

Display - CSS

Inline - Block

Propiedad display

La propiedad `display` especifica el comportamiento de visualización de un elemento.

Cada elemento tiene un valor de `display` por defecto dependiendo de la utilidad en la que fue diseñado. El valor por defecto para la mayoría de los elementos es usualmente `block` (de bloque) o `inline` (en línea).

Algunos elementos de tipo Bloque/Block

<code><article></code>	<code><aside></code>	<code><div></code>	<code><section></code>
<code><tr></code>	<code><td></code>	<code><table></code>	<code><footer></code>
<code></code>	<code><p></code>	<code><h1>/<h6></code>	<code><video></code>
<code></code>	<code><header></code>	<code><nav></code>	<code><hr></code>

display:block;

Un elemento de nivel de bloque siempre comienza en una **nueva línea** y ocupa todo el ancho disponible (se extiende hacia la izquierda y hacia la derecha tanto como sea posible).

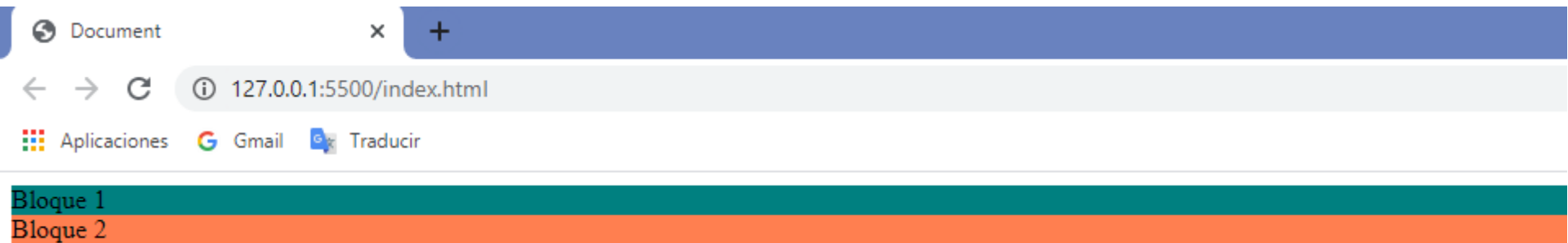
Por lo tanto, 2 elementos de tipo bloque no pueden estar uno al lado del otro.

¿Cuál sería el resultado esperado?

```
.teal {  
  background-color: teal;  
}  
  
.coral {  
  background-color: coral;  
}
```

```
<div class="teal">Bloque 1</div>  
<div class="coral">Bloque 2</div>
```

Resultado:



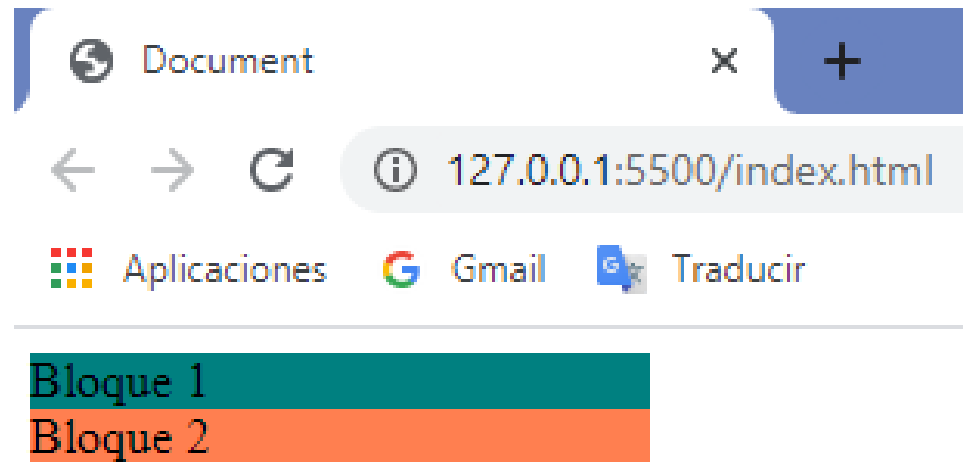
Ahora cada bloque tiene un ancho de 200px.

