ThePower business school

Display CSS ↕🎨

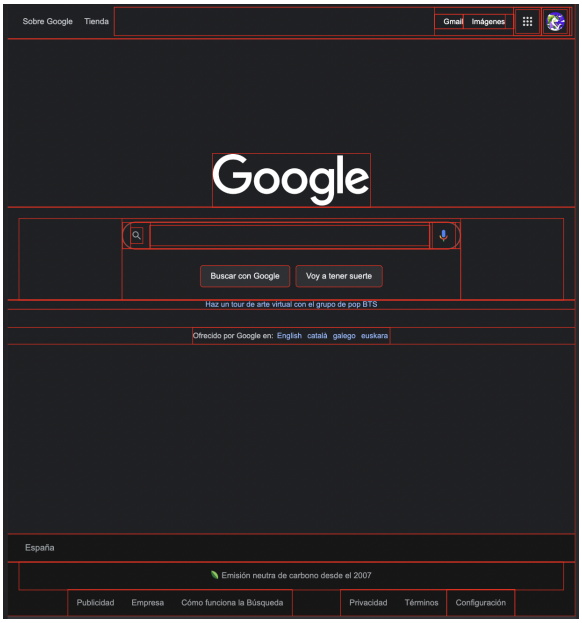
Es importante que entiendas que cada elemento HTML en una pagina web es una caja rectangular, esta es la forma en que se representan todos los elementos, no existen elementos triangulares, redondos, poligonales etc. Todos los elementos en HTML por defecto son rectangulares ya que internamente el navegador dibuja un rectángulo.

Inline serán los elementos en línea, que se colocan uno a continuación de otro de forma horizontal. Ocuparán el ancho necesario y no todo el ancho disponible como lo harán los elementos block o de bloque. Estos elementos ocuparán la totalidad del ancho y se llaman así porque se entienden como un bloque de contenido.

En el siguiente ejemplo, con ayuda de la propiedad **outline**, podemos ver todos los elementos de la pagina de inicio de Google de forma rectangular y esto lo puedes hacer con cualquier otro sitio web.

```
* {
    outline: 1px solid red;
}
```

Para ello en el navegador - click izquierdo inspeccionar y meter el estilo por la consola. En nuestro caso sería atacar a todas las etiquetas **div** y meter el **outline**.



Os dejamos un ejemplo de los comportamientos principales de la mayoría de elementos HTML, que serán inline y block:

```
div {
    background-color: blueviolet;
}
span {
    background-color: steelblue;
}
```

Como vemos en el ejemplo, un div será un ejemplo de elemento que se comporta como un bloque y un span será un ejemplo de comportamiento en línea.

```
<div>Este es un ejemplo de comportamiento de bloque</div>
<span>Este es un ejemplo de comportamiento en línea</span>
```

Este es un ejemplo de comportamiento de bloque
Este es un ejemplo de comportamiento en línea

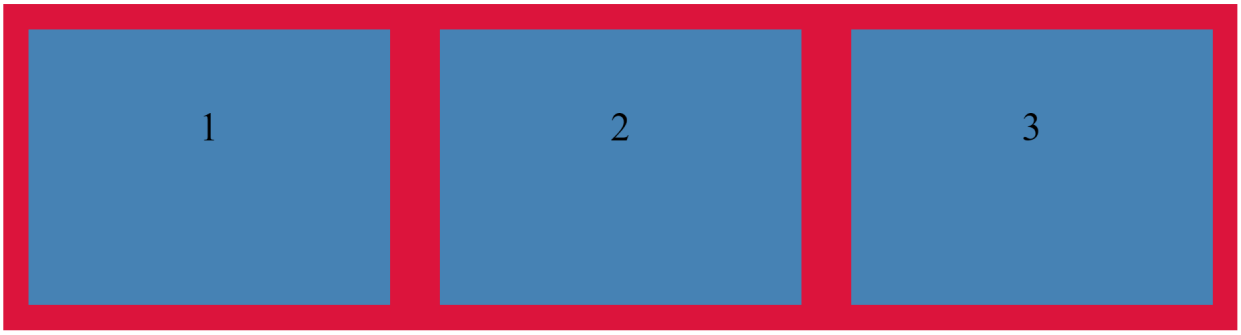
Ahora bien, el comportamiento por defecto de los elementos se puede modificar con la propiedad display, que dependiendo del valor que debemos modificará de una forma u otra el comportamiento del elemento. Vamos a ver los tipos:

- block: hace que el elemento ocupe la totalidad del ancho, por lo que los elementos se apilarán en vertical y en una línea solo podrá existir un elemento.
- inline: adapta el ancho del elemento al contenido del mismo. Como veíamos en el ejemplo anterior de código, el ocupa únicamente lo que ocupa su contenido. Estos elementos no tienen en cuenta los valores de width o height que les añadamos.
- inline-block: Combina los dos anteriores, ya que permite que creemos bloques pero del width y height que queramos.
- flex: utiliza un modelo de cajas flexibles que será una sección en sí misma debido a su riqueza de contenido.
- grid: al igual que flex, será una sección en sí misma, pero lo importante es que usa cuadrículas o “grillas” en su modelo para repartir el contenido.
- contents: necesitaremos nociones de flex y grid para usarlo corretamente. Será de ayuda a la hora de mantener la relación padre-hijo que es necesaria en los elementos con display flex/grid. Este valor se le suele dar a la propiedad display por necesidades de javascript.
- none: ocultará el elemento y nos será muy útil cuando lo mezclemos con javascript, para renderizados condicionales o en los que en algunos casos debamos mostrar unos elementos y en otros casos otros elementos distintos. Es un comportamiento similar a la propiedad visibility con el valor de hidden, solo que visibility además hace “desaparecer” el elemento, por lo que no se verá en nuestro árbol HTML o DOM.

Existen más opciones pero estas son las más importantes y las que más usaréis. Os dejamos un ejemplo de código para que practiquéis:

```
<div>
    <span>1</span>
    <span>2</span>
```

```
    <span>3</span>
</div>
```



Lo único que tendréis que hacer es cambiar el valor de la propiedad display de block a las que os hemos ido explicando para ver los resultados y que os hagáis una mejor idea!

```
div{
    margin: 0;
    padding: 0;
    display: block;
    background-color: crimson;
}
span {
    background-color: steelblue;
    width: 100px;
    height: 50px;
    padding: 30px;
    list-style-type: none;
    text-align: center;
    margin: 10px 10px;
}
```

¿Qué hemos aprendido?

1. Qué es la propiedad display y cuáles son sus valores más frecuentes.
2. Los distintos comportamientos que tendrán los elementos por defecto y cómo modificarlo.