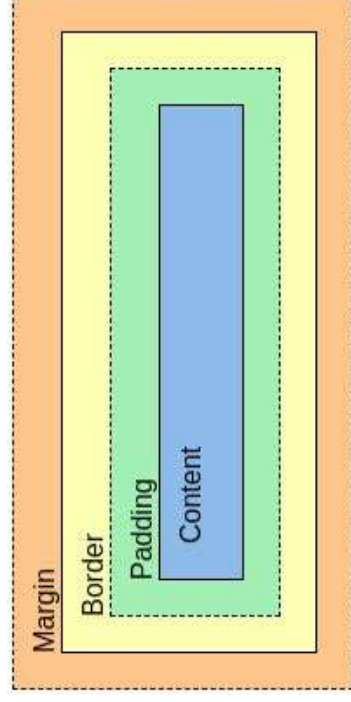


BOX MODEL

Qué es el Box Model

CSS trata a cada elemento HTML como una **caja**, con varias propiedades diferentes que determinan dónde aparece en la página.

Es, en esencia, una caja que envuelve al elemento, y consiste en margen, padding, border y el contenido.



Content, padding, border & margin

El "modelo de caja" es un conjunto de reglas que determinan las dimensiones de cada elemento en una página web. Le da a cada caja cuatro propiedades:

- **Contenido:** textos, imágenes u otros contenidos multimedia del elemento.
- **Padding** (relleno): el espacio entre el contenido de la caja y su borde.
- **Border** (borde): la línea entre el relleno y el margen de la caja.
- **Margin** (margen): el espacio entre la caja y las cajas a su alrededor.

El contenido es lo que creamos en un documento HTML y es el único que tiene algún valor semántico. El resto de ellos son puramente de presentación, por lo que están definidos por reglas CSS.

Padding

La propiedad **padding** agrega un relleno al elemento.

El *padding: 50px*; agrega 50 píxeles de espacio a cada lado (2do ejemplo). Y el espacio gris se expande para cubrir el espacio.

```
h1 {  
  font-family: Impact, sans-serif;  
  color: #b71b59;  
  background-color: lightgrey;  
}
```

```
h1 {  
  font-family: Impact, sans-serif;  
  color: #b71b59;  
  background-color: lightgrey;  
  padding: 50px;  
}
```

ADA LOVELACE

ADA LOVELACE

Padding

La sintaxis, en caso que necesitemos modificar el espacio de solo un lado, CSS nos provee las siguientes propiedades:

```
p { padding-top: 20px;  
padding-bottom: 20px;  
padding-left: 10px;  
padding-right: 10px;  
}
```

Y podemos utilizar solo 1, 2, 3 o las 4.

Padding

CSS también nos provee de un shorthand (atajo) para definir el padding:

- Definiendo 2 valores, el primero para el padding top y bottom, y el segundo para el padding left y right:

```
p {  
  padding: 20px 10px; /* vertical horizontal */  
}
```

- Definiendo 4 valores: uno por cada lado, empezando por el top en el orden de un reloj