¿Cómo se verá?

¿Los div se pondrán uno al lado de otro?

```
.teal {  
  background-color: teal;  
  width: 200px;  
}  
  
.coral{  
  background-color: coral;  
  width: 200px;  
}
```

Resultado:





Recordar: Un elemento de tipo bloque siempre inicia en una nueva línea, sin importar si cuenta con espacio suficiente.

Importante:

Un elemento Bloque:

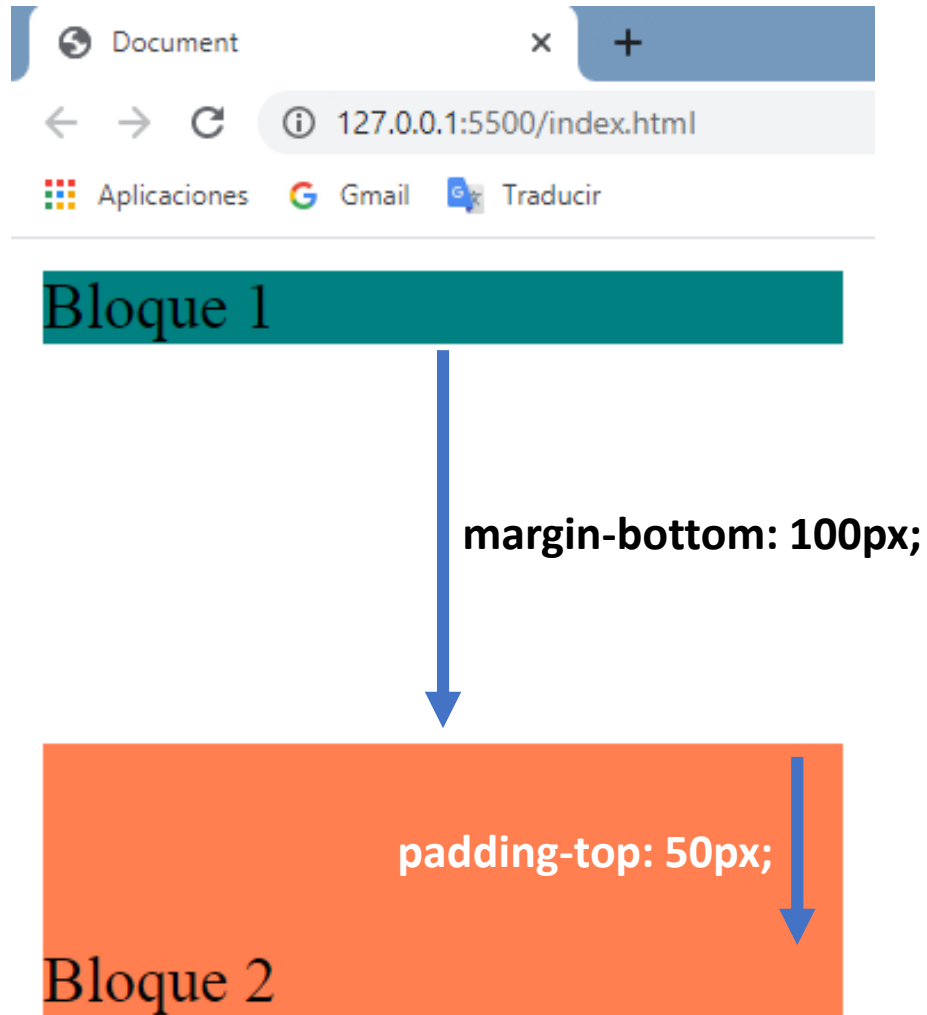
1. Siempre comienza en una nueva línea.
2. Si no se le indica un ancho, llena todo el espacio horizontal.
3. Márgenes Externos/Margin: Se puede aplicar hacia cualquier dirección (Top/Bottom/Right/Left).
4. Márgenes Internos/Padding: Se puede aplicar hacia cualquier dirección (Top/Bottom/Right/Left).

