



El futuro digital
es de todos

MinTIC



Manejo de CSS

OPERADO POR:



Mision
TIC2022

ruta de aprendizaje 1



Manejo de CSS

Recordemos que los archivos **CSS** son los que permiten que nuestra pagina web tenga un **aspecto visual mas atractivo**, nos permite que tenga los colores y diseños necesarios.



Recordemos que **Margin** es la separación entre una caja y las cajas cercanas a ella, mientras que el **padding** es la separación entre una caja y su borde o límite. Esto es bastante útil para **ubicar cada sección de nuestra pagina como queremos**.





- **Margin:** es la separación entre un elemento y el resto, en otras palabras es el margen del elemento.
- **Padding:** Es el margen interno, es decir es la separación que hay entre el contenido del elemento y el borde del elemento.

Ejemplo 1:

En el siguiente ejemplo tenemos dos **div**, uno de **color verde** con el **id** de **"box-1"** y otro de **color rosa** con el **id** de **"box-2"**.

box-1 tiene **padding** de **50px** y un **margen** de **0 px**, mientras que **box-2** tiene **padding** de **0** pero un **margen** de **200px**.





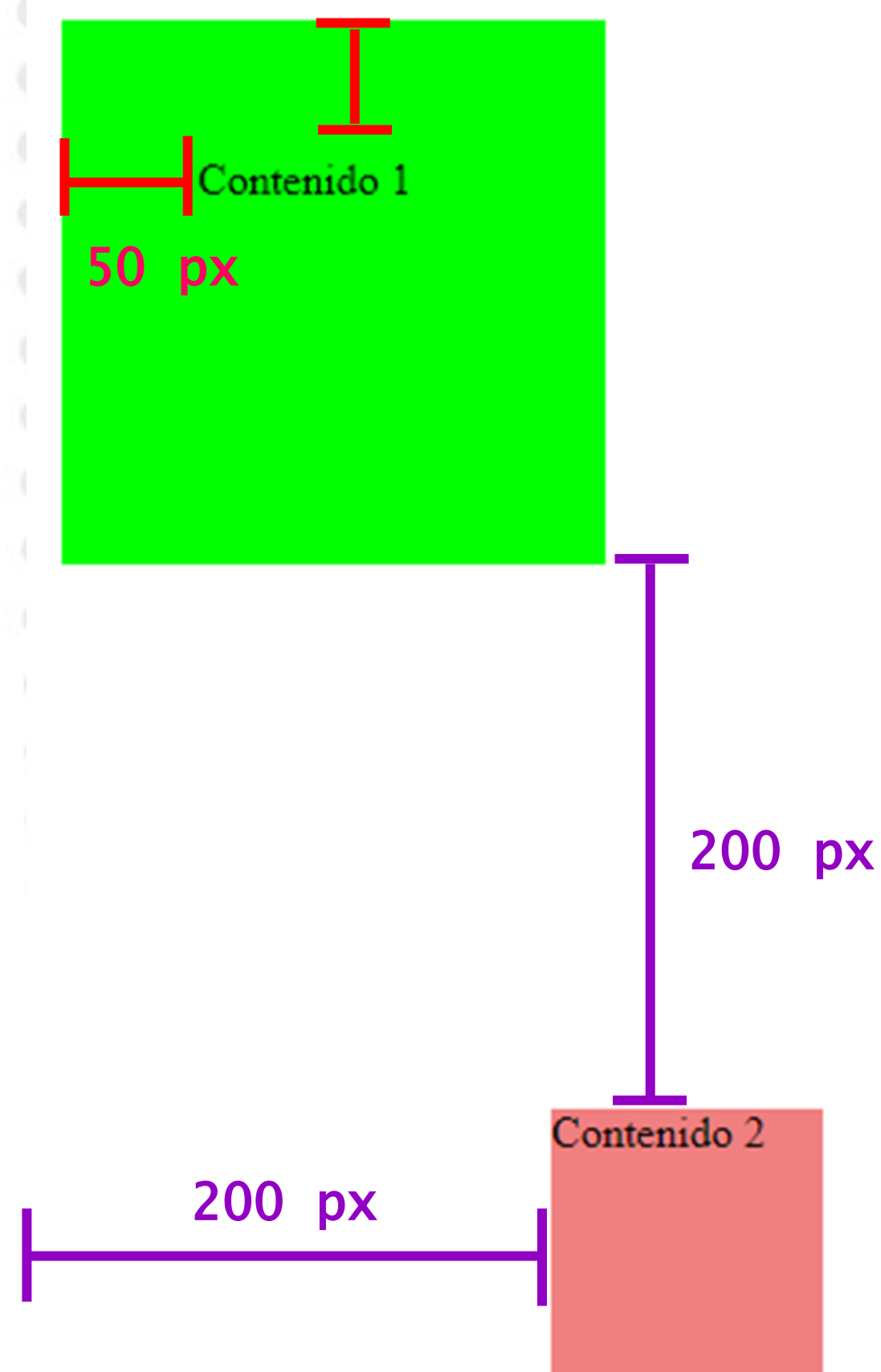
Primero vamos a crear las **características** del cuadro numero 1 ó **box-1**, queremos que sea **verde lima**, y que tenga cierto tamaño.