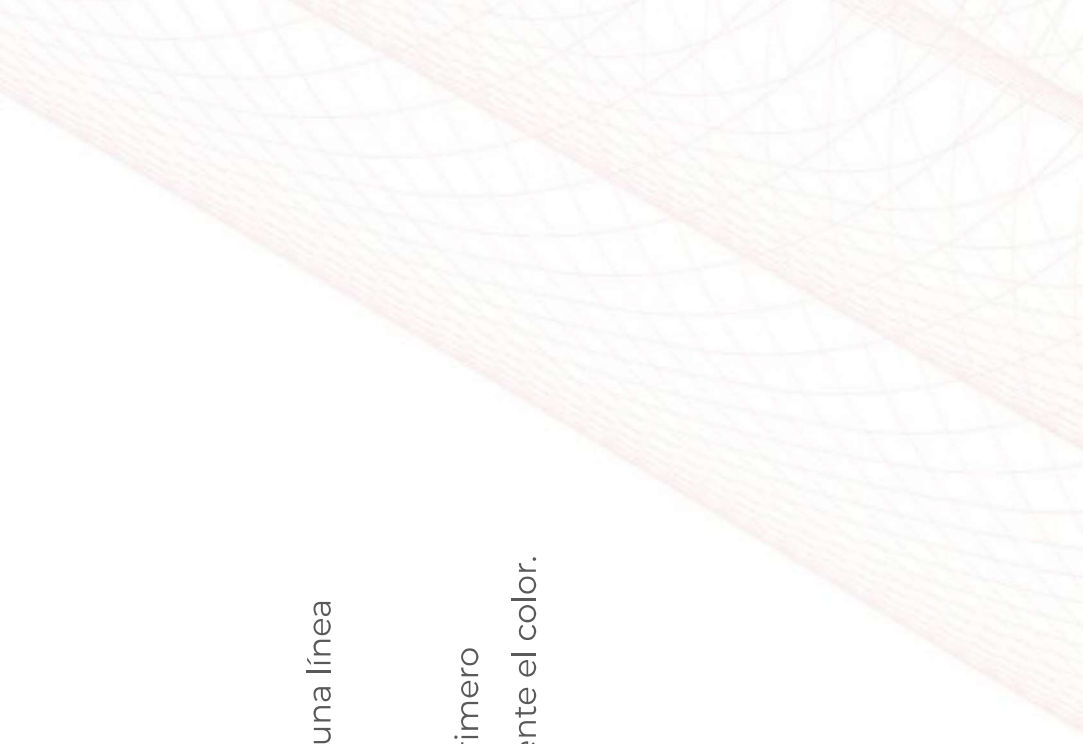
```
p {  
  padding: 20px 0 20px 10px; /* Top Right Bottom Left */  
}
```

Border

Continuando de adentro hacia afuera, tenemos el borde: una línea dibujada alrededor del contenido y el padding.

Para la propiedad *border* se agrega una sintaxis nueva: primero definimos el grosor del borde, después su estilo y finalmente el color.

border: 1px solid #f5f5f5;



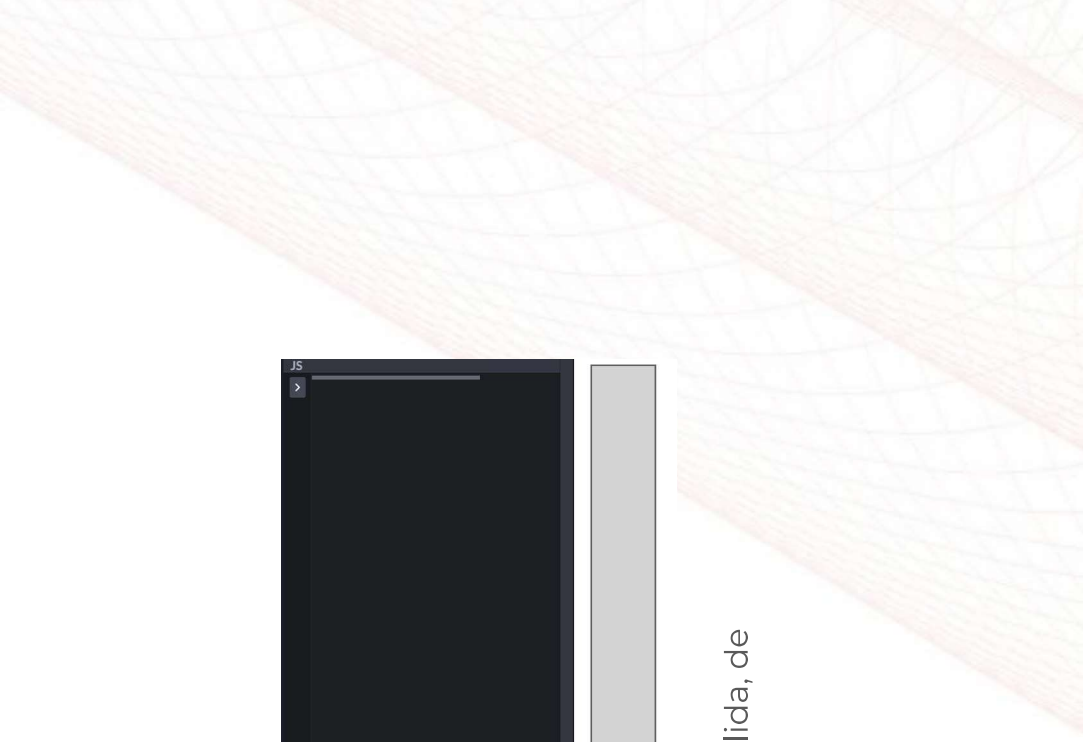
Border

```
HTML
1 <html>
2 <head>
3 <title>HTML CSS JS</title>
4 </head>
5 <body>
6 <h1>ADA LOVELACE</h1>
7 </body>
8 </html>

CSS
1 h1 {
2   font-family: Impact, sans-serif;
3   color: #071B59;
4   background-color: lightgrey;
5   padding: 20px;
6   border: 2px solid #506063;
7 }
8
```