¿Cuál será el resultado?

```
.teal {  
  background-color:  teal;  
  width: 200px;  
  margin-bottom: 100px;  
}  
  
.coral{  
  background-color:  coral;  
  width: 200px;  
  padding-top: 50px;  
}
```

```
<div class="teal">Bloque 1</div>  
<div class="coral">Bloque 2</div>
```


Resultado:



Algunos elementos de tipo en Línea/Inline

<a>		
	<i>	

: Aunque br realice un salto de línea, también tiene un display inline.

display:inline;

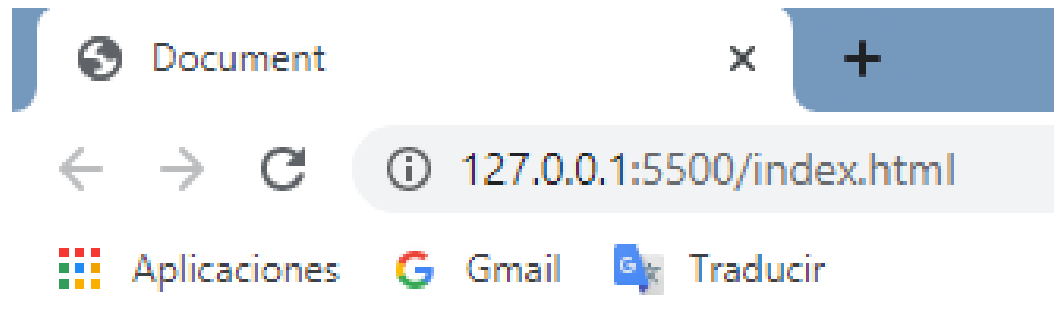
Un elemento en línea no comienza en una nueva línea y solo ocupa el ancho que sea necesario.

Por lo tanto, 2 elementos de tipo en línea estarán uno al lado del otro siempre que haya espacio suficiente.

¿Cuál sería el resultado esperado?

```
<span>En línea 1</span>  
<i>En línea 2</i>
```

Resultado



En línea 1 En línea 2

Recordar: Un elemento en línea no comienza en una nueva línea y solo ocupa el ancho que sea necesario.

Importante:

Un elemento en línea:

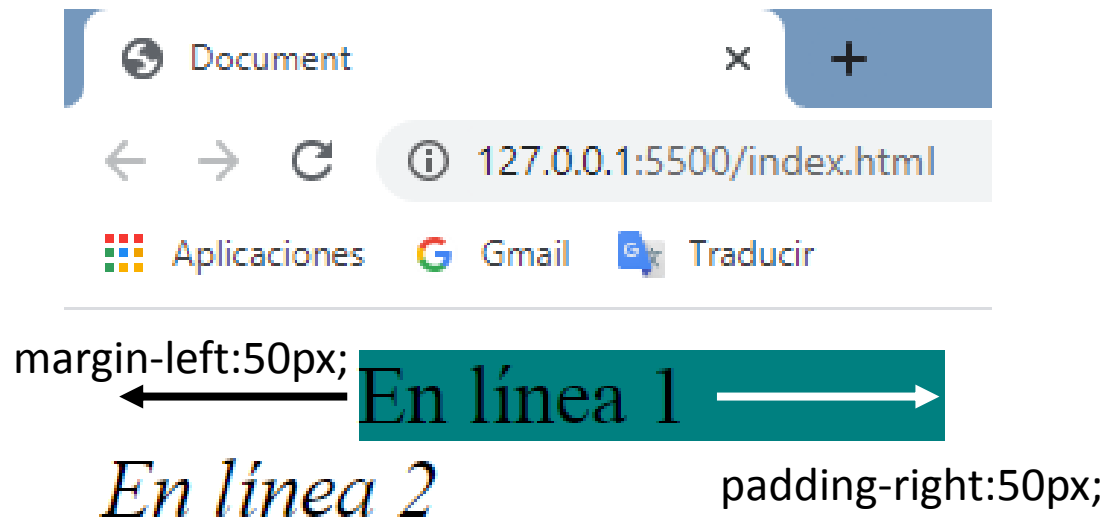
1. No comienza en una nueva línea.
2. Ocupa el espacio necesario.
3. Márgenes Externos/Margin: Se puede aplicar **solamente hacia la izquierda(Left) y derecha (Right)**.
4. Márgenes Internos/Padding: Se puede aplicar **solamente hacia la izquierda(Left) y derecha (Right)**.
5. Las propiedades de Alto (Height) y Ancho (Width) **no funcionan**.

