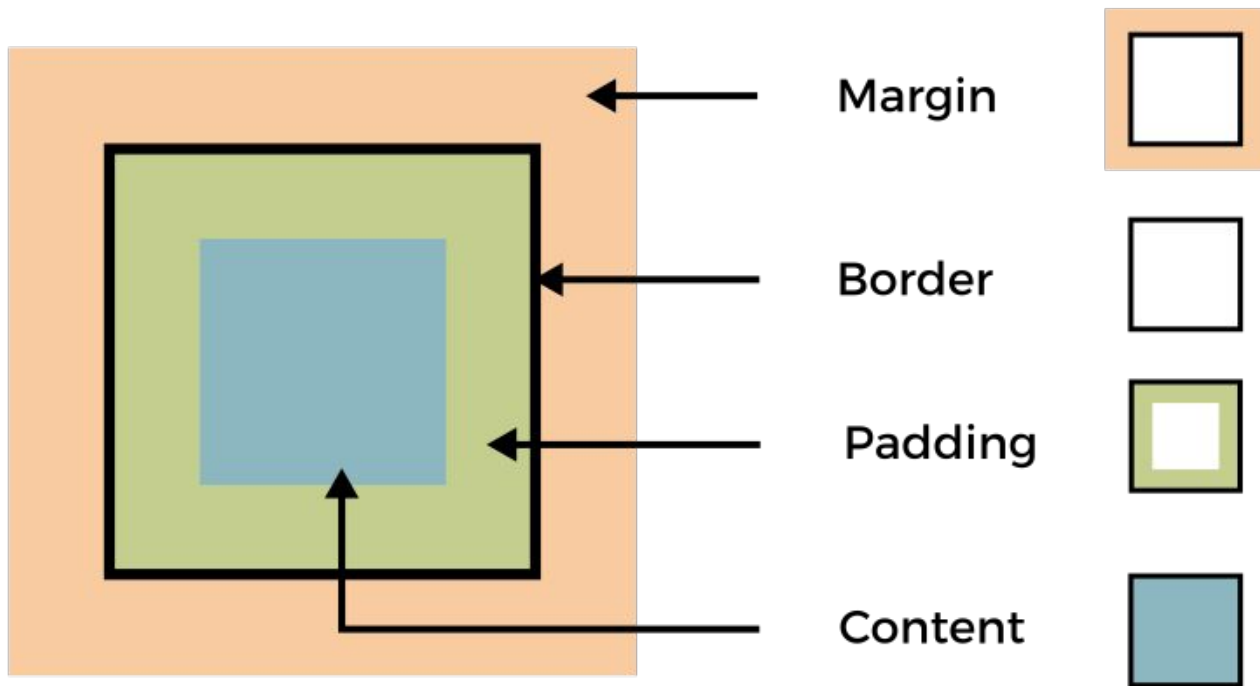


Elementos principales CSS

Edición Web I

Box model



Box model

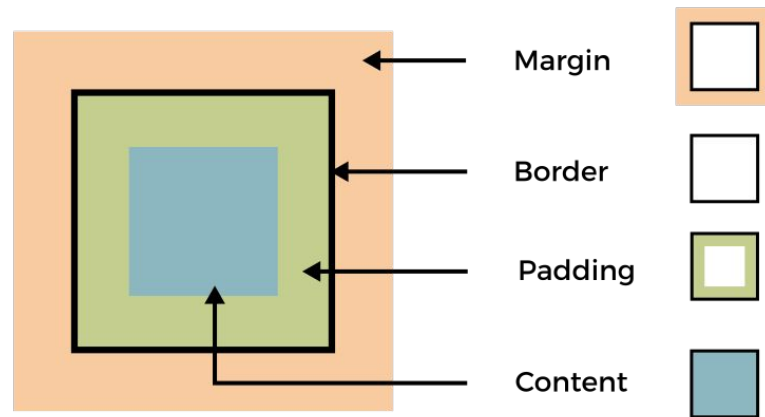
Box model es la representación de un contenedor de información, por ejemplo un **DIV**.

Content: el contenido, un párrafo, una imagen...