Con **width: 100px** le estamos dando el ancho al cuadro de un tamaño de 100 pixeles, igual con la instrucción **height: 100px** le indicamos la altura del cuadro.

La instrucción **margin** le va a indicar al recuadro que distancia tiene entre la margen del espacio donde fue creado, y la instrucción **padding** le va a indicar que tan lejos de la margen del cuadrado se ubicara el texto.

```
.box-1{  
  width: 100px;  
  height: 100px;  
  background: lime;  
  padding: 50px;  
  margin: 20px;  
}
```

```
.box-2{  
  margin: 200px;  
  width: 100px;  
  height: 100px;  
  background: lightcoral;  
}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Margin y padding</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="box-1">Contenido 1</div>
  <div class="box-2">Contenido 2</div>

</body>
</html>
```

```
.box-1{
  width: 100px;
  height: 100px;
  background: lime;
  padding: 50px;
  margin: 20px;
}

.box-2{
  margin: 200px;
  width: 100px;
  height: 100px;
  background: lightcoral;
}
```

Estas serian las **instrucciones** que debo usar para indicarle como quiero visualizar las dos figuras, haciendo uso de los **comandos** revisados anteriormente.



Margin y Padding

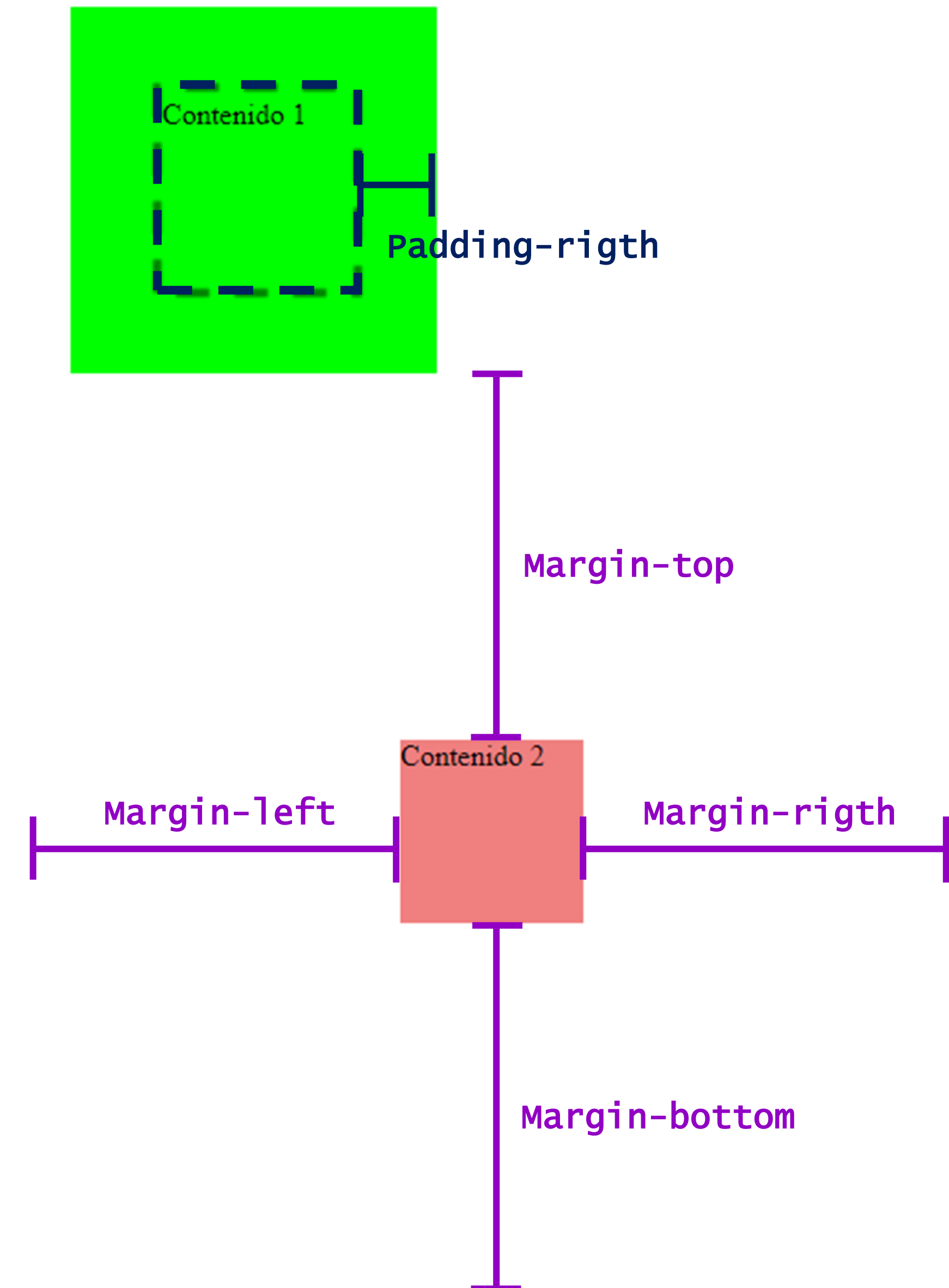
Es importante aclarar que la instrucción margin es un atajo, esta instrucción engloba las instrucciones

- margin-top:
- margin-right:
- margin-left:
- margin-bottom:

De la misma forma tenemos el padding:

- padding-top:
- padding-left:
- padding-right:
- padding-bottom:

Estas instrucciones nos permiten ser más específicos con los valores de margen y padding a cada lado de la caja.