¿Cuál será el resultado?

```
span {  
  margin-left: 50px;  
  margin-bottom: 100px;  
  padding-right: 50px;  
  
  background-color:  teal;  
}  
  
i {  
  margin-top: 100px;  
  padding-top: 100px;  
}
```

```
<span>En línea 1</span>  
<br />  
<i>En línea 2</i>
```

¿Cuál será el resultado?



1. `margin-bottom: 100px` – No funcionó en etiqueta ``.
2. `margin-top:100px` – No funcionó en etiqueta `<i>`.
3. `padding-top:100px` – No funcionó en etiqueta `<i>`.

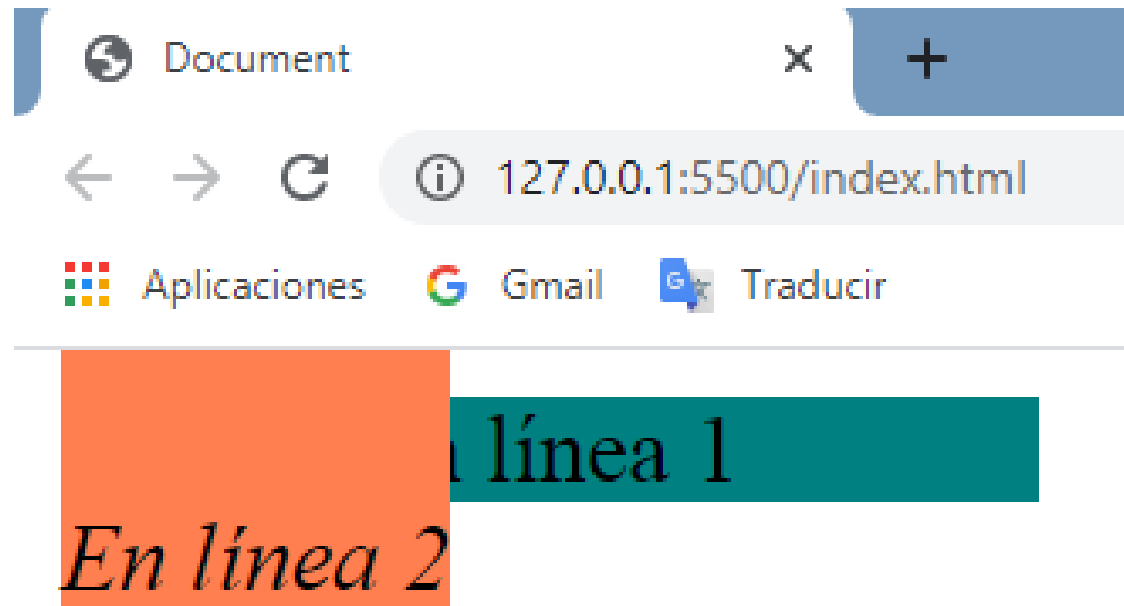
Utilizando el mismo ejemplo anterior.

Agregue un **background-color: coral;** al bloque CSS del selector `<i>`.

