1) Operational precedence

Operator precedence describes an order in which the operations in an expression are evaluated.

Certain operators have higher level of priority compared to other operators.

Multiplication has higher precedence than addition.

In JavaScript, if two or more operators with same precedence appear in an expression, the concept of

Operator precedence describes which one is evaluated first.

EXAMPLE

10 + 3 – 4 + 2 \* 3 + 5

10 + 3 – 4 + 6 + 5

13 – 4 + 6 + 5

9 + 6 + 5

15 + 5

20

2) Differences between == and ===

Differences between != and !==

== is a comparison operator basically checks the values for equality.

== Operator don’t check the type, it only compares the value.

EXAMPLE

<html>

<body>

<p>Assign 5 to x, and display the value of the comparison (x == 5):</p>

<p id="demo"></p>

<script>

var x = 5;

document.getElementById("demo").innerHTML = (x == 5);

</script>

</body>

</html>

Output: true

-----🡪

<html>

<body>

<p>Assign 5 to x, and display the value of the comparison (x == 5):</p>

<p id="demo"></p>

<script>

var x = 5;

document.getElementById("demo").innerHTML = (x == "5");

</script>

</body>

</html>

Output: true

=== operator checks for equal value and also equal type

<html>

<body>

<p>Assign 5 to x, and display the value of the comparison (x === 5):</p>

<p id="demo"></p>

<script>

var x = 5;

document.getElementById("demo").innerHTML = (x === 5);

</script>

</body>

</html>

OUTPUT: TRUE

<html>

<body>

<p>Assign 5 to x, and display the value of the comparison (x === 5):</p>

<p id="demo"></p>

<script>

var x = 5;

document.getElementById("demo").innerHTML = (x === "5");

</script>

</body>

</html>

OUTPUT: FALSE

!= operator checks for Inequality, doesn’t check the type but just the value.

EXAMPLE:

<html>

<body>

<p>Assign 5 to x, and display the value of the comparison (x != 8).</p>

<p id="demo"></p>

<script>

var x = 5;

document.getElementById("demo").innerHTML = (x != 8);

</script>

</body>

</html>

OUTPUT: TRUE

<html>

<body>

<p>Assign 5 to x, and display the value of the comparison (x != “5”).</p>

<p id="demo"></p>

<script>

var x = 5;

document.getElementById("demo").innerHTML = (x != “5”);

</script>

</body>

</html>

OUTPUT: FALSE

!== Non Identity operator checks for Inequality for both value and data type i.e., returns true if not equal value or not equal type

EXAMPLE

<html>

<body>

<p>Assign 5 to x, and return the value of the comparison (x !== 5):</p>

<p id="demo"></p>

<script>

var x = 5;

document.getElementById("demo").innerHTML = (x !== 5);

</script>

</body>

</html>

OUTPUT: FALSE

<html>

<body>

<p>Assign 5 to x, and display the value of the comparison (x !== "5"):</p>

<p id="demo"></p>

<script>

var x = 5;

document.getElementById("demo").innerHTML = (x !== "5");

</script>

</body>

</html>

OUTPUT: TRUE

3) diff null and undefined

|  |  |
| --- | --- |
| Undefined | Null |
| Undefined is a special type used when an object cannot be assigned to a variable. It means a variable has been defined but has not been assigned any value yet. | Null is an assignment value that can be assigned to a variable for representing no value. |
| By default, unassigned variables are considered as undefined. | Setting a value to null must be done programmatically. |

4) loop (for, while and do while) and conditons (if else, switch).

For Loop: Iterates through the block a number of times.

for (var i = 0; i < 3; i++) {

console.log(i);

}

While loop: As long as the specified condition is true, while loop loops through the block of code repeatedly. Once the expression becomes false, the loop terminates.

EXAMPLE:

var score=0;

while(score<10){

document.write(“The current score is : “ + score+ “<br/>”);

score++;

}

Do while loop: The Do while loop executes a statement until the condition evaluates to false.

The condition is evaluated after executing the statement. So, the loop is executed at least once.

EXAMPLE:

var i =0;

do {

console.log(i);

i++;

}while(i<10);

for(var i=0; i<10; i++) {

console.log(i);

}

IF ELSE:

It is a conditional statement.

EXAMPLE:

Var age=15;

if(age<18)

{

document.write(“Eligible for working”);

}

else

{

document.write(“Not eligible for working”);

}

SWITCH:

Var grade=’A’;

Switch(grade)

{

Case ‘A’: document.write(“Good job”);

Break;

Case ‘B’: document.write(“passed”);

Break;

default: document.write(“No Grade”);

}