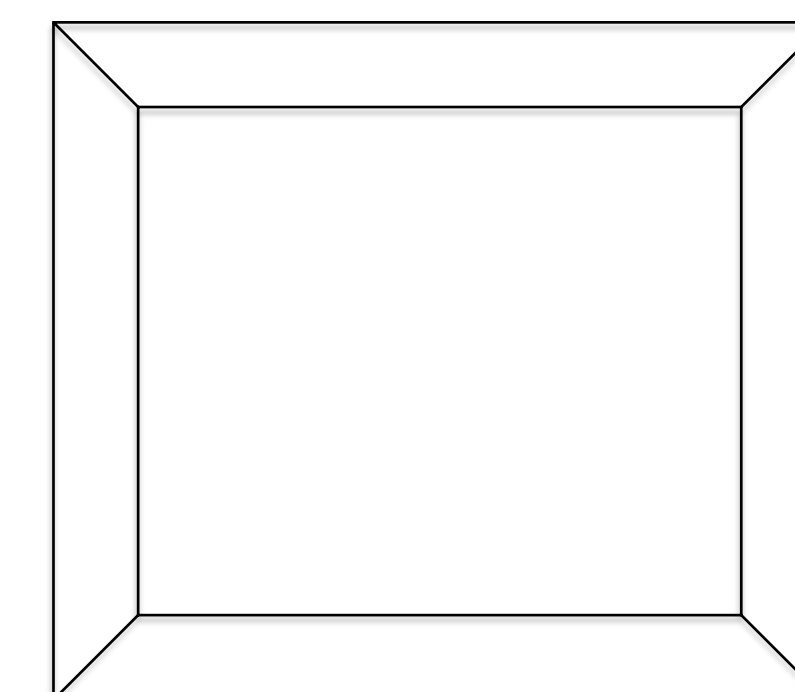
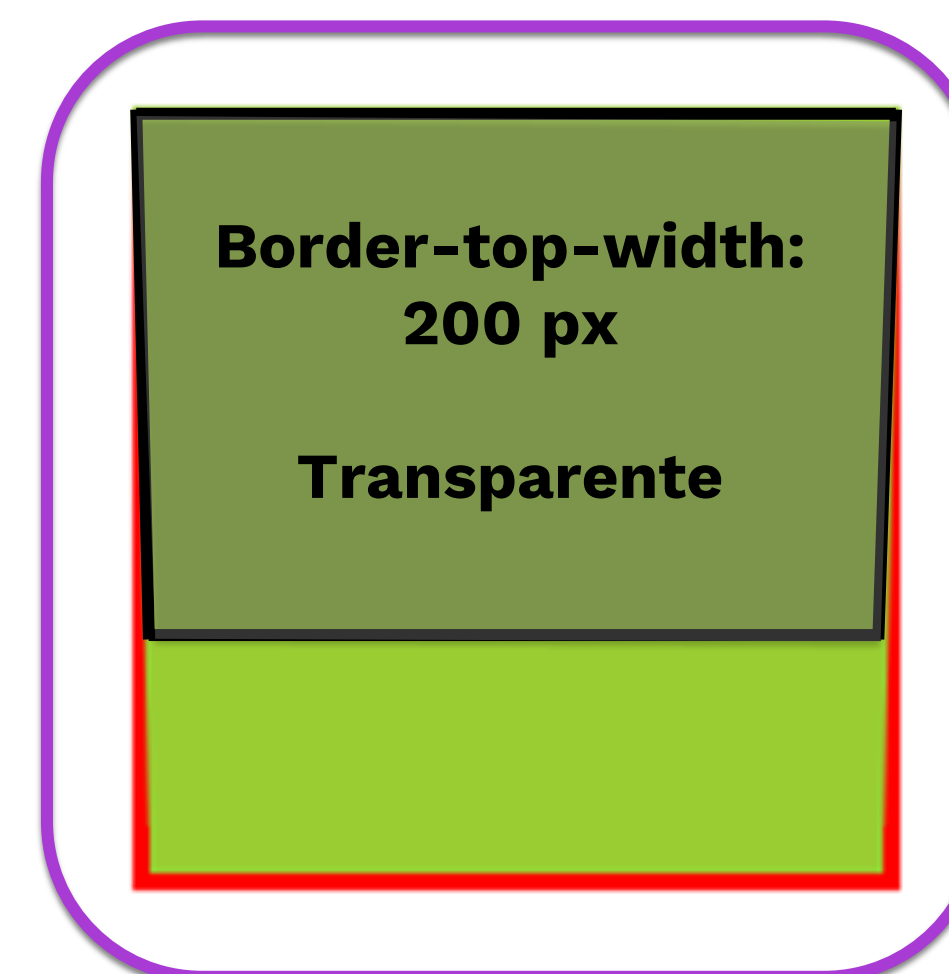
Bordes:

La instrucción border, nos permite especificar las características del borde del elemento.

Entre las características que podemos definir tenemos:

- **border-width:** nos permite indicar el ancho del borde
- **border-style:** nos permite especificar el estilo del borde
- **border-color:** nos permite cambiar el color del borde.

```
[class*=box] {  
    width: 200px;  
    height: 200px;  
    margin: 2em;  
}  
  
.box1 {  
    background: yellowgreen;  
    border: 5px solid red;  
    border-top-width: 200px;  
    border-top-color: transparent;  
}
```