Este ejemplo le dice al navegador que dibuje una línea sólida, de color gris, que mide 2px de grosor.



Border

Al igual que padding, también tenemos las propiedades para definir el borde de cada lado por separado.

```
HTML
1 <html>
2 <head>
3   <title>HTML CSS JS</title>
4 </head>
5 <body>
6   <h1>ADA LOVELACE</h1>
7 </body>
8 </html>

CSS
1 h1 {
2   font-family: Impact, sans-serif;
3   color: #b71b59;
4   background-color: lightgrey;
5   padding: 20px;
6   border-top: 2px solid blue;
7   border-right: 2px solid red;
8   border-bottom: 2px solid black;
9   border-left: 2px solid orange;
10 }

JS
```

ADA LOVELACE

Border

Hay varios estilos que se le pueden dar a los bordes, y se pueden combinar

.....
A dotted border.

A dashed border.

A solid border.

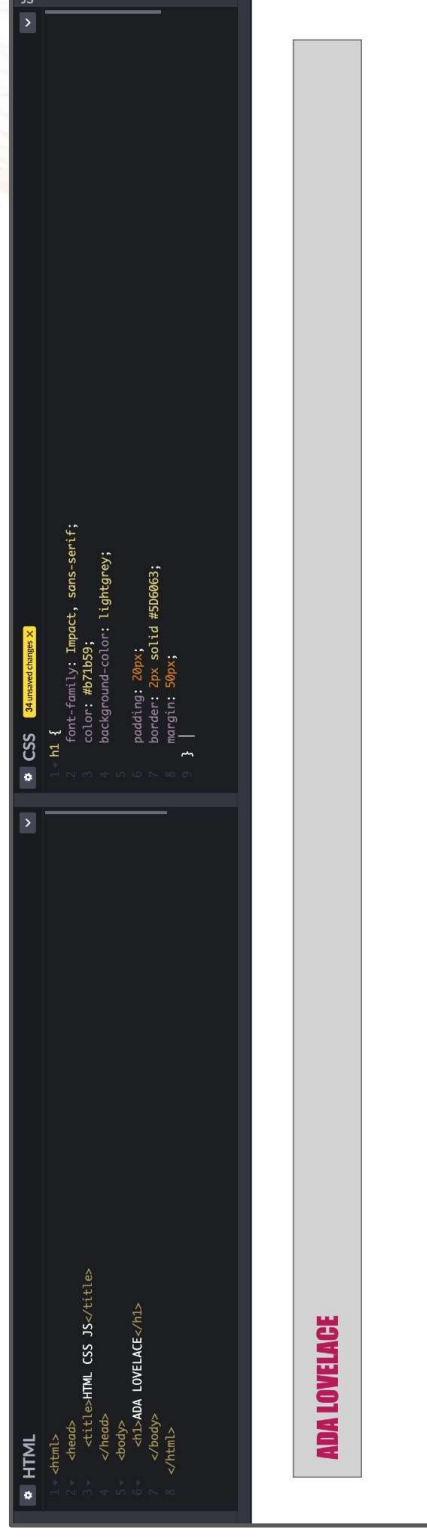
=====

No border.

Margin

Los márgenes definen el espacio fuera del borde de un elemento. Es el espacio entre una caja y las cajas que la rodean.

Si comparamos con los ejemplos de padding/border, vemos que se separó del límite de la pantalla (pero el fondo gris no se extendió).