Observe el comportamiento indeseado por agregar una propiedad (`padding-top`) la cual ignora.

```
span {  
  margin-left: 50px;  
  margin-bottom: 100px;  
  padding-right: 50px;  
  
  background-color: teal;  
}  
  
i {  
  margin-top: 100px;  
  padding-top: 100px;  
  background-color: coral;  
}
```


Resultado:



display:inline-block;

Fusiona lo mejor de cada tipo (inline + block).

1. No comienza en una nueva línea. **(inline)**
2. Ocupa el espacio necesario. **(inline)**
3. Márgenes Externos/Margin: Se puede aplicar hacia cualquier dirección (Top/Bottom/Right/Left). **(block)**
4. Márgenes Internos/Padding: Se puede aplicar hacia cualquier dirección (Top/Bottom/Right/Left). **(block)**
5. Las propiedades de Alto (Height) y Ancho (Width) funcionan. **(block)**

¿Qué resultado se obtendrá?

```
div {  
  display: inline-block;  
  height: 100px;  
  width: 100px;  
}  
  
.div-1 {  
  background-color:  coral;  
  margin-bottom: 100px;  
  margin-left: 100px;  
}  
  
.div-2 {  
  padding-bottom: 100px;  
  background-color:  teal;  
}
```

```
<div class="div-1">Texto 1</div>  
<div class="div-2">Texto 2</div>
```

Resultado:

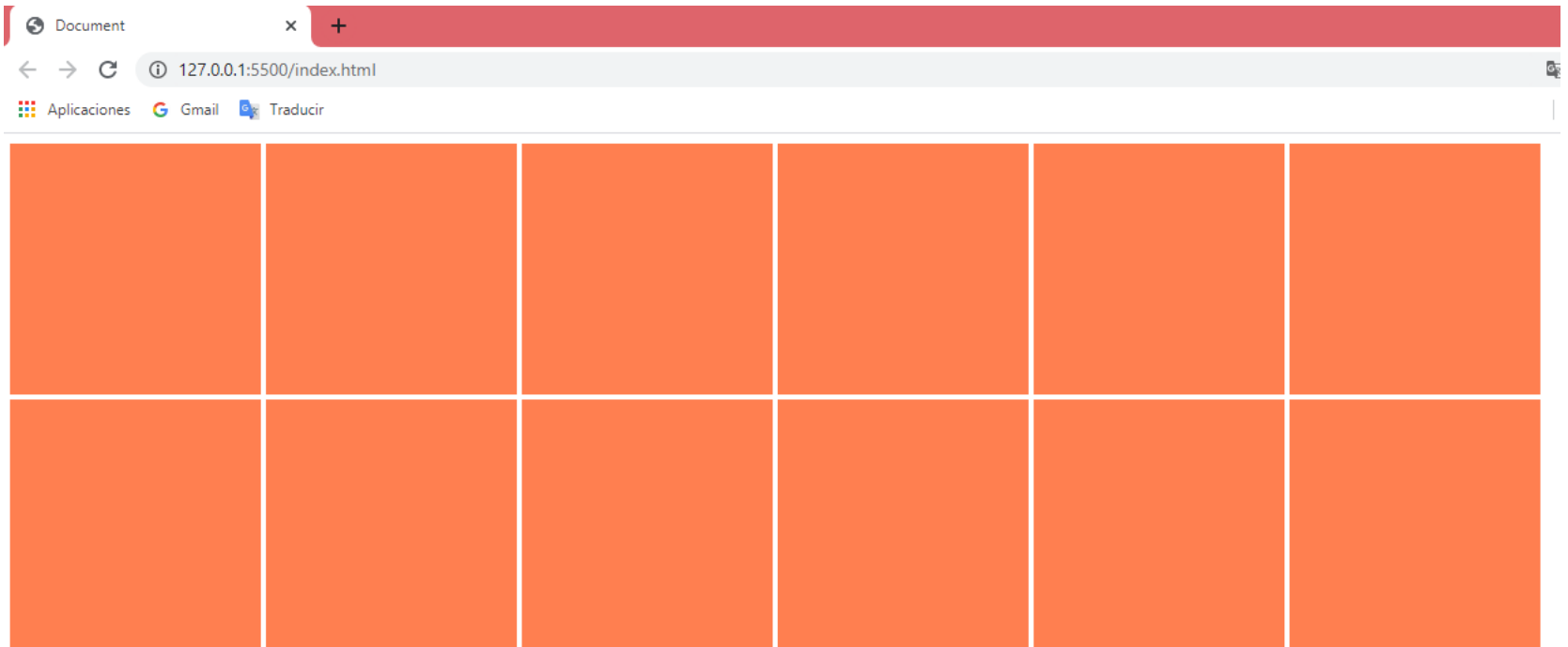


Ejercicio

Implemente el código necesario para obtener este resultado.

Utilice `display: inline-block`;

Observación: Realice zoom a la página 25% a 200% y observe el comportamiento.



Solución:

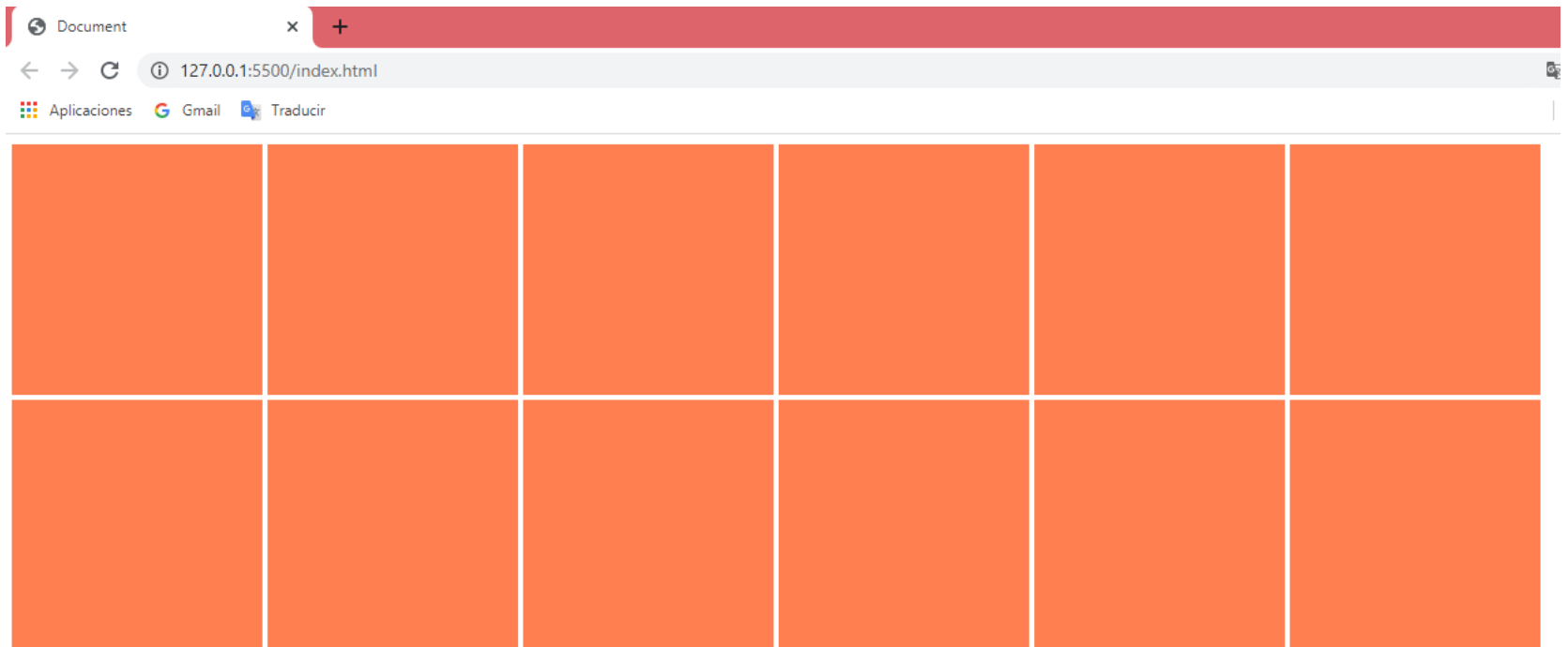
```
div {  
  display: inline-block;  
  background-color: coral;  
  height: 200px;  
  width: 200px;  
}
```

```
<body>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
  <div></div>  
</body>
```

Observación: El resultado puede variar dependiendo del tamaño del monitor.