Estructura de bordes

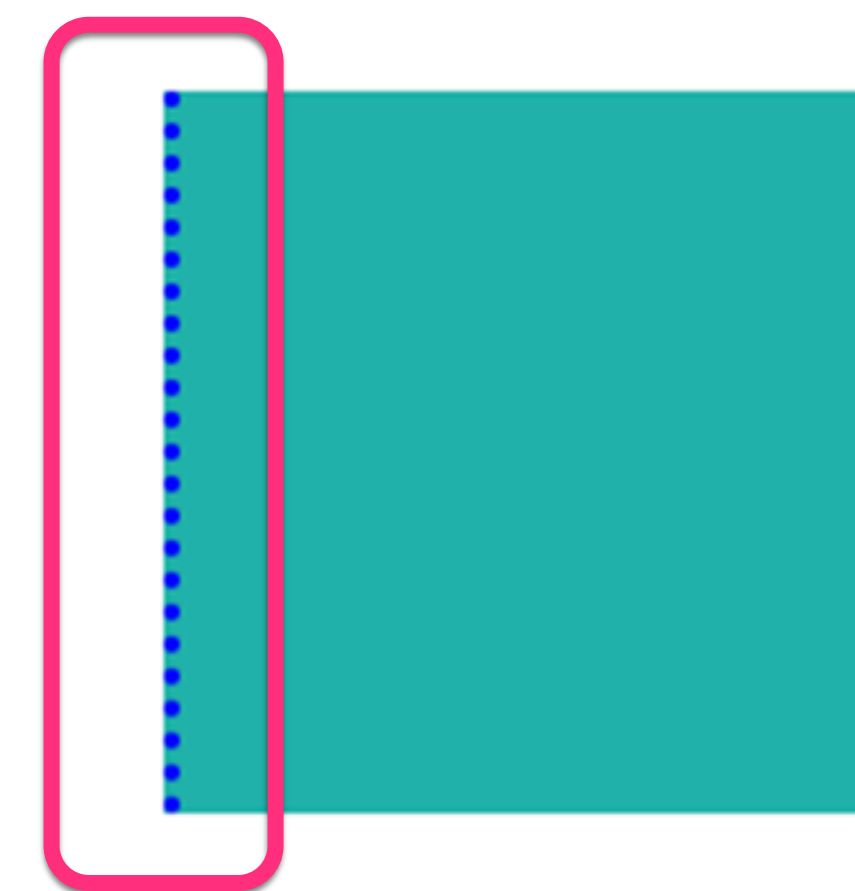


Border es Tambien un atajo ya que englobe las siguientes instrucciones:

- border-top-(width | style | color)
- border-right-(width | style | color)
- border-bottom-(width | style | color)
- border-left-(width | style | color)

En el ejemplo Podemos observar una variedad de cajas con diferentes bordes

```
.box2 {  
    background: lightseagreen;  
    border-left-width: 5px;  
    border-left-style: dotted;  
    border-left-color: blue;  
}
```