Margin

En el ejemplo se ve una variante específica de la propiedad *margin*, pero también acepta los mismos formatos abreviados que *padding*.

```
p {  
  margin-top: 20px;  
  margin-bottom: 20px;  
  margin-left: 10px;  
  margin-right: 10px;  
}  
  
p {  
  margin: 20px 10px; /* vertical horizontal */  
}  
  
p {  
  margin: 20px; /* top bottom right left */  
}
```

Margin

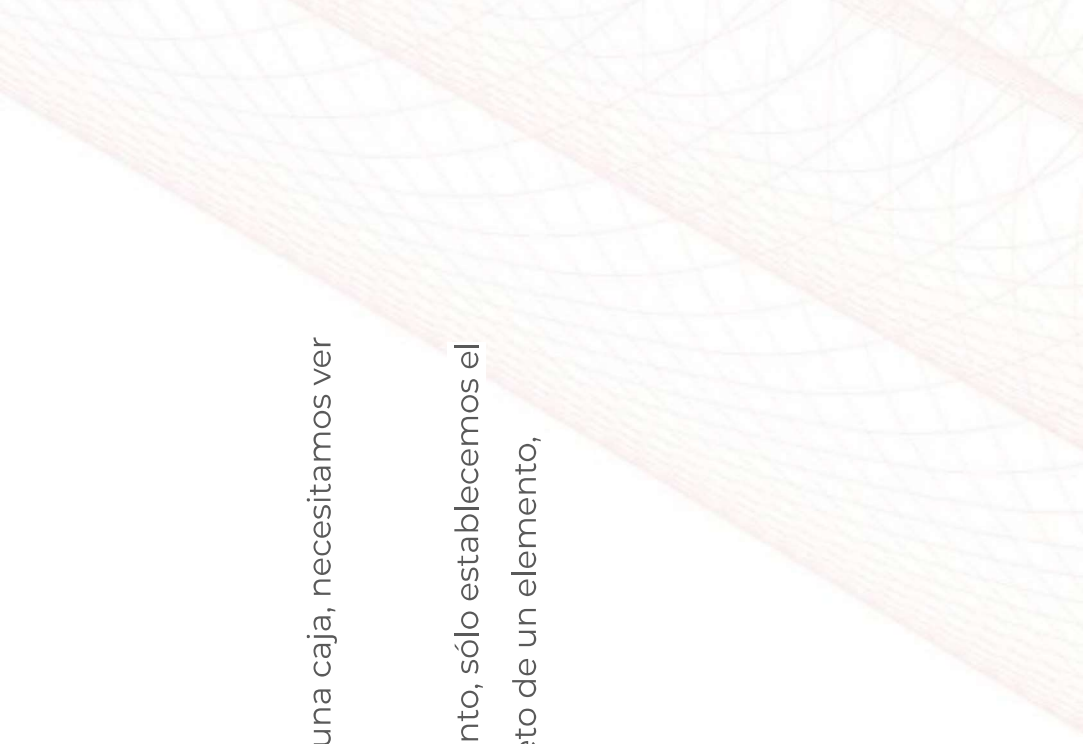
Las propiedades *margin* y *padding* pueden lograr, en muchas situaciones, lo mismo. Esto dificulta determinar cuál es la opción correcta al momento de usarlos. Algunas razones por las que se elegiría una opción sobre la otra:

- El relleno de un cuadro tiene un fondo, mientras que los márgenes son siempre transparentes.
- El relleno se incluye en el área de click de un elemento, mientras que los márgenes no.
- Los márgenes se contraen verticalmente, mientras que el relleno no.

Ancho/Alto de una caja

Para entender bien cómo configurar el alto y/o ancho de una caja, necesitamos ver cómo trabaja **box model**.

Importante: cuando usamos *width* o *height* de un elemento, sólo establecemos el ancho/alto del contenido. Para calcular el tamaño completo de un elemento, tenemos que sumar `padding`, `borders` y `margins`.



Ancho/Alto de una caja

```
div {  
  width: 320px;  
  padding: 10px;  
  border: 5px solid gray;  
  margin: 0;  
}
```

El ancho total de la caja es 350px, por que?