Utilizando el ejemplo anterior

¿Por qué cada div tiene un margen de separación, si margin y padding son 0px?



Respuesta:

Porque los divs obtuvieron una de las propiedades de inline.

Para cada etiqueta inline se le aplica un “espacio” para separar las palabras por defecto.

En línea 1 *En línea 2*

```
<span >En línea 1</span>  
<i >En línea 2</i>
```

Soluciones posibles:

1. Indicar font-size:0px; en la etiqueta padre.
2. Utilizar la propiedad float:left; **//Muy pronto sabremos más de ella.**

Importante!

Si una de las soluciones era poner el font-size en 0px...

¿Qué sucedería si el font-size fuera de 100px?

Envuelva al grupo de divs en una etiqueta padre con la propiedad de font-size: 100px.

Consejo: Si manipulas un elemento inline-block junto con textos habrá que tener mucho cuidado con la interacción de propiedades **heredables** como font-size.

Interacción de propiedades

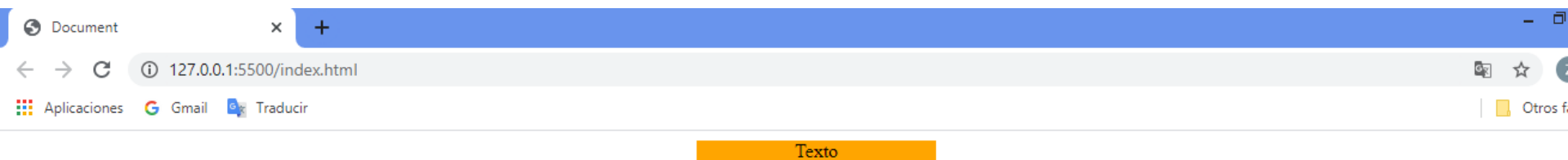
Si font-size afectaba un inline-block, ¿Qué sucedería con la propiedad text-align?

¿Tendrá algún efecto?

text-align ahora interacciona con un div

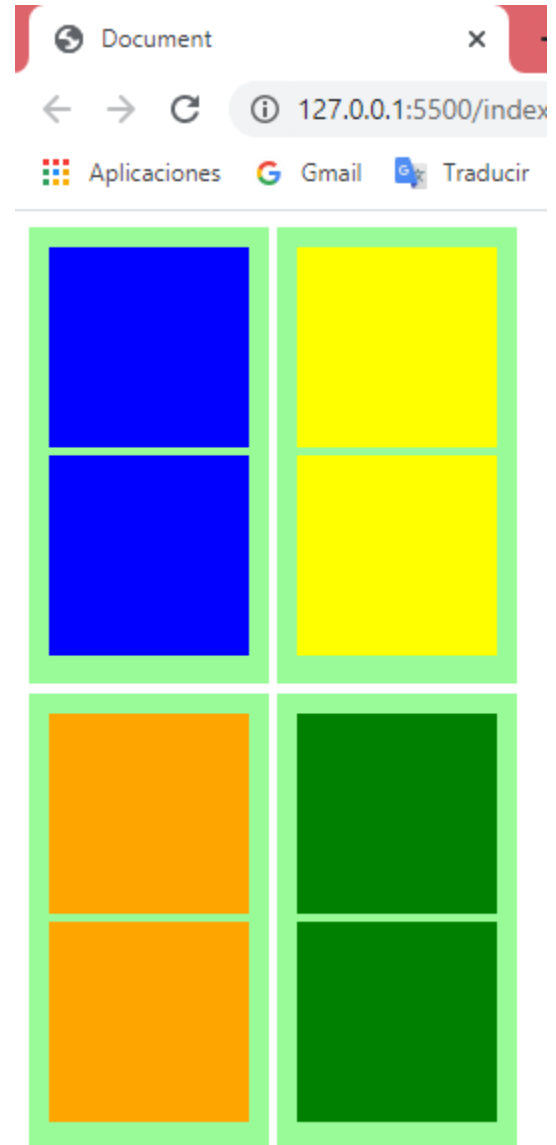
```
section {  
  text-align: center;  
}  
  
div {  
  display: inline-block;  
  width: 200px;  
  background-color: orange;  
}
```

```
<section>  
  <div>Texto</div>  
</section>
```