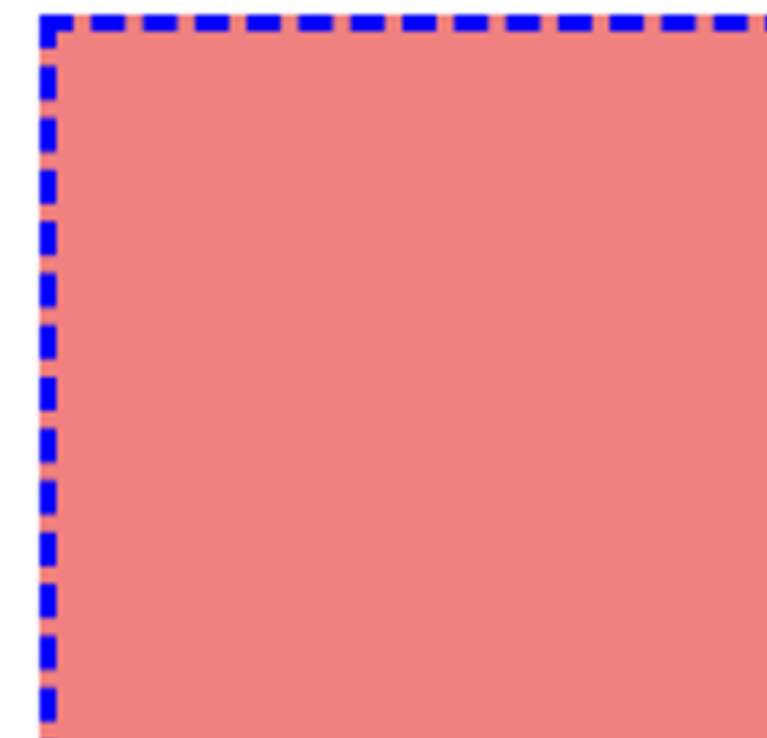


Border es Tambien un atajo ya que englobe las siguientes instrucciones:

- border-top-(width | style | color)
- border-right-(width | style | color)
- border-bottom-(width | style | color)
- border-left-(width | style | color)

