Conditionals

1. What will be the output of this code:

```
if (5 > 10) {
  console.log("5 is greater than 10");
} else {
  console.log("5 is less than or equal to 10");
}
```

Output:

5 is less than or equal to 10

Explanation: The condition 5 > 10 is false, so the else block executes.

2. What will be the output of this code:

```
if (NaN === NaN) {
  console.log("NaN is equal to NaN");
} else {
  console.log("NaN is not equal to NaN");
}
```

Output:

NaN is not equal to NaN

Explanation: In JavaScript, NaN is not equal to any value, including itself. Thus, NaN === NaN is false.

3. What will be the output of this code:

```
if (!null) {
  console.log("null is truthy");
} else {
  console.log("null is falsy");
}
```

Output:

null is truthy

Explanation: null is falsy, so !null is true. The if block executes.

4. What will be the output of this code:

```
if (0 === "") {
  console.log("0 is equal to empty string");
} else {
  console.log("0 is not equal to empty string");
}
```

Output:

0 is not equal to empty string

Explanation: 0 (number) is not strictly equal to '' (empty string).

5. What will be the output of this code:

```
if (true && false) {
  console.log("true and false is true");
} else {
  console.log("true and false is false");
}
```

Output:

true and false is false

Explanation: The && operator returns false if any operand is false. Here, true && false is false.

6. What will be the output of this code:

```
if (false || true) {
  console.log("false or true is true");
} else {
  console.log("false or true is false");
}
```

Output:

false or true is true

Explanation: The || operator returns true if any operand is true. Here, false || true is true.

7. What will be the output of this code:

```
if (!false && true) {
  console.log("not false and true is true");
} else {
  console.log("not false and true is false");
}
```

Output:

not false and true is true

Explanation: !false is true, and true && true is true.

8. What will be the output of this code:

```
if (5 === "5") {
  console.log("5 is equal to string 5");
} else {
  console.log("5 is not equal to string 5");
}
```

Output:

5 is not equal to string 5

Explanation: 5 (number) is not strictly equal to '5' (string).

9. What will be the output of this code:

```
if (undefined == null) {
  console.log("undefined is equal to null");
} else {
  console.log("undefined is not equal to null");
}
```

Output:

undefined is equal to null

Explanation: undefined is loosely equal to null.

10. What will be the output of this code:

```
if (typeof [] === "object") {
  console.log("Array is an object");
} else {
```

```
  console.log("Array is not an object");
}
```

Output:

Array is an object

Explanation: In JavaScript, arrays are considered objects. typeof [] returns 'object'.