



null, undefined, NaN

- Casos especiales -

Undefined

Undefined

En primer lugar `undefined` es es un tipo de dato y por lo tanto se puede consultar con `typeof`

```
if (typeof variable === "undefined") {  
    console.log('ok');  
}
```

Existe una **variable global** que contiene este valor y que se denomina también `undefined`, por lo que podemos hacer comparaciones de este tipo:

```
if (variable === undefined) {  
    console.log('ok');  
}
```



Errores con Undefined

Errores con undefined

Comparar la variable `undefined` con el resultado de `typeof` NO TIENE SENTIDO. `typeof` siempre devuelve una cadena de texto.

```
if (typeof variable === undefined) {  
    console.log('ok'); // No se ejecuta  
}
```

`undefined` no es una palabra reservada de Javascript y por lo tanto podemos definir una variable con ese nombre (**MUY MALA IDEA**).

```
function test() {  
    var variable;  
    var undefined = 10;  
    if (variable === undefined) {  
        console.log('ok'); // No se ejecuta  
    }  
}
```

null

null

`null` es un literal definido en la especificación del lenguaje. Por tanto **ES** una palabra reservada. No podremos utilizarla como nombre de variable.

```
var variable = null;
if (variable === null) {
    console.log('ok');
}
```

`null` no dispone de un tipo específico y en el caso de preguntar con `typeof` sobre un elemento con valor `null` nos dirá que es un objeto.

```
var variable = null;
if (typeof variable === "object") {
    console.log('ok');
}
```

Que `null` es un objeto es algo peculiar y debemos tener en cuenta que una variable que dice ser un objeto realmente puede contener `null` y producir un error en nuestro código si lo lo validamos.



NaN

NaN (Not-A-Number)

NaN es un **valor** de tipo numérico, pero no podemos operar con él, ya que cualquier operación que se realice con NaN siempre devolverá NaN.

Normalmente se obtiene NaN como resultado de las operaciones con funciones y métodos de Math donde no ha sido posible convertir un valor a numérico o se ha producido algún error.

```
Math.sqrt(-1);  
parseInt('no');
```

El problema a la hora de comprobar si un valor es NaN es que si usamos `typeof` nos dirá que es un valor numérico y podemos considerar que tenemos entre manos un número válido. PERO NO. Para resolver el problema existe la función `isNaN()`.

```
var variable = parseInt('xxx');  
if (isNaN(variable)) {  
    console.log('Is NaN');    // Se ejecuta  
}
```



NaN – Más problemas

NaN – Más problemas

NaN **no** es una palabra reservada de Javascript y por lo tanto podemos definir una variable con ese nombre.

```
function test() {  
    var variable = 10;  
    var NaN = 10;  
    if (variable === NaN) {  
        console.log('ok'); // Se ejecuta  
    }  
}
```



END



prof.jduran@iesalixar.org