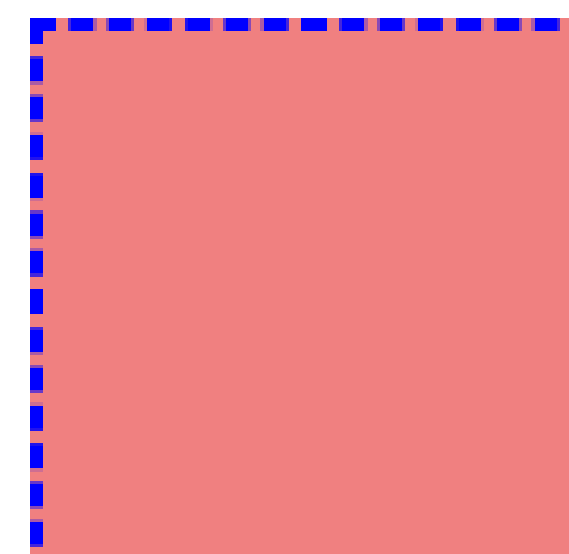
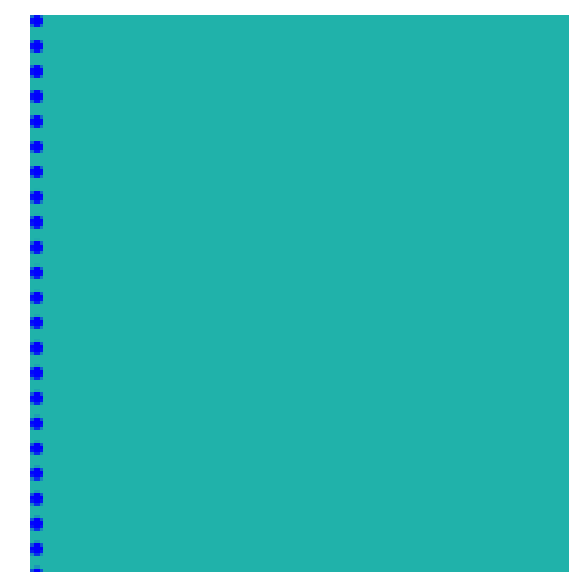
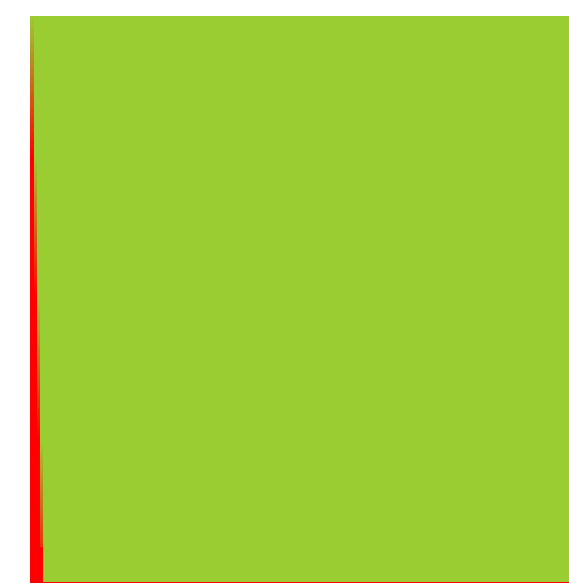
En el ejemplo Podemos observar una variedad de cajas con diferentes bordes

```
.box3 {  
    background: lightcoral;  
    border: 5px dashed blue;  
}
```





