

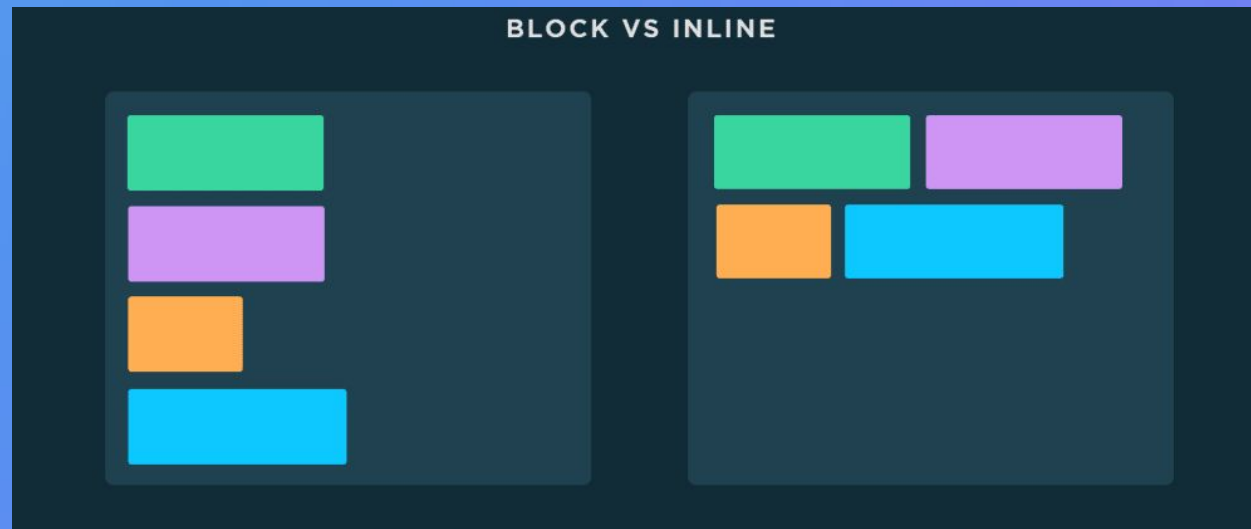


# Position

## Position - Problema a solucionar

Hasta ahora conocemos que los elementos HTML se dividen en dos grandes tipos:

- **block**, que tienden a ocupar el espacio disponible a todo lo ancho y en caso de existir varios se sitúan unos debajo de otros
- **inline**, que ocupan el espacio necesario dentro de una línea y en caso de existir varios se sitúan uno junto a otro en la misma línea (siempre que haya espacio).

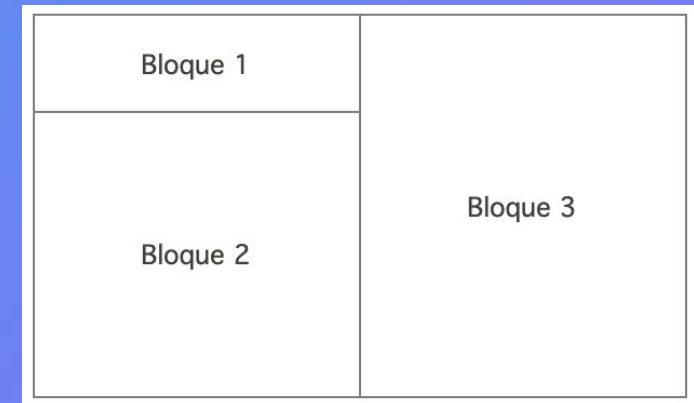


## Position - Problema a solucionar

Recordar que las propiedades width y height sólo son aplicables a elementos tipo **block o inline-block**.

Ahora bien, cuando creamos páginas web necesitaremos que elementos de dimensiones determinadas se sitúen en **posiciones determinadas**.