$320\text{px} (\text{width}) + 20\text{px} (\text{left+right}) + 10\text{px} (\text{left+right}) + 0\text{px} \text{ margin}$



Ancho/Alto de una caja

El ancho total de un elemento debe calcularse así:

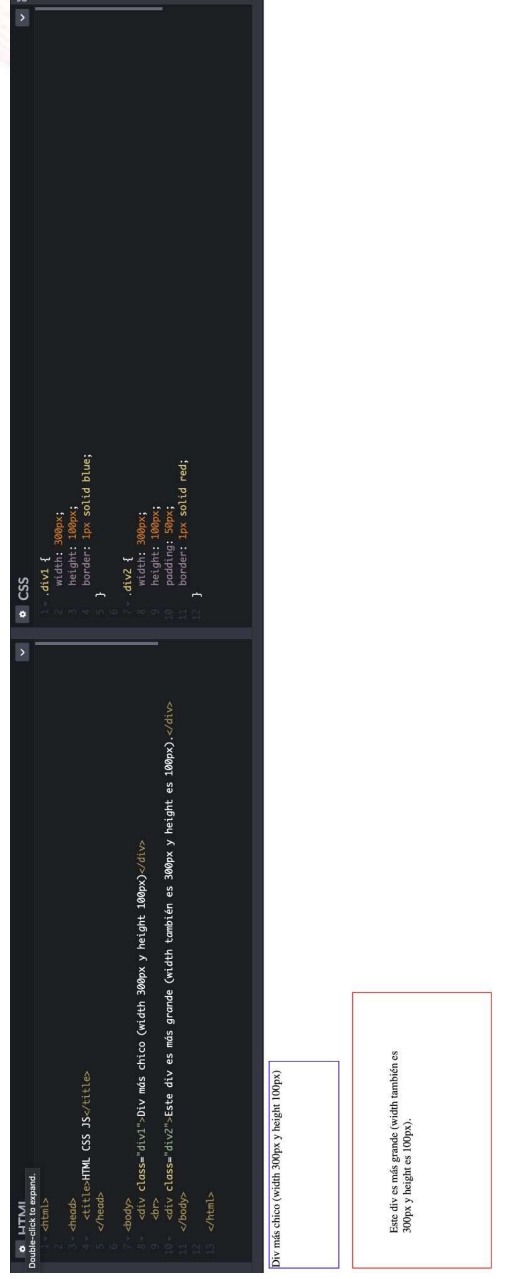
Ancho total = ancho + padding izquierdo + padding derecho + border izquierdo + border derecho + margin izquierdo + margin derecho

La altura total de un elemento debe calcularse así:

Altura total = altura + padding superior + padding inferior + border superior + border inferior + margin superior + margin inferior

Box Sizing

La propiedad de CSS **box-sizing** nos permite incluir el padding y border en el ancho y alto total de un elemento. Como vimos antes, por defecto la altura/ancho real de un elemento es $\text{ancho/alto} + \text{padding} + \text{border} + \text{margin}$. Entonces, cuando configuramos un ancho/alto para un elemento, si tiene padding o border, va a ser más grande del tamaño que seteamos.



The image shows a code editor with two panels. The left panel displays HTML code, and the right panel displays CSS code.

HTML Code:

```
1 <html>
2
3 <head>
4   <title>HTML CSS JS</title>
5 </head>
6
7 <body>
8   <div class="div1">Div más chico (width 300px y height 100px)</div>
9 </div>
10  <div class="div2">Este div es más grande (width también es 300px y height es 100px)</div>
11 </body>
12 </html>
```

CSS Code:

```
1 .div1 {
2   width: 300px;
3   height: 100px;
4   border: 1px solid blue;
5 }
6
7 .div2 {
8   width: 300px;
9   height: 100px;
10  padding: 50px;
11  border: 1px solid red;
12 }
```

Below the code editor, two visual representations of the box models are shown:

- Div más chico (width 300px y height 100px):** A blue-bordered rectangle representing the first div.
- Este div es más grande (width también es 300px y height es 100px):** A red-bordered rectangle representing the second div, which includes padding.