Con la estructura que tienen los bordes podemos hacer muchas cosas, podemos generar diferentes figuras y estilos que nos ayudaran a mejorar el aspecto de nuestra pagina web, estos son solo unos ejemplos de como utilizarlos.



```
[class*=box] {  
  width: 200px;  
  height: 200px;  
  margin: 2em;  
}  
  
.box1 {  
  background: yellowgreen;  
  border: 5px solid red;  
  border-top-width: 200px;  
  border-top-color: transparent;  
}  
  
.box2 {  
  background: lightseagreen;  
  border-left-width: 5px;  
  border-left-style: dotted;  
  border-left-color: blue;  
}  
  
.box3 {  
  background: lightcoral;  
  border: 5px dashed blue;  
}
```




Posicionamiento de elementos:

la posición de un elemento en un inicio esta determinada por el flujo. Podemos observar que los elementos se posicionan según el orden en el documento HTML, los elementos se puede posicionar usando el comando **“position”**, una vez usado este comando el elemento puede ser movido en los eje x, y ,z

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Position - Relative</title>
  <link rel="stylesheet" href="normalize.css">
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <!-- <div class="box1">Box 1</div>
       <div class="box2">Box 2</div> -->
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Incidunt exp
licabo architecto commodi accusamus hic? Voluptatem voluptates aliquid excep
turi quod quas, modi harum animi autem ex, vitae ut consequuntur numquam non
.
  </p>
  <div class="box1">Box 1</div>
  <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente, u
nde porro! Sit consectetur delectus cumque, sapiente cum eius provident quam
ad neque quod, eum suscipit ab similique? Saepe, earum quam!</p>
</body>
</html>
```



Lorem ipsum dolor sit amet consectetur adipisicing elit. Incidunt explicabo architecto commodi accusamus hic? Voluptatem voluptates aliquid excepturi quod quas, modi harum animi autem ex, vitae ut consequuntur numquam non.

Box 1

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sapiente, unde porro! Sit consectetur delectus cumque, sapiente cum eius provident quam ad neque quod, eum suscipit ab similique? Saepe, earum quam!

Este sería el resultado que obtendríamos para el ejemplo anterior. Las frases se quedan en la posición que le indicamos, al igual que el cuadrado de **color verde**.





El comando “position” puede tener estos valores:

- position: relative;
- position: absolute;
- position: fixed;

Position relative: posiciona el elemento como punto de referencia en el lugar donde apareció el elemento según el flujo del documento html. Sin embargo usando las instrucciones top, left, right y bottom podemos indicar la distancia que deseamos que se mueva el elemento desde el punto de referencia.

```
.box1, .box2{  
  width: 250px;  
  height: 250px;  
  margin: 1em auto;  
  font-size: 2.5em;  
  text-align: center;  
}  
  
.box1{  
  background: lightgreen;  
  position: relative;  
  