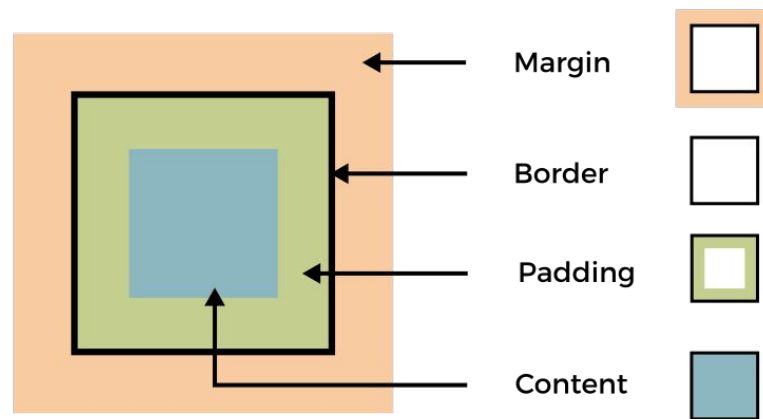
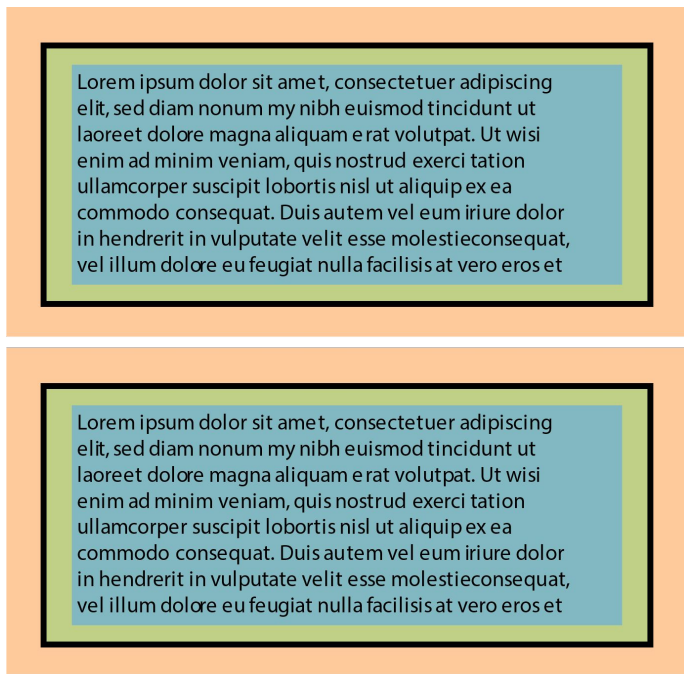
Padding: el espaciado entre el contenido y el límite del contenedor.

Margin: el espaciado del contenedor con otro contenedor.

Border: el borde del contenedor.



Box model



Padding

Podemos definir el padding con el mismo valor para todos los lados, también podemos definir un valor diferente en cada lado en una única fila de código.

El orden que sigue para definir cada lado es el de las agujas del reloj.

Definir el padding de cada lado de forma independiente.

Padding-top: 6px;

Padding-bottom: 6px;

Padding-left: 6px;

Padding-right: 6px;

```
/* Apply to all four sides */
```

```
padding: 1em;
```

```
/* top and bottom | left and right */
```

```
padding: 5% 10%;
```

```
/* top | left and right | bottom */
```

```
padding: 1em 2em 2em;
```

```
/* top | right | bottom | left */
```

```
padding: 5px 1em 0 2em;
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/padding>

Margin

Podemos definir el margin con el mismo valor para todos los lados, también podemos definir un valor diferente en cada lado en una única fila de código.

El orden que sigue para definir cada lado es el de las agujas del reloj.

Definir el margin de cada lado de forma independiente.