Box Sizing

Esto se puede solucionar con la propiedad `box-sizing`. Esta propiedad nos permite incluir `padding` y `border` en la suma de la anchura/altura total de un elemento. Si en el ejemplo anterior ponemos **`box-sizing: border-box;`** en el segundo `div`, el relleno y el borde se incluyen en el ancho y el alto:

```
1 <html>
2 <head>
3   <title>HTML CSS JS</title>
4 </head>
5
6 <body>
7   <div class="div1">Div más chico (width 300px y height 100px)</div>
8   <div class="div2">Este div es más grande (width también es 300px y height es 100px)</div>
9 </body>
10 </html>
```

```
1 -div1 {
2   width: 300px;
3   height: 100px;
4   border: 1px solid blue;
5 }
6
7 -div2 {
8   width: 300px;
9   height: 100px;
10  padding: 50px;
11  border: 1px solid red;
12  box-sizing: border-box;
13 }
```

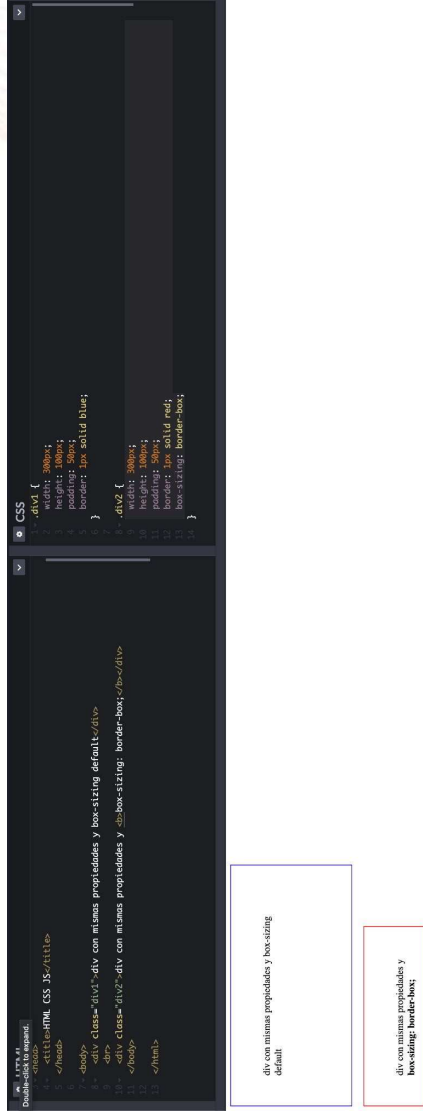
Div más chico (width 300px y height 100px)

Este div es más grande (width también es 300px y height es 100px)

Box Sizing

En resumen, la propiedad **box-sizing** define cómo se calcula el ancho y el alto de un elemento: si deben incluir rellenos y bordes, o no. Las dos opciones disponibles son:

- Content-box (default): Las propiedades width y height incluyen solo el contenido. **No** incluye padding y border.
- Border-box: Las propiedades width y height incluyen el contenido, padding y borde del elemento.



