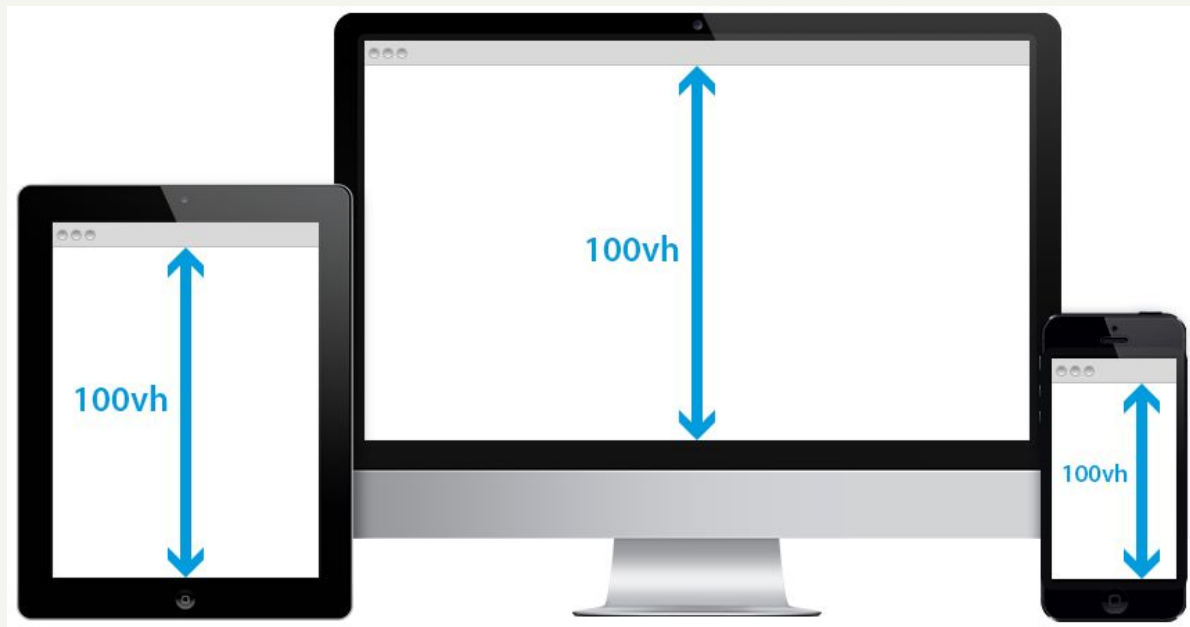
Es el área visible del navegador sin incluir la barra de navegación, favoritos, etc.

Viewport Height

Medidas Relativa

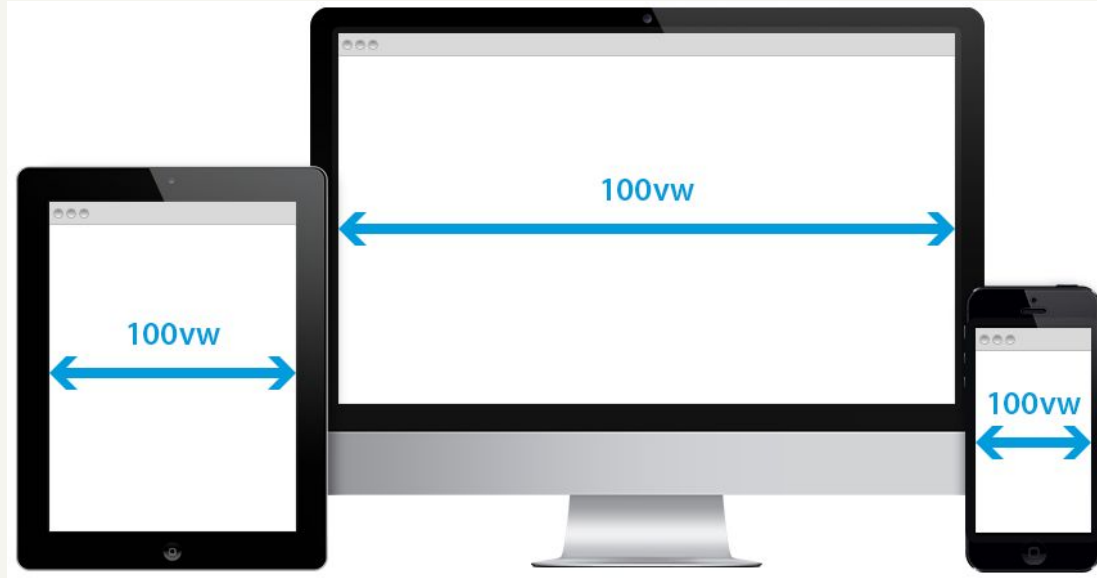
Para poder desarrollar un diseño del layout adaptable a distintos tipo de dispositivos (Pantalla) es **necesario** usar **medidas relativas** al tamaño de **pantalla** o a un **elemento root** (de referencia)

Unidades Viewport



vh (viewport-height) = Altura de la ventana gráfica

Unidades Viewport



vw (viewport-width) = Anchura de la ventana gráfica.

Unidades Viewport

Sus valores se expresan en porcentaje (de 0 a 100)

- **width:100vw** Ancho del 100% de la ventana gráfica
- **height:50vh** Alto del 50% de la ventana gráfica
- **line-height:3vh** Alto de línea del 3% del alto de la ventana gráfica

La diferencia entre las unidades porcentuales expresadas con “%” (width:100%) y las viewport-units (width:100vw) es que las primeras heredan los valores de su elemento padre, y las segundas siempre heredan los valores del bloque inicial o ventana gráfica.

Unidades Viewport

La diferencia entre las unidades porcentuales expresadas con “%” (width:100%) y las unidades del viewport (width:100vw) es que las primeras heredan los valores de su elemento padre, y las segundas **siempre heredan los valores del bloque inicial o ventana gráfica.**

Más información

Les dejamos algunos links para ampliar la clase de hoy:

- https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectors
- <https://web.dev/learn/css/the-cascade/>
- <https://web.dev/learn/css/specificity/>
- <https://www.freecodecamp.org/news/css-box-model-explained-with-examples/>



Muchas Gracias

jcruz@emae.edu.ar