```
margin-top: 6px;  
margin-bottom: 6px;  
margin-left: 6px;  
margin-right: 6px;
```

```
/* Apply to all four sides */
```

```
margin: 1em;
```

```
margin: -3px;
```

```
/* top and bottom | left and right */
```

```
margin: 5% auto;
```

```
/* top | left and right | bottom */
```

```
margin: 1em auto 2em;
```

```
/* top | right | bottom | left */
```

```
margin: 2px 1em 0 auto;
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/margin>

Border

Border tiene 3 propiedades:

Color: ***border-color***

```
/* <color> values */
border-color: red;

/* top and bottom | left and right */
border-color: red #f015ca;

/* top | left and right | bottom */
border-color: red rgb(240 30 50 / 70%) green;

/* top | right | bottom | left */
border-color: red yellow green blue;
```

Estilo: ***border-style***

```
/* Keyword values */
border-style: none;
border-style: hidden;
border-style: dotted; /* top and bottom | left and right */
border-style: dashed; border-style: dotted solid;
border-style: solid;
border-style: double; /* top | left and right | bottom */
border-style: groove; border-style: hidden double dashed;
border-style: ridge;
border-style: inset; /* top | right | bottom | left */
border-style: outset; border-style: none solid dotted dashed
```

Grosor: ***border-width***

```
/* Keyword values */
border-width: thin;
border-width: medium;
border-width: thick;

/* <length> values */
border-width: 4px;
border-width: 1.2rem;

/* top and bottom | left and right */
border-width: 2px 1.5em;

/* top | left and right | bottom */
border-width: 1px 2em 1.5cm;

/* top | right | bottom | left */
border-width: 1px 2em 0 4rem;
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/border>

Border

Podemos definir las 3 propiedades (*color*, *style* y *width*) En una misma línea de código.

Pero solo podremos poner un único valor por propiedad.

Podemos definir las propiedades en el orden que queramos.

!!! Muy importante: si no definimos alguno de los 3 valores (color, style o width) el borde no se visualizará.



```
border: 1px solid blue;
```



```
border: 1px 4px solid dotted blue red
```

Border-radius

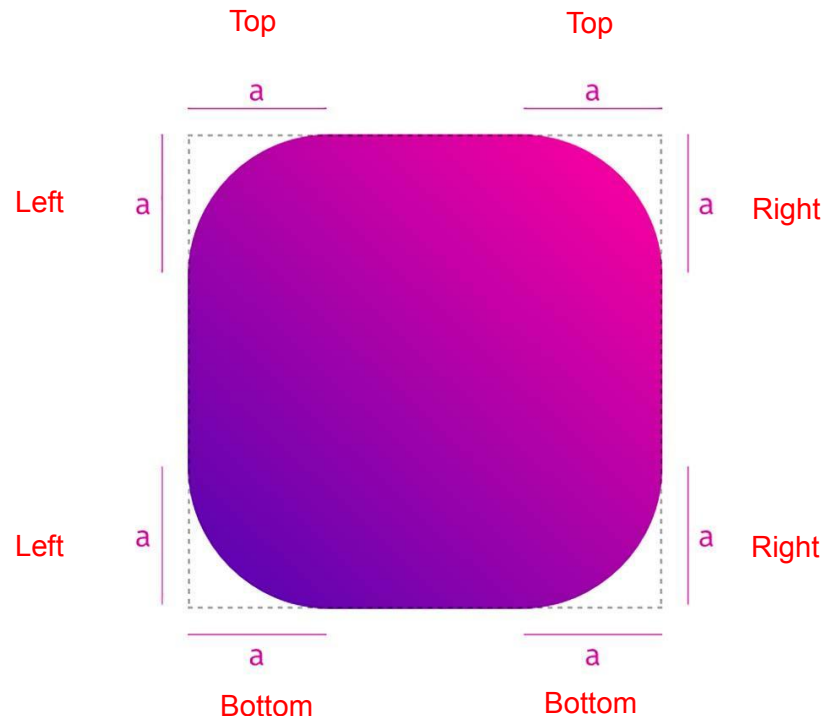
Definimos el radio de las esquinas del borde:
border-radius: 25px;

Podemos definir el borde de cada esquina de forma independiente:

```
border-top-left-radius: 25px;  
border-top-right-radius: 25px;  
border-bottom-left-radius: 25px;  
border-bottom-right-radius: 25px;
```

Definimos independiente las 2 caras de una esquina:

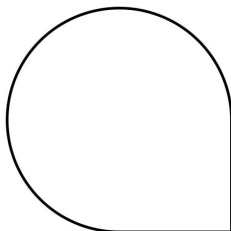
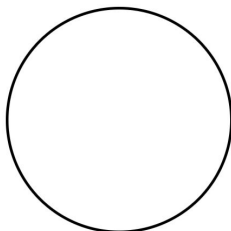
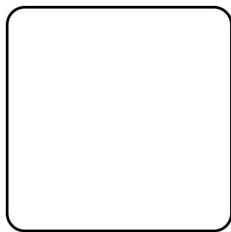
```
border-top-left-radius: 25px 80px;  
border-top-right-radius: 25px 80px;  
border-bottom-left-radius: 25px 80px;  
border-bottom-right-radius: 25px 80px;
```



Ejemplos Border-radius

Podemos definir las 4 esquinas en una misma línea de código exactamente igual que con la propiedad **border**

!!!
Aplicar border-radius a una etiqueta ****