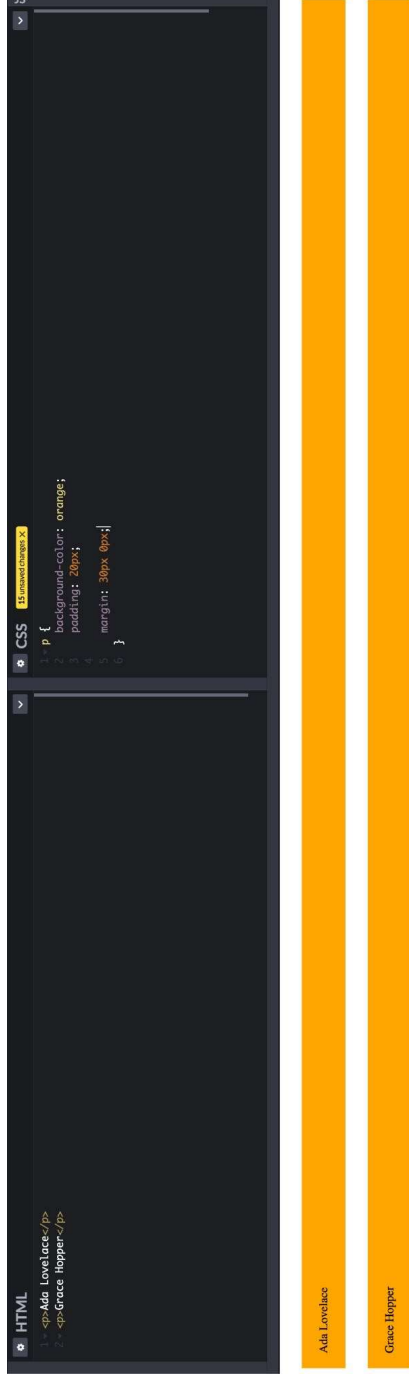
Colapsamiento de margen

"En CSS, los márgenes adyacentes de dos o más bloques (que podrían o no ser hermanos) pueden combinarse para formar un único margen. Cuando los márgenes se combinan de esta manera decimos que colapsan, y el margen combinado resultante se denomina margen colapsado."

W3C

Los márgenes adyacentes son márgenes verticales que cumplen algunas reglas. Esto significa que los márgenes horizontales (margin-left y margin-right) no se consideran márgenes adyacentes.

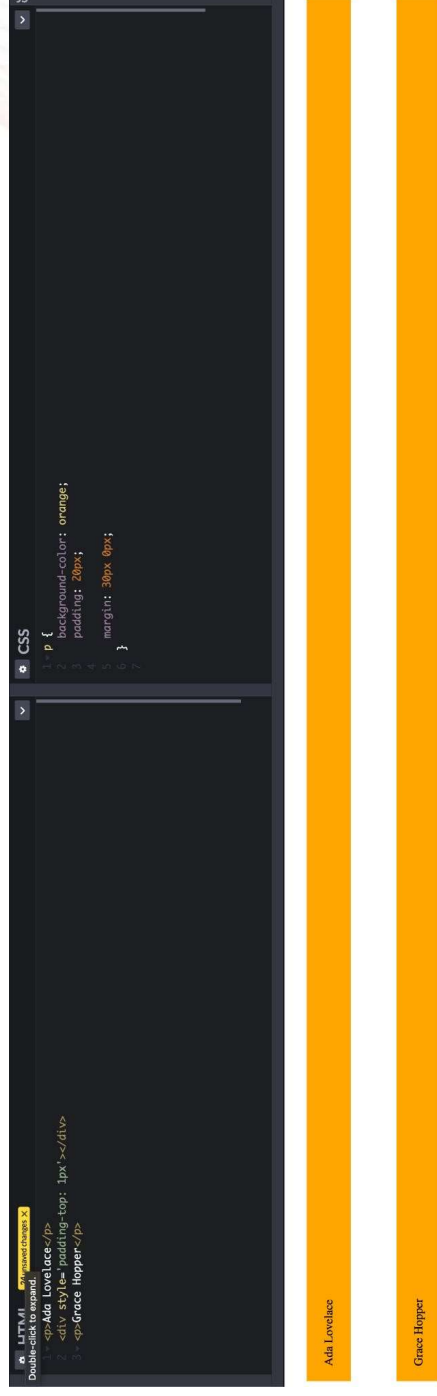
Colapsamiento de margen



En este ejemplo, cada `<p>` tiene un margen de 30px. De esta forma, esperaríamos que el margen total entre ambos `<p>` sea de 60px. Pero no es así. Solo van a tener 30px, ya que el margen más pequeño (en este caso son iguales) se colapsa con el más grande.

Colapsamiento de margen

Una posible solución si quisiéramos evitar ese colapso, podría ser agregar un elemento vacío en el medio. La parte importante acá es que solo los elementos consecutivos pueden colapsar. Poner un elemento con una altura distinta de cero (de ahí el padding-top) entre nuestros párrafos los obliga a mostrar tanto el margen superior de 30px como el margen inferior de 30px.



Display

La propiedad display especifica el comportamiento de visualización (el tipo de cuadro de representación) de un elemento.

Posibles valores de la propiedad: <https://developer.mozilla.org/en-US/docs/Web/CSS/display>