

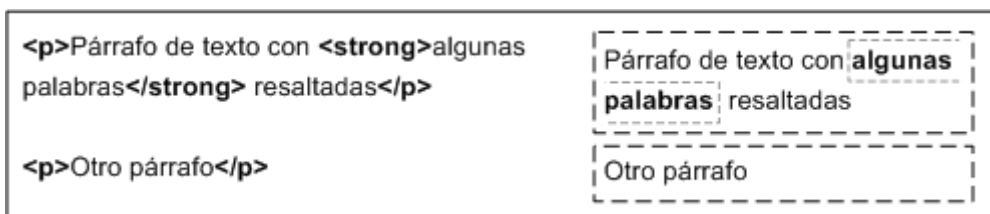
### Modelo de caja

El modelo de cajas o "**box model**" es uno de los conceptos más importante del lenguaje de hojas de estilos CSS, ya que **condiciona el diseño** de todas las páginas web.

Es decir cómo se renderiza (como se dibuja) la estructura que escribimos HTML en la página final (resultado final) que es visible para todos.

Las cajas se crean automáticamente al definir cada elemento HTML. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja (**box**) rectangular que encierra los contenidos de ese elemento.

La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:



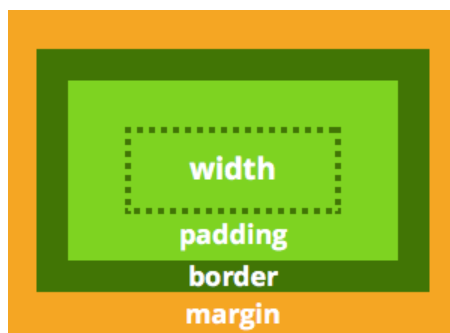
### Importante:

Es importante primero diferenciar los elementos inline y los elementos en bloques

- Los elementos inline:
  - No crean nuevas líneas para cada elemento (una línea es toda una división horizontal de toda la pantalla) esto quiere decir que aparece un elemento y el siguiente elemento se pone al lado y así sucesivamente.
  - Tiene ancho, pero no alto dado que se ajustan a su contenido
  - Por ejemplo, un elemento span, una imagen
- Los elementos de bloque:
  - Crean una nueva línea para cada elemento, por ejemplo, ponemos un elemento y éste ya crea una línea, por lo que el siguiente elemento se pondrá por debajo
  - Ocupan todo el espacio horizontal disponible
  - Tienen ancho y alto
  - Por ejemplo, los encabezados h1-h6 son elementos en bloque al igual que los párrafos

## Box model

El Box Model (Modelo de caja) está compuesto por la suma de unas variables:



## Todo elemento HTML tiene un WIDTH (contenido)

Como dijimos los elementos inline ocupan el tamaño de su contenido y los elementos en bloques ocupan toda la anchura del navegador.

Con CSS podremos modificar la anchura de los elementos de tipo bloque:

```
h1{  
    width:100px;  
}
```



## Propiedad PADDING

El padding nos permitirá definir un espacio vacío alrededor de un elemento. Retomando el ejemplo anterior, vamos a declarar un espacio de 15px al rededor del encabezado h1. Para ello haríamos:

```
h1{  
    width: 100px;  
    padding-top: 15px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
}
```

Es decir, para cada lado la caja le asignamos un espacio vacío de 15px.

Es importante aclarar que **podemos elegir el valor que quisiéramos para cada lado**. En caso de que todos los lados tengan la misma medida tan solo es necesario escribir:

```
h1{  
    width: 100px;  
    padding: 15px;  
}
```

Por ejemplo, si quisiéramos 10px arriba, 5px a la derecha, 15px abajo y 0px a la izquierda. En este caso podríamos hacer como en el ejemplo inicial o utilizar la sintaxis de una sola fila:

```
h1{  
    width: 100px;  
    padding: 10px 5px 15px 0;  
}
```

Su sintaxis funciona en sentido horario:  
padding: arriba derecha abajo izquierda  
padding: top right bottom left

De esta forma podemos ver en el esquema que la zona vacía que se corresponde al padding está pintado del mismo verde que el width. Esto es porque como vemos el padding comparte el mismo color de fondo que el elemento en sí.



## Propiedad BORDER

Si queremos asignar un borde a la “caja” tendremos que definir tres propiedades: la **anchura**, el **estilo** y el **color**:

```
h1{
  width: 100px;
  padding: 15px;
  border-style: solid;
  border-width: 4px;
  border-color: blue;
}
```



```
h1{
  width: 100px;
  padding: 15px;
  border: 4px solid blue;
}
```

El estilo puede tomar distintos valores, que podrás observar en la siguiente página

[https://www.w3schools.com/css/css\\_border.asp](https://www.w3schools.com/css/css_border.asp)



## Propiedad MARGIN

En CSS el margen se aplica fuera de la caja (por fuera de mi objeto), es decir el margen no mantiene el color del fondo del elemento, sino que siempre será transparente.

Su sintaxis es idéntica al padding.

Si a nuestro ejemplo le declaramos un **margen superior (top)** de 20px, escribiremos:

```
h1{
  width: 100px;
  padding: 15px;
  border: 4px solid green;
  margin-top: 20px;
}
```