

Boolean Logic

True, False, and Beyond

Boolean Logic

- Everything starts with the idea that a statement is either True or False
- Then we can combine those initial statements to create more complex statements that also evaluate to True or False

Comparison Operators

Assuming x = 5

Operator	Name	Example	Result
>	Greater than	x > 10	false
>=	Greater than or equal to	x >= 5	true
<	Less than	x < -50	false
<=	Less than or equal to	x <= 100	true
==	Equal to	x == "5"	true
!=	Not equal to	x != "b"	true
===	Equal value and type	x === "5"	false
!==	Not equal value or equal type	x !== "5"	true

Equality Operators

== vs. ===

```
var x = 99;  
  
x == "99" //true  
x === "99" //false  
  
var y = null;  
  
y == undefined //true  
y === undefined //false
```

"==" performs *type coercion*, while "===" does not

A Few Interesting Cases

```
true == "1"           //true  
0 == false            //true  
null == undefined    //true  
NaN == NaN           //false
```

Logical Operators

AND, OR, and NOT

Operator	Name	Example	Result
&&	AND	<code>x < 10 && x !== 5</code>	false
	OR	<code>y > 9 x === 5</code>	true
!	NOT	<code>!(x === y)</code>	true

Assuming `x = 5` and `y = 9`

Exercise 1

```
var x = 10;  
var y = "a"
```

```
y === "b" || x >= 10
```

Exercise 2

```
var x = 3;
```

```
var y = 8;
```

```
!(x == "3" || x === y) && !(y != 8 && x <= y)
```


Truthy and Falsy Values

Values that aren't actually *true* or *false*, are still inherently "truthy" or "falsey" when evaluated in a boolean context

Try These Examples:

```
!"Hello World"
```

```
!" "
```

```
!null
```

```
!0
```

```
!-1
```

```
!NaN
```

Truthy and Falsy Values

Falsy Values:

- false
- 0
- ""
- null
- undefined
- NaN

Everything Else Is Truthy

Exercise 3

```
var str = ""  
var msg = "haha!"  
var isFunny = "false"  
  
!(( str || msg ) && isFunny)
```