```
border-radius:25px;
```

```
border-radius:500px;
```

```
border-radius:500px 500px 0px 500px;
```

```

```

Ejemplos

Border-radius



A la imagen aparte de un **border-radius** le hemos aplicado un **borde punteado** y un **padding**

```

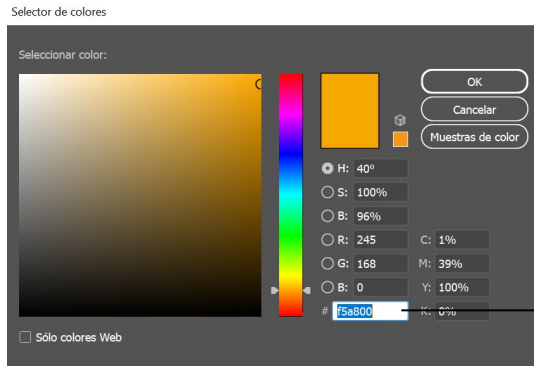
```

Color / background-color

Hay 4 formas de definir el color:

1- escribiendo el nombre del color, hay 140 nombres de color definidos en css ([ir a listado colores](#)).

2- escribiendo el valor hexadecimal, antes de escribir los 6 dígitos que forman este valor pondremos el símbolo de #



Valor
hexadecimal

```
.class{color: aquamarine;}
```

```
.class{color: #ff0000;}
```

Color / background-color

3- escribiendo los valores **RGB**. Estos valores irán entre paréntesis y separados por comas.

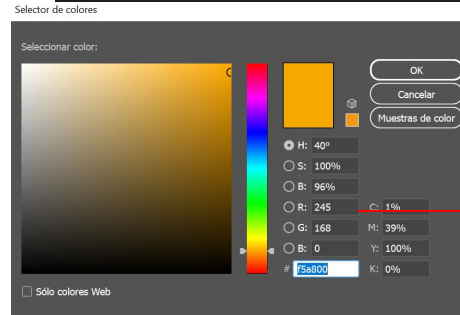
4- escribiendo los valores **RGBA**. Estos valores irán entre paréntesis y separados por comas, el último valor define el canal **Alpha** (la transparencia).

Los decimales del valor Alpha se definen con el símbolo del punto:

0	→	Opacidad=0%
0.5	→	Opacidad=50%
1	→	Opacidad=100%

```
.class{color:  rgb(245, 168, 0);}
```

```
.class{color:  rgba(245, 168, 0, 0.5);}
```



Valores
RGB

Unidades

Pixels- valor determinado en pixels.

EM- una unidad EM es equivalente a 16px

Porcentaje- valor determinado en tanto%

Viewport- el valor lo determina el área visible de la pantalla: 100 es el valor máximo 0 el mínimo.

Vh: *viewport height*

Vw: *viewport width*

Herencia:

% es la única unidad que hereda el valor del contenedor padre: si tenemos un contenedor padre del 80% y luego un hijo de 50%... Este 50% es respecto al 80% del padre y no al total.

El resto de unidades son independientes al valor del padre.

