EXAMPLE 32: UNDERSTANDING `null`

EXAMPLE 32: UNDERSTANDING `null`

`null` IS A SPECIAL VALUE, SPECIFICALLY ASSIGNED TO A VARIABLE

`null` IS USED IN SOME SPECIFIC SITUATIONS, SUCH AS WHILE ACCESSING THE HTML OF A WEBPAGE.

```
if(someVariable == null)
```

`null` IS USED IN SOME SPECIFIC SITUATIONS, SUCH AS WHILE ACCESSING THE HTML OF A WEBPAGE.

THE MOST COMMON USE OF JAVASCRIPT IS TO MODIFY THE HTML OF A WEBPAGE USING SOMETHING CALLED THE DOM

DOM = "DOCUMENT-OBJECT-MODEL"

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

THEN, THE APPEARANCE OF THE HTML CAN BE MODIFIED BY ALTERING THE PROPERTIES OF THE "HTML" ELEMENT OBJECT.

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

THE ENTIRE HTML DOCUMENT IS TREATED AS ONE BIG OBJECT CALLED `document`

THIS `document` OBJECT IS AVAILABLE, BY DEFAULT, INSIDE THE SCRIPT ELEMENTS (WHERE ALL JAVASCRIPT CODE RESIDES)

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

ONE BIG OBJECT CALLED document

THIS document OBJECT IS AVAILABLE, BY DEFAULT, INSIDE THE SCRIPT ELEMENTS (WHERE ALL JAVASCRIPT CODE RESIDES)

ERRM - VERY VERY INTERESTING - BUT WHAT DOES THIS HAVE TO DO WITH null?

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

THE ENTIRE HTML DOCUMENT IS TREATED AS ONE BIG OBJECT CALLED `document`

THIS `document` OBJECT IS AVAILABLE, BY DEFAULT, INSIDE THE SCRIPT ELEMENTS (WHERE ALL JAVASCRIPT CODE RESIDES)

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

```
<script id = "code">

var code = document.getElementById("code
function nullExample() {
  var code = document.getElementById("c
  if(code != null) {
    console.log("The code in this webpa
contains this html: " + code.innerHTML)
  }
  else {
    console.log("code is null!");
  }
```

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

"code"

```
<script id =          >
    var code = document.getElementById(        "cod
    function nullExample() {
        var code = document.getElementById("co
        if(code != null) {
            console.log("The code in this webpa
    contains this html: " + code.innerHTML)
        }
        else {
            console.log("code is null!");
        }
```

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

```
        var code = document.getElementById("cod
console.log("The code in this webpage contains this html: "
code.innerHTML);
```

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

```
        var code = document.getElementById("cod
console.log("The code in this webpage contains this html: "
code.innerHTML);
```

DOM = "DOCUMENT-OBJECT-MODEL"

THE DOM IS A WAY OF ACCESSING ANY PART OF THE HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

```
    var code = document.getElementById("cod
console.log("The code in this webpage contains this html: "
code.innerHTML);
```

WE CAN NOW ACCESS THE HTML VIA AN OBJECT PROPERTY!!

DOM =

THE DOM IS A WAY OF ACCESSING ANY PART OF THE
HTML OF A WEBPAGE AS A JAVASCRIPT OBJECT.

var code = document.getElementById(

console.log(
code.innerHTML

**ERRM - VERY VERY INTERESTING - BUT
WHAT DOES THIS HAVE TO DO WITH `null`?**

WE CAN NOW ACCESS THE HTML VIA AN
OBJECT PROPERTY!!

ERRM - VERY VERY INTERESTING - BUT WHAT DOES THIS HAVE TO DO WITH `null`?

```
var code = document.getElementById("co
```

IF THERE IS NO HTML ELEMENT WITH ID = "CODE", THEN `document.getElementById` RETURNS `null`

document.getElementById RETURNS null

BECAUSE document.getElementById IS ONE OF THE MOST IMPORTANT, AND MOST COMMONLY USED FUNCTIONS IN ALL JAVASCRIPT, null IS PRETTY IMPORTANT TO UNDERSTAND

null IS PRETTY

CASE #1: YOU CALL THE **typeof** OPERATOR ON SOMETHING THAT IS **null** YOU GET - **object**

```
console.log("Type of null is ... wait for it.. " + (typeof nullVariable));
```

Type of null is ... wait for it.. object

CASE #2: YOU COMPARE undefined AND null YOU GET - true