  top: 200px;  
  right: 200px;  
}
```

Hagamos una prueba moviendo el cuadrado de color verde de lugar o posición, vamos a moverlo **200 px** del punto de referencia inicial.





..orem ipsum dolor sit amet consectetur adipisicing elit. Incidunt explicabo architecto commodi accusamus hic? Voluptatem voluptates aliquid excepturi quod quas, modi harum animi autem ex, vitae ut
:onsequuntur numquam non.

Box 1

..orem, ipsum dolor sit amet consectetur adipisicing elit

delectus cumque, sapiente cum eius provident quam ad neque quod, eum suscipit ab similique? Saepe, earum quam!

En el ejemplo podemos observar como ahora el cuadrado **box1** se ha separado del punto de referencia **200px** de del arriba y **200px** de la derecha.

Sin embargo es importante recalcar que los demás elementos siguen manteniendo su posición, lo que quiere decir que aunque movamos un elemento de **posición**, este elemento sigue manteniendo su lugar en el flujo.





Position absolute; el posicionamiento “**absolute**” toma como referencia el elemento posicionado mas cercano, si no hay un elemento posicionado cerca a él, tomara como referencia el de la ventana del navegador, y a diferencia del posicionamiento relativo, el elemento no conserva su lugar en el flujo.

Ejemplo:

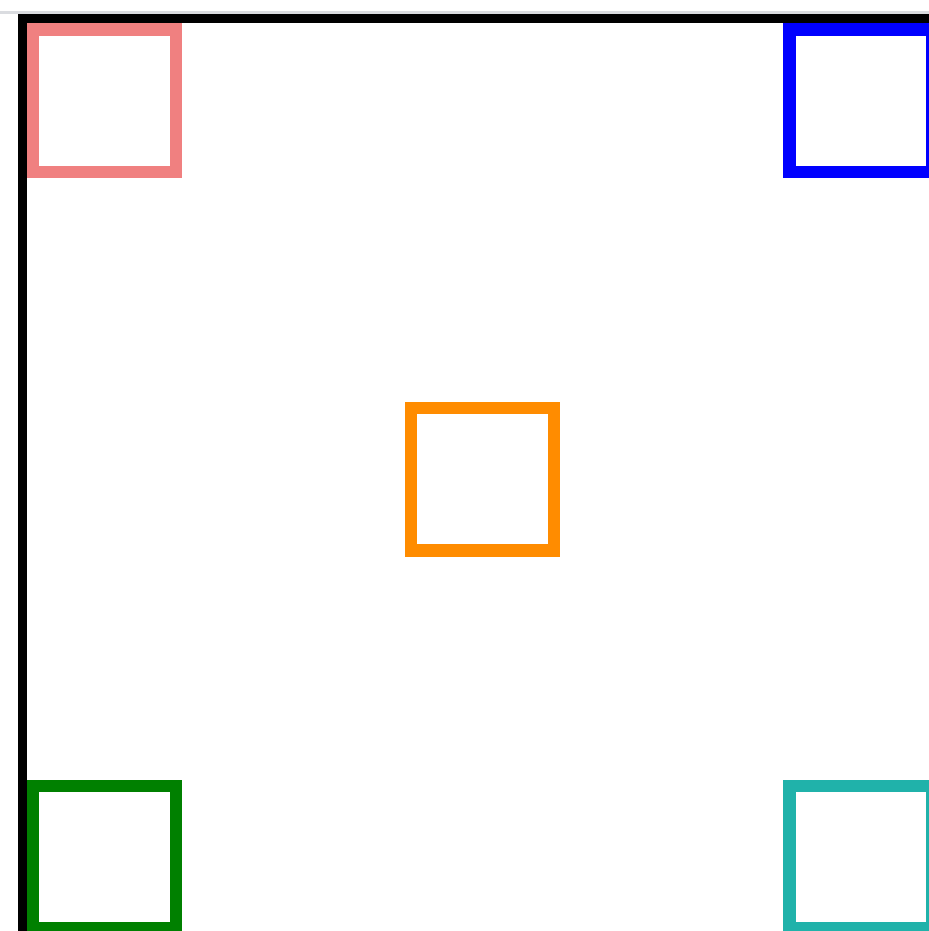
Vamos a generar una **imagen** diferente, vamos a crear **6 cuadros**, con características diferentes, y un contenedor que los contenga a todos.



```
.container {  
  width: 300px;  
  height: 300px;  
  margin-left: auto;  
  margin-right: auto;  
  border: 3px solid black;  
  position: relative;  
}
```

```
[class^=box] {  
  width: 50px;  
  height: 50px;  
}
```

```
.box2 {  
  border: 4px solid blue;  
  position: absolute;  
  top: 0;  
  right: 0;  
}
```



Cuando lo **ejecutamos**, vamos a tener como solución la figura que vemos arriba.

```
.box3 {  
  border: 4px solid green;  
  position: absolute;  
  bottom: 0;  
  left: 0;  
}
```

```
.box4 {  
  border: 4px solid lightseagreen;  
  position: absolute;  
  bottom: 0;  
  right: 0;  
}
```

```
.box5 {  
  border: 4px solid darkorange;  
  position: absolute;
```