Ejercicio

Implemente el código necesario para obtener el siguiente resultado (Sin usar `<table>`).
Utilice `display: inline-block;`



Solución

```
<section>
  <article>
    <div class="blue"></div>
    <div class="blue"></div>
  </article>
  <article>
    <div class="yellow"></div>
    <div class="yellow"></div>
  </article>
  <article>
    <div class="orange"></div>
    <div class="orange"></div>
  </article>
  <article>
    <div class="green"></div>
    <div class="green"></div>
  </article>
</section>
```

```
section { width: 250px; }

article {
  width: 100px;
  display: inline-block;
  background-color:  palegreen;
  padding: 10px;
  margin-bottom: 5px;
}

div {
  display: inline-block;
  background-color:  coral;
  height: 100px;
  width: 100px;
}

.blue { background-color:  blue; }
.yellow { background-color:  yellow; }
.green { background-color:  green; }
.orange { background-color:  orange; }
```

Problemas de alturas

Implemente el siguiente código e indique qué sucede.

```
.div-1 {  
  display: inline-block;  
  background-color: teal;  
}  
  
.div-2 {  
  display: inline-block;  
  background-color: coral;  
}  
  
.div-1 > p {  
  font-size: 50px;  
}  
  
.div-2 > p {  
  font-size: 20px;  
}
```

```
<div class="div-1">  
  <p>Esto es un texto alíneado</p>  
  <p>Esto es un texto alíneado</p>  
</div>  
<div class="div-2">  
  <p>Esto también es un texto alineado... Espera..</p>  
</div>
```

Respuesta

El segundo Div, se alinea con la línea del último texto a la izquierda.

Para eliminar este problema **agregar** al interior de cada div la propiedad **vertical-align:top;**



Ejercicio

1. En la primera fila hay 3 contenedores de color verde, azul y naranja.
 1. En ancho de cada uno es de 33.33% respecto a su padre.
2. En la segunda fila hay un contenedor violeta de ancho 100% respecto a su padre.
3. En la tercera fila hay 2 contenedores de colore verde y naranja.
 1. El ancho de cada uno es de 50% respecto a su padre.
4. Todos los contenedores tiene un alto de 100px.







Solución

```
section { font-size: 0px; }

.w-33 { width: 33.33%; }
.w-50 { width: 50%; }
.w-100 { width: 100%; }

div {
  display: inline-block;
  height: 100px;
}

.blue { background-color:  blue; }
.green { background-color:  yellowgreen; }
.violet { background-color:  violet; }
.orange { background-color:  orange; }
```

```
<section>
  <div class="w-33 green"></div>
  <div class="w-33 blue"></div>
  <div class="w-33 orange"></div>
  <div class="w-100 violet"></div>
  <div class="w-50 green"></div>
  <div class="w-50 orange"></div>
</section>
```

En conclusión

Ahora existe una nueva forma para agrupar elementos entregando muchas ventajas, pero a la vez se debe tener cuidado como se utilizará los elementos inline-block, ya que su comportamiento está basado en un bloque y texto a la vez , eso podría llegar a producir en algunas ocasiones comportamientos difíciles de entender, sobre todo con propiedades heredables.