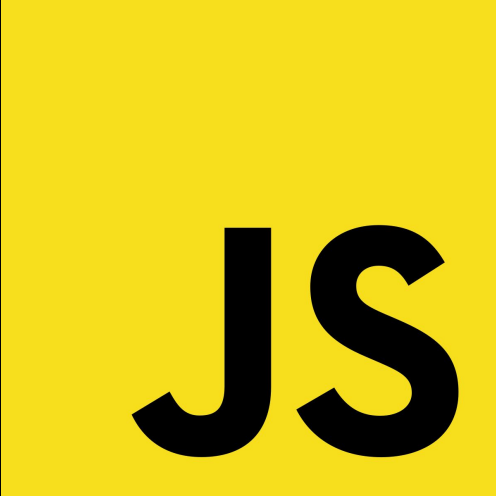


El objeto “Screen”

—

Sergio Moreno Antequera

The JavaScript logo, consisting of the letters 'JS' in a bold, black, sans-serif font, centered within a bright yellow square.

¿Que es?

—

Screen es un objeto
que se crea al cargar
la página, con todas
las propiedades ya
asignadas.

¿Para qué sirve?

—

Nos sirve para obtener
información de la
pantalla donde la
ventana actual está
siendo visualizada.

Propiedades y métodos

—

Propiedades

Todas las propiedades de la
interfaz

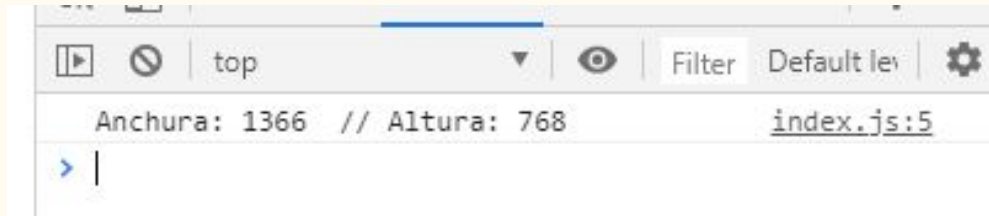
- height & width
 - availHeight & availWidth
 - screenTop & screenLeft
 - colorDepth
 - pixelDepth
 - orientation
-

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);
}
```

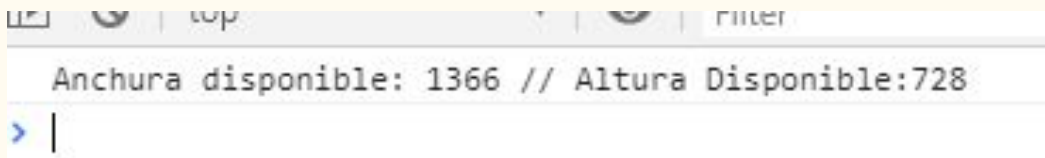


AvailHeight & AvailWidth

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);
}
```

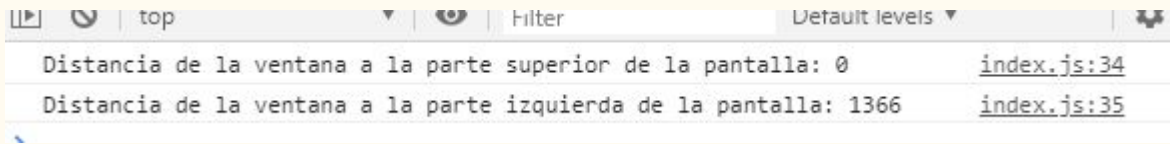


ScreenTop & ScreenLeft

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);
}
```



ColorDeth

Esta propiedad nos devuelve el número de bits necesarios para representar el color de un pixel en la 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 bits necesarios para representar un color: " + screen.colorDepth);
```

```
Numero de bits necesarios para representar un color: 24
```

PixelDeth

Esta propiedad nos informa de la cantidad de tonos de colores que nuestra pantalla nos permite representar en bits. Esto suele estar determinado por la habilidad de la pantalla de reproducir los colores primarios (RGB)



```
console.log("Numero de bits empleados en el cambio del tono de color: " + screen.pixelDepth);
```

```
Numero de bits empleados en el cambio del tono de color: 24
```

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	index.js:46
Angulo que indica la rotacion: 0	index.js:47
>	

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?