

**EXAMPLE 45: CLOSURES ARE WAY COOLER
THAN GLOBAL VARIABLES**

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)
```

RECAP

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"

RECAP

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.**

RECAP

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)

RECAP

DOM = "DOCUMENT-OBJECT-MODEL"

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

```
<script id = "code">
```

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

RECAP

DOM = "DOCUMENT-OBJECT-MODEL"

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

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

RECAP

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("code")
```

```
console.log("The code in this webpage contains this html: " +  
code.innerHTML);
```

RECAP

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("code")
```

```
console.log("The code in this webpage contains this html: " +  
code.innerHTML);
```

RECAP

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("code");  
console.log("The code in this webpage contains this html:"  
code.innerHTML);
```

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

RECAP

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

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

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

RECAP

EXAMPLE 45: CLOSURES ARE WAY COOLER THAN GLOBAL VARIABLES

We all love Closures!

**Closures are SO cool, I just love
closures.**

Agree?

Yes!

No!

We all love Closures!

Closures are SO cool, I just love closures.

Agree?

Yes!

No!

**LET'S BUILD THE CODE FOR THIS LITTLE POLL
IN 2 WAYS.**

LET'S BUILD THE CODE FOR THIS LITTLE POLL IN 2 WAYS.

We all love Closures!

**Closures are SO cool, I just love
closures.**

Agree?

THE "NO" BUTTON WILL KEEP TRACK OF THE
NO VOTES WITHOUT USING CLOSURES

LET'S BUILD THE CODE FOR THIS LITTLE POLL IN 2 WAYS.

We all love Closures!

**Closures are SO cool, I just love
closures.**

Agree?



Yes! No!

**THE "YES" BUTTON WILL KEEP TRACK OF
THE YES VOTES USING A CLOSURE**

LET'S BUILD THE CODE FOR THIS LITTLE POLL IN 2 WAYS.

We all love Closures!

Closures are SO cool, I just love

closures.

**THIS IS THE FIRST TIME WE WILL USE JAVASCRIPT
TO DO SOMETHING MEANINGFUL WITH THE HTML.**

Agree?

Yes!

No!

LET'S BUILD THE CODE FOR THIS LITTLE POLL IN 2 WAYS.

We all love Closures!

**Closures are SO cool, I just love
closures.**

Agree?



Yes! No!

**THE "YES" BUTTON WILL KEEP TRACK OF
THE YES VOTES USING A CLOSURE**

WE HAVE 2 BUTTONS IN THE HTML



```
<button id = "yesButton"> Yes! </b>  
<button id = "noButton"> No! </but
```

EACH HAS AN ID THAT WE CAN USE FROM THE
JAVASCRIPT USING `document.getElementById`

```
window.onload = function() {
```

```
    var yesButton = document.getElementById("yesButton");  
    var noButton = document.getElementById("noButton");  
    var yesVotes = 0;
```

```
    noButton.onclick = countNoVotes;
```

```
    yesButton.onclick = function() {  
        yesVotes++;  
        var yesVotesMessage =  
document.getElementById("yesVotes");  
        yesVotesMessage.innerHTML = "# Yes votes = <b>" +  
yesVotes + "</b><br/>";  
    }  
}
```

Yes!

No!

```
function countNoVotes() {  
    noVotes++;  
    var noVotesMessage = document.getElementById("noVotes");  
    noVotesMessage.innerHTML = "# No votes = <b>" + noVotes +  
"</b><br/>";  
}
```

```
var yesButton = document.getElementById("yesButton");  
var noButton = document.getElementById("noButton");  
var yesVotes = 0;
```

**WE HAVE A LOCAL (NOT GLOBAL) VARIABLE TO
COUNT THE NUMBER OF YES VOTES**

```
yesButton.onclick = function() {  
    yesVotes++;  
    var yesVotesMessage =  
document.getElementById("yesVotes");  
    yesVotesMessage.innerHTML = "# Yes votes = <b>" +  
yesVotes + "</b><br/>";  
}
```

