



Topic 9: JavaScript

Comparison Operators and
Logical Operators

Lecture Contents

- boolean type
- Comparison Operators
- The `if` and `if...else` statements
- Logical Operators

The boolean Type

- The boolean type can take only two values, namely:
 - true
 - false
- Declaration (and initialization) of a boolean variable

```
var myBool; // declaration
var anotherBool = false; //declaration & initialization
```
- Note: in this case `true` is not a string

```
true != "true"
```

Lecture Contents

- boolean type
- ***Comparison Operators***
- The `if` and `if...else` statements
- Logical Operators

Operator Types

- *Arithmetic*
- *Assignment*
- ***Comparison (Relational / Equality)***
- *Logical*
- *Bitwise*

JavaScript *Comparison* Operators

- Comparison operators return a boolean value (`true` or `false`)

Symbol	Name
<code>==</code>	equal to
<code>!=</code>	not equal to
<code>===</code>	strict equal to
<code>!==</code>	strict not equal to
<code>></code>	greater than
<code>>=</code>	greater than or equal to
<code><</code>	less than
<code><=</code>	less than or equal to

Strict equality always considers operands of different types to be different

JavaScript *Comparison* Operators

- Type each of the following expressions into a console to find the truth value:

`5 < 6`

`5 > 6`

`true == "true"`

`5 == "5"`

`5 === "5"`

`5 === 5.0`

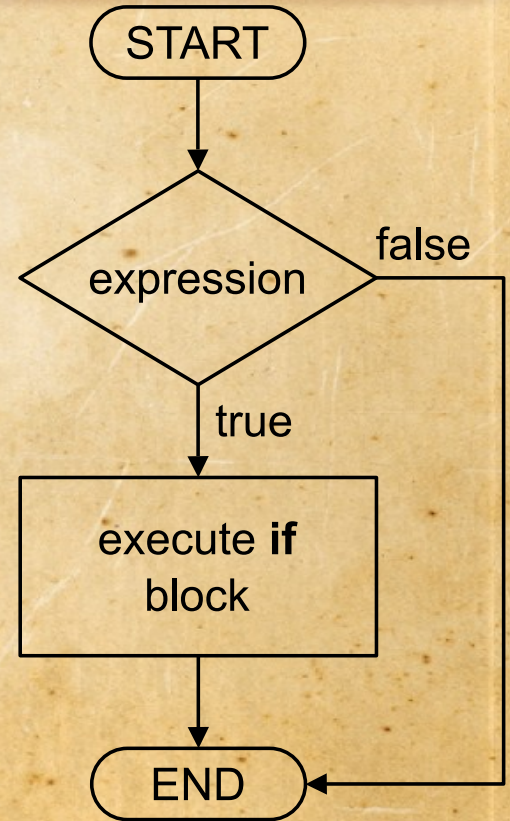
`true === (5<6)`

Lecture Contents

- boolean type
- Comparison Operators
- ***The `if` and `if...else` statements***
- Logical Operators

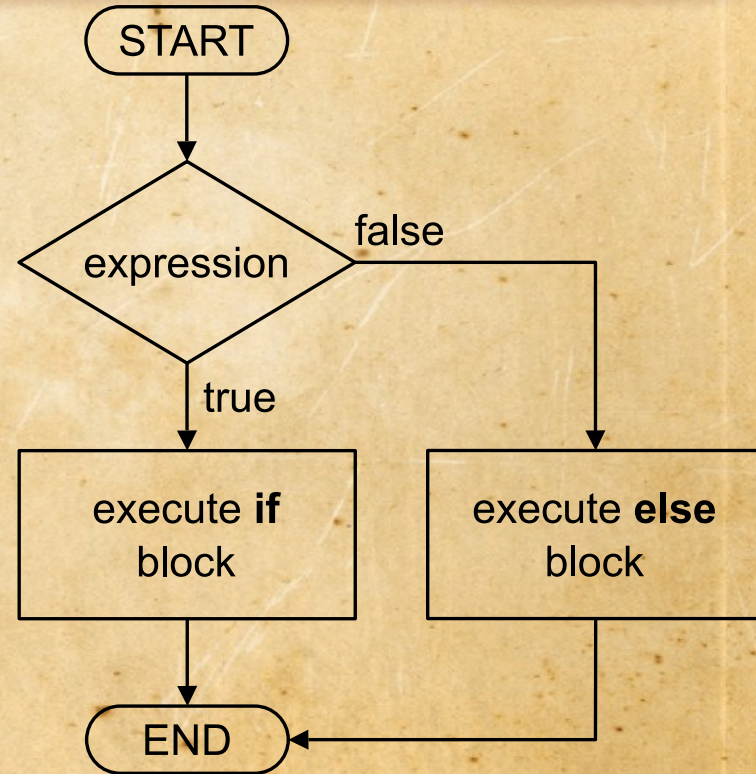
JavaScript `if` statement

```
var expr = (5+6 == 11);  
if(expr) {  
    console.log("expression is true");  
}
```



JavaScript `if...else` statement

```
var expr = (5+6 == 11);  
if(expr) {  
    console.log("expression is true");  
}  
else {  
    console.log("expression is false");  
}
```



JavaScript *Operator Precedence*

Level	Description	Operators	Associativity
18	parentheses	()	N/A
15	post inc/dec	++ --	N/A
14	pre inc/dec	++ --	N/A
14	unary	! ~ + -	N/A
13	exponentiation	**	Right-to-Left
12	multiplicative	* / %	Left-to-Right
11	additive	+ -	Left-to-Right
9	relational	> >= < <=	Left-to-right
8	equality	== != === !==	Left-to-right
2	assignment	: = += -= *= /= %= **=	Right-to-Left

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Operator_precedence

Lecture Contents

- boolean type
- Comparison Operators
- The `if` and `if...else` statements
- ***Logical Operators***

JavaScript *Logical Operators*

Symbol	Name
!	Logical NOT
&&	Logical AND
	Logical OR

Truth Tables

Input	!
false	true
true	false

Inputs		&&
false	false	false
false	true	false
true	false	false
true	true	true

Inputs		
false	false	false
false	true	true
true	false	true
true	true	true

Truth Tables

Input	!
0	1
1	0

Inputs		&&
0	0	0
0	1	0
1	0	0
1	1	1

Inputs		
0	0	0
0	1	1
1	0	1
1	1	1

JavaScript *Operator Precedence*

18	parentheses	()	N/A
15	post inc/dec	++ --	N/A
14	pre inc/dec	++ --	N/A
14	unary	! ~ + -	N/A
13	exponentiation	**	Right-to-Left
12	multiplicative	* / %	Left-to-Right
11	additive	+ -	Left-to-Right
9	relational	> >= < <=	Left-to-right
8	equality	== != === !==	Left-to-right
4	logical AND	&&	Left-to-right
3	logical OR		Left-to-right
2	assignment	: = += -= *= /= %= **=	Right-to-Left

JavaScript Practicing Conditionals

- Write a JavaScript function, `isTeenager ( )` that takes a number, `age`, as a parameter, and returns `true` if the age is between 13 and 19 (inclusive), and returns `false` if it is outside of this age range.
- Write a second JavaScript function, `printIsTeenager ( )` that takes a number, `age`, as a parameter, calls the method `isTeenager ( )` and prints a message stating whether a person of that age is a teenager or not. The method should conform to these example outputs:

Age 21 is not a teenager.

Age 19 is a teenager.



Topic 9: JavaScript

Comparison Operators and
Logical Operators