```
<div style="width:80%">
  <div style="width:600px">
    Contenido
  </div>
</div>
```

```
<div style="width:80%">
  <div style="width:50vw">
    Contenido
  </div>
</div>
```

Width / Height

Width: Definimos la anchura de un elemento.

Height: Definimos la altura de un elemento.

width:800px;

El contenedor siempre ocupa el tamaño asignado independientemente de lo que ocupe el contenido.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed diam nonummy nibh
euismod tincidunt ut laoreet

```
<div style="width:100%;">  
  <div style="width:50%;">contenido</div>  
  <div style="width:50%">contenido</div>  
</div>
```

```
<div>  
  <div style="height:80vh;">contenido</div>  
  <div style="height:20vh">contenido</div>  
</div>
```

Max-width / Min-width

Max-width: definimos el ancho máximo del contenedor, una vez lleguemos al ancho el contenedor no seguirá creciendo.

Si el contenido ocupa 200px el contenedor será de 200px... pero si el contenido tiene 800px de ancho y tenemos un max-width de 600 en el contenedor... una vez lleguemos a 600px de ancho hará un salto de línea.

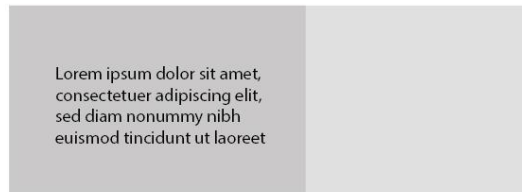
Min-width: Definimos la anchura mínima.

Max-height / Min-height

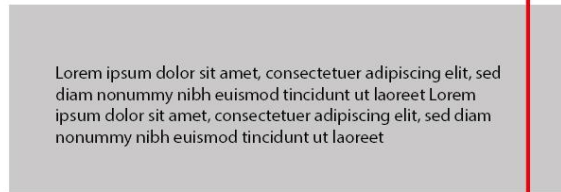
Tiene el mismo funcionamiento que el width

max-width:800px;

El contenedor ocupará el espacio que ocupa el contenido como máximo llegará a 800px, si supera este ancho hará un salto de línea.



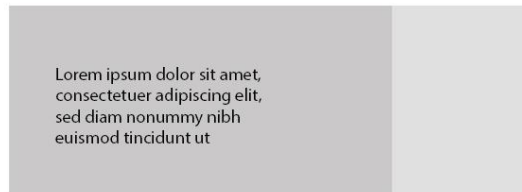
800px



No rebasará el máximo asignado

min-width:600px;

El contenedor ocupará como mínimo 600px aunque el contenido ocupe menos, si el contenido supera el ancho mínimo de 600px en este caso seguirá creciendo.



Espaciado	Bordes	Color / Color fondo	Unidades	Tamaño
Padding: Espacio entre el contenido y el contenedor Margin: Espacio entre contenedores.	border: border-style border-width border-color border-radius	color: color; #ff000; rgb(15,26,18); rgba(15,26,18,0.6); background-color: color; #ff000; rgb(15,26,18); rgba(15,26,18,0.6);	Pixels - Px Porcentaje - % Em Viewport: Width vw Height vh	Tamaños fijos: width Height Tamaños relativos: Max-width Max-height Min-width Min-height
			Texto	
			font-size: font-size:16px; text-align: Left Right Center Justify	

Ejercicio clase

Vamos a hacer una landing page similar al ejemplo de la imagen, utilizaremos las siguientes propiedades css:

1. Padding
2. Text-align
3. Background-color
4. Color
5. Border
6. Border-radius

#f1ed8e
R:241 G:237 B:142

#bcd13e
R:188 G:209 B:62

#9dc656
R:157 G:198 B:86

Diseño multimedia

Si estás interesado en matricularte en nuestro Ciclo Formativo de Producto Gráfico Interactivo y explorar el mundo del diseño digital, has llegado al lugar indicado. Aquí te proporcionamos toda la información que necesitas para adentrarte en una emocionante carrera en el campo del diseño interactivo.

Durante este ciclo formativo tendrás la oportunidad de explorar una amplia gama de habilidades y conocimientos que te permitirán comenzar tu carrera en el mundo del diseño interactivo. Desde el diseño de interfaces de usuario intuitivas hasta la creación de contenido multimedia dinámico, aprenderás a utilizar las herramientas y técnicas necesarias para desarrollar experiencias digitales innovadoras y atractivas.

Además de familiarizarte con el uso de software especializado y tecnologías emergentes, explorarás conceptos fundamentales del diseño web y la comunicación visual en el entorno digital. A través de proyectos prácticos y colaborativos, podrás experimentar y descubrir tu propio enfoque dentro del campo del diseño gráfico interactivo.



Lorem ipsum dolor sit amet,
consectetuer adipiscing elit,
sed diam Lorem ipsum dolor
sit amet, consectetur
adipiscing elit, sed diam

