

FSD-Assignment-2

1.Explain the difference between null and undefined in JavaScript.

Answer: The difference between null and undefined is the null value does not take any memory space but undefined will take the memory space.

undefined means a variable has been declared but hasn't been given a value yet

null is a value that you can assign to a variable to show it has no value or is empty.

2.What will be the output of the following code snippet, and why?



```
console.log('10' + 5);  
console.log('10' - 5);  
console.log(true + 2);  
console.log(false + undefined);
```

Answer: 105,5,3,NaN

Explanation:

- for '10' +5 the 10 is a string and 5 number so in strings the values will concatenate each other when using + operator.
- '10'-5 in this case 10 will convert to a number when we perform different operations using operators accept + operator.
- true + 2 the boolean operator true values is 1 so, 1+2 it will give you 3.
- False + undefined it gives you a NaN because the false value is 0 and undefined is not a numeric context.

3.What is the difference between == and === in JavaScript? Provide examples.

Answer: when we are using == the value will be converted and check whether the value is equal or not. Example: I have given "10"==10 so here the string value will convert to number and check whether both are equal or not. So here the 10 ==10 so it returns true.

When we are using === the values will not convert the values will be the same and check equals or not. Example: I have given "10"===10 so here the string value will not convert to the number so it compares the values a string and number both are not equal so it returns false.

4.Predict the output of the following expressions and explain your reasoning:



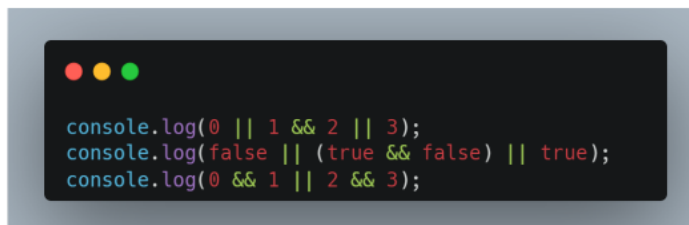
```
console.log(0 == false);
console.log(0 === false);
console.log('' == 0);
console.log('' === 0);
```

Answers: true,false,true,false

Explanation:

- in the 0==false the false value will convert to a number which is 0 so 0==0 so it is true.
- 0===false in this the value of false will not be converted to number so the false will be the boolean so 0===false which is false.
- ""==0 the "" which is the string the length of the string is 0 so 0==0 which is true.
- ""===0 which is false because the "" empty string will not convert to number so it is false.

5. Given the following code, what will be the output and why?



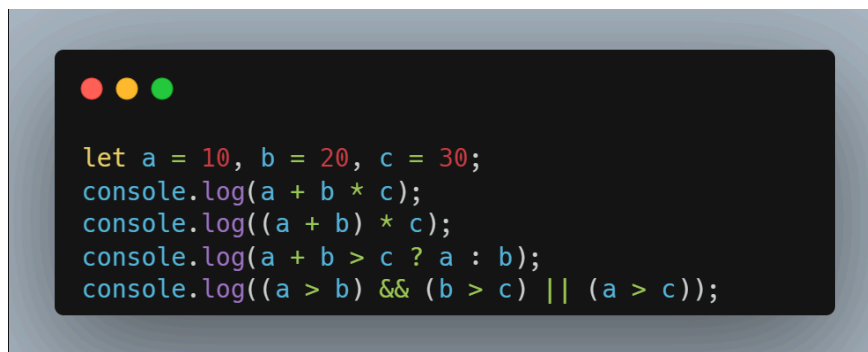
```
console.log(0 || 1 && 2 || 3);
console.log(false || (true && false) || true);
console.log(0 && 1 || 2 && 3);
```

Answer: 2,true,3

Explanation:

- The && operator is evaluated first, giving 2. Then, 0 is falsy, so 2 is returned as it's the first truthy value.
- The && inside the parentheses evaluates to false, then false || false moves to the next value, which is true, so true is returned.
- The && operators are evaluated first, giving 0 and 3. The || then returns 3, the first truthy value after 0.

6. Predict the output of the following expressions and explain your reasoning



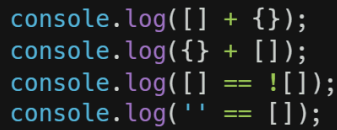
```
let a = 10, b = 20, c = 30;
console.log(a + b * c);
console.log((a + b) * c);
console.log(a + b > c ? a : b);
console.log((a > b) && (b > c) || (a > c));
```

Answer: 610,900,20,false

Explanation:

- The multiplication * operator has a higher precedence than the + operator so first it will calculate 20*30 so it is 600 and then + which is 10 so it will be 610.
- The parentheses will force to do the addition first so 10 + 20 which is 30 so 30*30 which is 900.
- The ternary operator is the same as a conditional statement which is like if and else if the first is true it returns a value else return b so 10+20>30 which is false so it returns 20.
- The && operator is evaluated first, checking if a > b and b > c. Then, the || operator checks if either the && result is true or a > c is true, returning true if any part is true.

7. Analyze and explain the output of the following code snippets



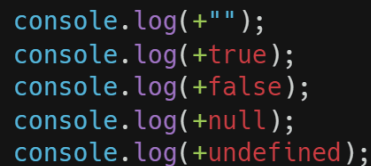
```
console.log([] + {});  
console.log({} + []);  
console.log([] == ![]);  
console.log('' == []);
```

Answer: [object,object],[object,object],true, true

Explanation:

- In java script [] and {} both are objects so it prints objects.
- In java script {} and [] both are objects so it prints objects.
- first evaluates ![] to false because an empty array is truthy, so ![] becomes false. Then the comparison becomes [] == false, which is true because [] is converted to an empty string "", and false is also converted to an empty string when compared with ==.

8.What will be the output of the following code, and why?



```
console.log(+ "");  
console.log(+true);  
console.log(+false);  
console.log(+null);  
console.log(+undefined);
```

Answer: 0,1,0,0,NaN

Explanation:

- "" The string is empty so it returns 0.
- The true value is 1 so it return 1
- The false value is 0 so it print 0
- The null means empty so it prints 0
- The undefined it will print NaN were it is not a numeric value