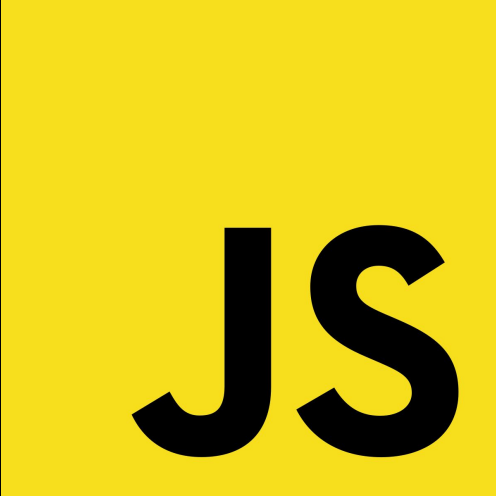


# La interfaz “Screen”

—

Sergio Moreno Antequera

A yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing the JavaScript logo.

¿Que es?

—

La interfaz “Screen”  
representa una pantalla,  
usualmente la misma  
donde la ventana actual  
está siendo visualizada.

# Propiedades y métodos

—

# Propiedades

Todas las propiedades de la  
interfaz

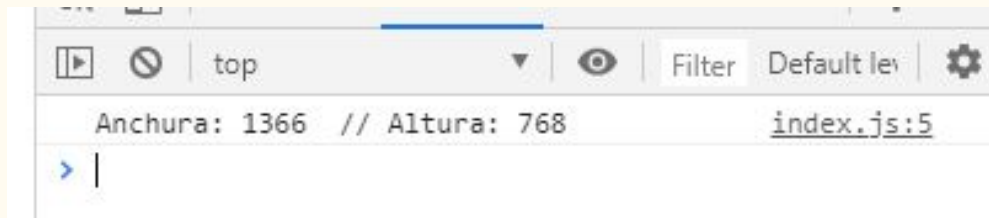
- height & width
  - availHeight & availWidth
  - screenTop & screenLeft
  - colorDepth
  - pixelDepth
  - orientation
-

# Height & Width

Estas propiedades contienen las dimensiones en píxeles de la pantalla en la cual se ha abierto la pestaña.

- Height -> La altura
- Width -> La anchura

```
//Para saber las dimensiones de nuestra pantalla en píxeles
function showScreenDimensions() {
  console.log("Anchura: " + screen.width + " // Altura: " + screen.height);
}
```

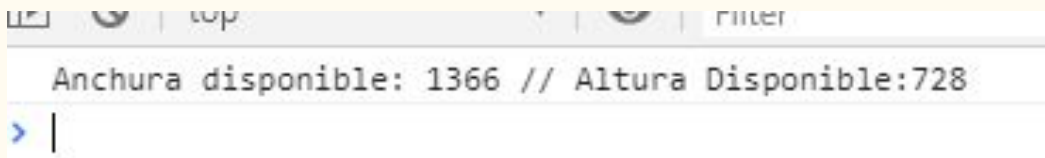


# AvailHeight & AvailWidth

Estas propiedades contienen las dimensiones en píxeles del espacio en el cual se puede trabajar en un navegador (Suele ser muy próximo a las dimensiones de la pantalla pero quitando la barra de búsqueda),

- AvailHeight -> La altura
- AvailWidth -> La anchura

```
//La diferencia entre este y el anterior es que nuestra página no cupa toda la pantalla
//Y por lo tanto por ejemplo la barra de busqueda ocupa un espacio, esto nos muestra el máximo posible
function showScreenAvailableDimensions(){
  console.log("Anchura disponible: " + screen.availWidth + " // Altura Disponible:" + screen.availHeight);
}
```

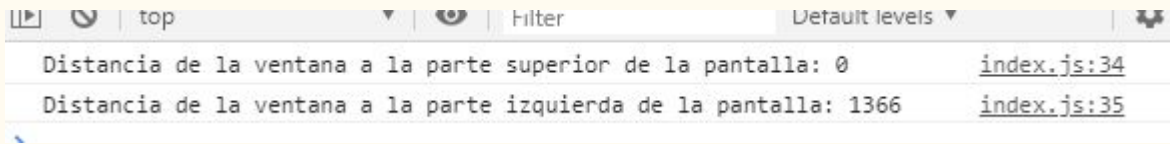


# ScreenTop & ScreenLeft

Estas propiedades tienen el valor en píxeles de la distancia de la esquina superior izquierda de nuestro navegador con la esquina superior izquierda de nuestra pantalla.