Por ejemplo podemos querer tener tres elementos con dimensiones width y height definidas (por tanto bloques) situados en determinadas posiciones como vemos en este esquema:



Con el código simple que hemos visto hasta ahora esto no sería posible, ya que unos elementos se situarán debajo de otros, sin cumplir nuestro objetivo de situarlos en posiciones específicas.

CSS tiene distintas propiedades y posibilidades que nos van a permitir **gestionar el posicionamiento de elementos** para crear diseños atractivos y conforme a nuestras necesidades.

Comenzaremos el estudio de estas posibilidades con la propiedad CSS **position**.

## Position - Intro

Esta propiedad nos permite establecer la posición de un elemento.

Ejemplo de uso:

```
.contenedor {  
  position: absolute;  
}
```

## Position - Valores disponibles

### **static**

Comportamiento normal o por defecto.



```
.contenedor {  
  position: static;  
}
```

### relative

Permite que un elemento se desplace respecto a lo que hubiera sido su posición normal; el resto de elementos continúan en su posición ignorando al que se desplaza, lo que puede crear superposiciones; el espacio libre que deja el elemento queda libre

```
.contenedor {  
  position: relative;  
}
```



### **absolute**

Permite que un elemento se desplace respecto al origen de coordenadas del primer elemento contenedor posicionado ó respecto a la esquina superior izquierda de la ventana de visualización; el resto de elementos actúan como si el desplazado no existiera, por lo que su espacio será ocupado por otros elementos; puede crear superposiciones.

```
.contenedor {  
  position: absolute;  
}
```

### **fixed**

Permite fijar un elemento en una posición respecto al origen de coordenadas del primer elemento contenedor posicionado ó respecto a la esquina superior de la ventana de visualización; el elemento se mantendrá en la ventana de visualización o viewport, siempre en una misma posición aunque el usuario se desplace por la web haciendo scroll.

```
.contenedor {  
  position: fixed;  
}
```

### **inherit**

Se heredan las características del elemento padre



```
.contenedor {  
  position: inherit;  
}
```