**AND A NESTED FUNCTION THAT DISPLAYS A
MESSAGE WHEN SOMEONE VOTES "YES"**

```
function countNoVotes() {  
    noVotes++;  
    var noVotesMessage = document.getElementById("noVotes");  
    noVotesMessage.innerHTML = "# No votes = <b>" + noVotes  
"</b><br/>";  
}
```

```

var yesButton = document.getElementById("yesButton");
var noButton = document.getElementById("noButton");
var yesVotes = 0;

noButton.onclick = countNoVotes;

yesButton.onclick = function() {
    yesVotes++;
    var yesVotesMessage =
document.getElementById("yesVotes");
    yesVotesMessage.innerHTML = "# Yes votes = <b>" +
yesVotes + "</b><br/>";
}

}

function countNoVotes() {
    noVotes++;
    var noVotesMessage = document.getElementById("noVotes");
    noVotesMessage.innerHTML = "# No votes = <b>" + noVotes
"</b><br/>";

```

CLOSURE

=

SAY WE HAVE A **NESTED**

+

**VARIABLES LOCAL TO THE
OUTER SCOPE**

"REFERENCING"

```
var yesButton = document.getElementById("yesButton");
var noButton = document.getElementById("noButton");
```

**THE "YES" BUTTON WILL KEEP TRACK OF
THE YES VOTES USING A CLOSURE**

```
yesButton.onclick = function() {
  yesVotes++;
  var yesVotesMessage =
document.getElementById("yesVotes");
  yesVotesMessage.innerHTML = "# Yes votes = <b>" +
yesVotes + "</b><br/>";
}
```

CLOSURE =

**SAY WE HAVE A NESTED +
VARIABLES LOCAL TO THE
OUTER SCOPE
"REFERENCING"**

```
function countNoVotes() {
  noVotes++;
  var noVotesMessage = document.getElementById("noVotes");
  noVotesMessage.innerHTML = "# No votes = <b>" + noVotes
"</b><br/>";
}
```


LET'S BUILD THE CODE FOR THIS LITTLE POLL IN 2 WAYS.

We all love Closures!

**Closures are SO cool, I just love
closures.**

Agree?



Yes! No!

**THE "YES" BUTTON WILL KEEP TRACK OF
THE YES VOTES USING A CLOSURE**

LET'S BUILD THE CODE FOR THIS LITTLE POLL IN 2 WAYS.

We all love Closures!

**Closures are SO cool, I just love
closures.**

Agree?

THE "NO" BUTTON WILL KEEP TRACK OF THE
NO VOTES WITHOUT USING CLOSURES

THE "NO" BUTTON WILL KEEP TRACK OF THE NO VOTES WITHOUT USING CLOSURES

```
var noVotes = 0;
```

```
window.onload = function() {
```

```
    var yesButton = document.getElementById("yesButton");
```

```
    var noButton = document.getElementById("noButton");
```

```
    var yesVotes = 0;
```

```
    noButton.onclick = countNoVotes;
```

```
    yesButton.onclick = function() {
```

```
        yesVotes++;
```

```
        var yesVotesMessage = document.getElementById("yesVotes");
```

```
        yesVotesMessage.innerHTML = "# Yes votes = <b>" + yesVotes +  
>";
```

```
    }
```

```
}
```

```
function countNoVotes() {
```

```
    noVotes++;
```

```
    var noVotesMessage = document.getElementById("noVotes");
```

```
    noVotesMessage.innerHTML = "# No votes = <b>" + noVotes + "</b>"
```

```
}
```

```
var noVotes = 0;
```

**THE "NO" BUTTON WILL KEEP TRACK OF
THE NO VOTES WITHOUT USING CLOSURES**

```
var yesButton = document.getElementById("yesButton");
var noButton = document.getElementById("noButton");
var yesVotes = 0;

noButton.onclick = countNoVotes;

yesButton.onclick = function() {
    yesVotes++;
    var yesVotesMessage = document.getElementById("yesvotes");
    yesVotesMessage.innerHTML = "# Yes votes = <b>" + yesVotes + "</b><br/>";
};
```

**TO DO THIS WITHOUT CLOSURES, WE
END UP NEEDING A GLOBAL VARIABLE!**

```
function countNoVotes() {
    noVotes++;
    var noVotesMessage = document.getElementById("noVotes");
    noVotesMessage.innerHTML = "# No votes = <b>" + noVotes + "</b><br/>";
}
```

We all love Closures!

Closures are SO cool, I just love closures.

Agree?

Yes votes = 20

No votes = 1

We all love Closures!

Closures are SO cool, I just love closures.

Agree?

Yes!

No!

Yes votes = 20

No votes = 1