- ScreenTop -> La altura
- ScreenLeft -> La anchura

```
//Para saber la distancia de nuestra ventana al margen superior e izquierdo de la pantalla
function showDistanceToTheScreenSides() {
  console.log("Distancia de la ventana a la parte superior de la pantalla: " + screenTop);
  console.log("Distancia de la ventana a la parte izquierda de la pantalla: " + screenLeft);
}
```



Distancia de la ventana a la parte superior de la pantalla: 0	<a href="#">index.js:34</a>
Distancia de la ventana a la parte izquierda de la pantalla: 1366	<a href="#">index.js:35</a>

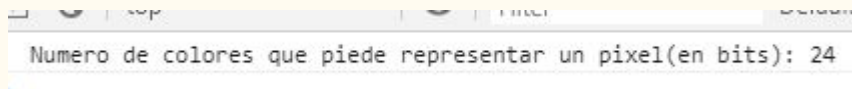


# ColorDeth

Esta propiedad nos devuelve el número en bits de la cantidad de colores que puede llegar nuestra pantalla.

```
/* Bit Depth    Number of Colours
1 bit    2 colours (usually black and white)
2 bits   4 colours
4 bits   16 colours
8 bit    greyscale 256 shades of grey
8 bit    colour    256 colours
16 bits   65, 536 colours (known as 'high' colour)
24 bits  16.7 million colours (known as 'true' colour)
32 bits  16.7million colours plus greyscale mask (alpha channel)*/
```

```
console.log("Numero de colores que piede representar un pixel(en bits): " + screen.colorDepth);
```

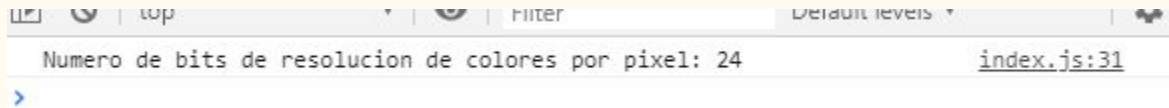


Numero de colores que piede representar un pixel(en bits): 24

# PixelDeth

De la mano con la anterior propiedad, esta nos informa de la misma manera, de la capacidad de pasar por colores intermedios antes de pasar a otro. Estas dos últimas propiedades suelen tener el mismo valor.

```
console.log("Numero de bits de resolucion de colores por pixel: " + screen.pixelDepth);
```



# Orientation.Type & Orientation.Angle

Para obtener esta información tendremos crear un objeto “Orientation” del siguiente modo:

- Type -> Como se llama la posición (landscape = horizontal // portrait = vertical)
- angle -> El ángulo de la rotación

```
//Informacion relacionados con la orientación de la pantalla
function showOrientationInfo(){
    //Primero conseguimos un objeto del tipo ScreenOrientation con el que trabajar
    var orientacion = screen.orientation;
    //Propiedades
    console.log("Tipo de orientacion: " + orientacion.type);
    console.log("Angulo que indica la rotacion: " + orientacion.angle);
}
```

Tipo de orientacion: landscape-primary	<a href="#">index.js:46</a>
Angulo que indica la rotacion: 0	<a href="#">index.js:47</a>
>	

# Métodos

Todos los métodos de la interfaz

- `orientation.unlock`
- `orientation.lock`



# Orientation.lock & Orientation.unlock

Estos métodos se encargan de hacer nuestra página responsiva o no a los cambios en la orientación de nuestra pantalla. Determinamos si al girar nuestra pantalla la ventana de nuestro navegador debería girar también. Esto es especialmente útil en dispositivos móviles por obvios motivos.

- Lock -> Desactivar la rotación de pantalla
- Unlock -> Activar la rotación de pantalla

```
//Informacion relacionados con la orientación de la pantalla
function showOrientationInfo(){
    //Primero conseguimos un objeto del tipo ScreenOrientation con el que trabajar
    var orientacion = screen.orientation;

    //Bloquean o desbloquean la rotacion de la página respectivamente
    orientacion.lock;
    orientacion.unlock;
}
```

¿Alguna pregunta?