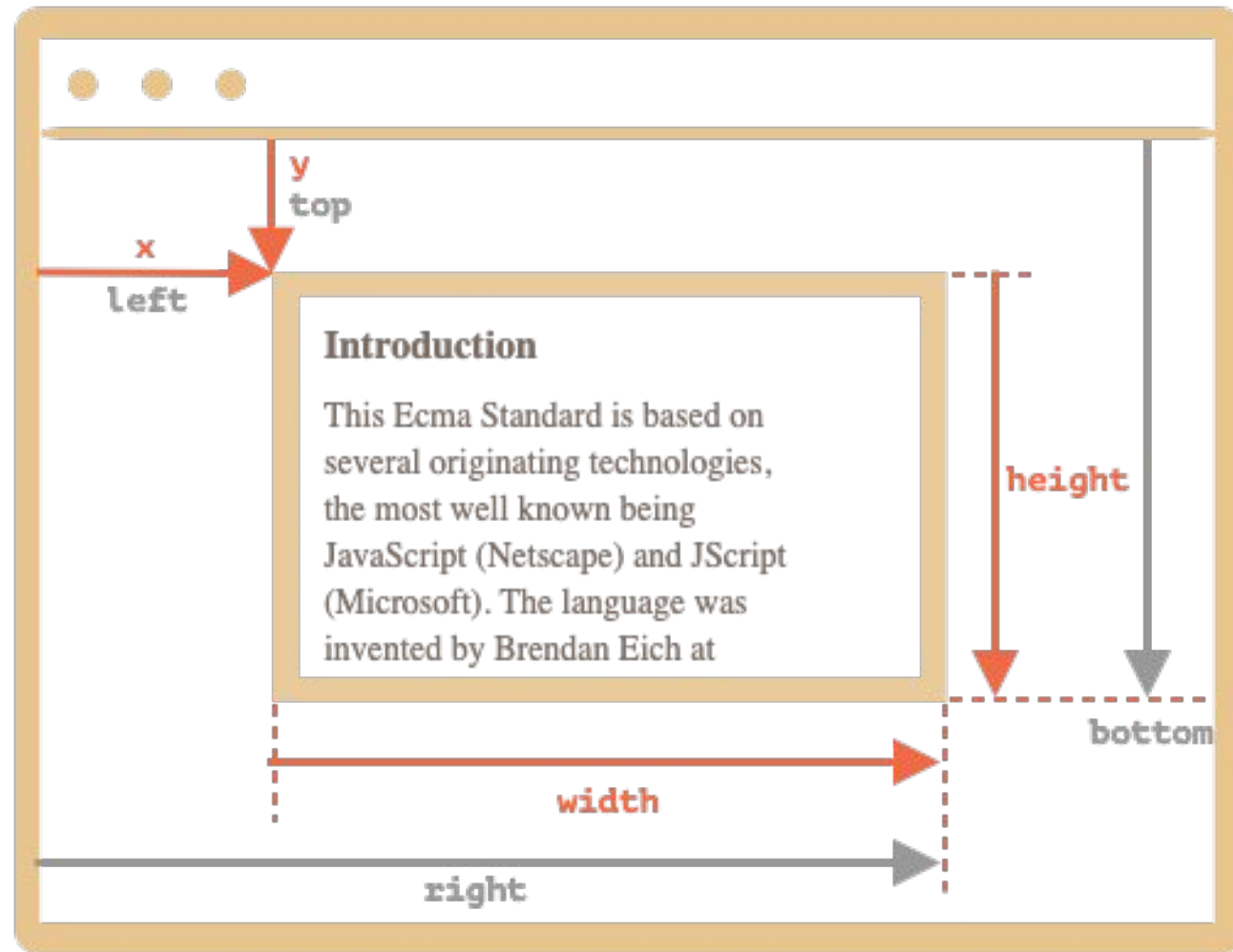
# Calcular position

1	2
3	4

La **posición absoluta** de un elemento que no está dentro de ningún otro se determina respecto a la **parte superior izquierda del elemento html** (parte superior izquierda de la ventana donde se visualiza la página web; normalmente coincide con la parte superior izquierda del elemento body).

Si un elemento está dentro de varios **contenedores** (p.ej. dentro de varios div) el origen de coordenadas **se establece respecto a la esquina superior izquierda del contenedor que tenga un valor de position establecido y distinto de static** o, en caso de no existir ningún elemento contenedor posicionado, respecto a la esquina superior izquierda del elemento html (parte superior izquierda de la ventana donde se ve la página web). Esto se comprenderá con los ejemplos de código que veremos.

## Position - Calcular



Top, left, right,  
bottom ?

## Position - Top, left, right, bottom

Estas propiedades nos permiten definir el desplazamiento de la posición de un elemento respecto a un origen de coordenadas y el origen de coordenadas que se toma.

```
.contenedor {  
  position: absolute;  
  top: 10px;  
  left: 20px;  
}
```



## Position - Top, left, right, bottom

Para elementos con **posición relative** estas propiedades definen cuánto se desplaza el elemento respecto a lo que sería su posición normal.

Para elementos con **posición absolute** o **posición fixed** estas propiedades definen cuánto se desplaza el elemento respecto a las coordenadas de origen.

Se admiten valores de desplazamientos negativos.

**No tiene sentido utilizar top y bottom al mismo tiempo, porque sería decir que el elemento sube y baja. Hay que utilizar sólo una de estas dos propiedades. Lo mismo podemos decir para right y left.**

# LINKS

- **Position**

- <https://css-tricks.com/almanac/properties/p/position/>
- <https://css-tricks.com/absolute-relative-fixed-positioning-how-do-they-differ/>



**GUSTAVO RODRIGUEZ**

FULL STACK DEVELOPER  
SOLCRE



[gustavgueez](#)



[gustavgueez](#)



[gustavgueez](#)