

Web Engineering Front-end Pt. 3

2. JavaScript: Introduction II





Revision



Revision



- Place JavaScript code in <script> tags.
- Display data to the browser console using console.log()

Contents



1. JavaScript Syntax
2. Variables
3. Arithmetic Operators
4. Comments



2.1 JavaScript Syntax



What is syntax?



- In programming, syntax is the set of rules that determine how program sentences are structured.
- Different programming languages will have different syntaxes.

Statements



- A program is a list of “instructions” to be executed by a computer.
- Each “instruction” is known as a statement.
- A JavaScript program is a list of these statements.

Statements



When a JavaScript program is run, the statements are executed one by one, from top to bottom.

Statements



A JavaScript statement can consist of...

- Values (Literals and variables)
- Operators
- Expressions
- Keywords
- Comments

Discussion: Why is syntax important?



What would happen if syntax for programming languages didn't exist?

2.2 Variables



Values



JavaScript values are either literals or variables. Literals have fixed values which can't be changed and come in different data types.

What is a variable?



- In programming, variables are used to store values so it can be referenced.
e.g. Instead of writing out 3.14159265359 every time, you can set a variable named “pi” and reference it whenever you want.
- You already have an idea of what variables are if you learned algebra.
e.g. $x + 2 = 10$. x is a variable that has the value of 8.

Data types



The type of data is used to tell JavaScript how the data should be used. It is important when trying to operate on values.

e.g.

The number 10 could also be text, and JavaScript will perform differently depending on which it is.

Data types



If 10 is a number...

$$10 + 20 = 30$$

If 10 is text...

$$\text{"10"} + 20 = \text{"1020"}$$

Which is why you have to specify which one.

String



Strings store text data. Strings are made up of characters inside quotes. You can also have quotes inside of a string as long as they don't match the quotes surrounding it.

e.g. "Hello, World!", "ERB002", "He said 'Good morning' to me"

Number



Number data store numeric values. They can represent integers and decimals (floating point), as well as positive and negative numbers.

e.g. 12, -24, 16.7

Boolean



A Boolean can represent either True or False. Useful when comparing values and writing conditions.

undefined and null



- undefined represents a variable that has been declared by not assigned a value yet.
- null is a value you can assign that represents nothing.

typeof()



Use `typeof()` to find out the data type of a variable.

e.g.

```
var x = 6;  
typeof(x); // Number  
  
var name = "Steve"  
typeof(name); // String
```

Short quiz



What are the data types of the following?

- a) 200
- b) "School"
- c) False
- d) "12"

Declaring variables



Use the `var` keyword to declare a new variable. By default, the data type will be `undefined`.

e.g.

```
var x;
```

```
var name;
```

Note: JavaScript is case-sensitive. (`cat` ≠ `Cat` ≠ `CAT`)

Reserved words



There are certain words you can't use to name a variable, because they are already used by JavaScript for different functions.

e.g. var var = 10; is not allowed.

For a full list of reserved words, visit [here](#).

Naming variables



- Variable names should be meaningful, not just to yourself. Other people will read your code one day.
- For variable names with multiple words, use camelCase. First word starts with a lowercase letter, and start subsequent words with an uppercase letter. (e.g. firstName)

Assigning variables



- To assign a value to a variable, use the = assignment operator.

e.g. x = 5; will assign the value 5 to the variable x.

Assigning variables



- When declaring a new variable, you can also assign it a value at the same time.

e.g. var y = 20;

Assigning variables



To change the value in a variable, simple assign the new value over it.

e.g.

```
var z = 10;
```

```
z = 20;
```

z is now equal to 20 instead of 10.

Strict mode



Strict mode will prevent undeclared variables from being used.

Add “use strict”; to tell the browser that the code should be run in strict mode.

Strict mode



```
"use strict";  
  
x = 10; // ReferenceError: x is not defined
```

Strict mode



- Strict mode makes it easier to spot silent errors.
- For example, if you mistyped a variable name in normal mode, a new variable would be created.
- In strict mode, a mistyped variable will throw an error.

Practice: Variables



Spend some time practicing declaring and assigning variables. Try it with strict mode off and on.

Swapping variables



Imagine you had two boxes, box x and box y, and you wanted to swap their contents.

Swapping variables



A

B

Swapping variables

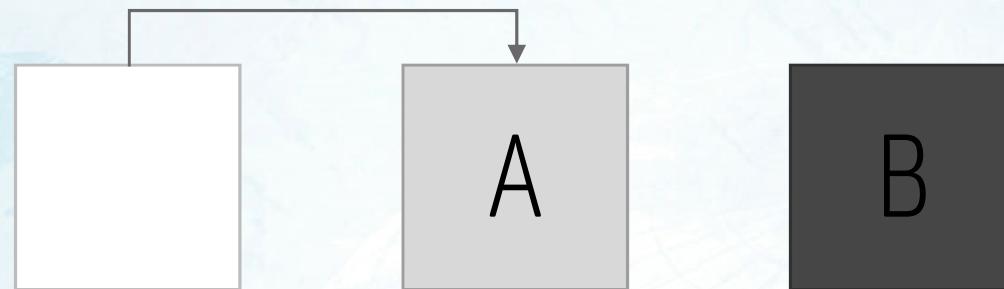


A

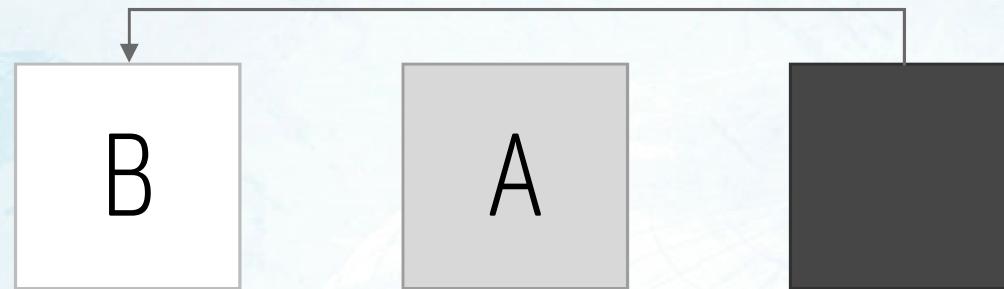


B

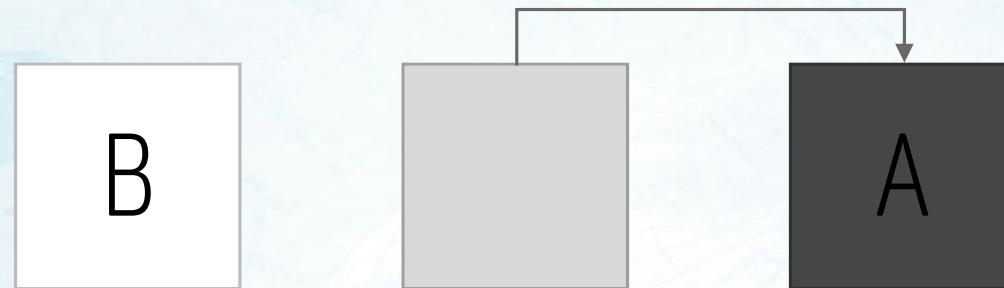
Swapping variables



Swapping variables



Swapping variables



Swapping variables



B

A

Swapping variables



```
var x = "A";
var y = "B";
var temp;
```

```
temp = x;
x = y;
y= temp;
```

```
x; // "B"
y; // "A"
```



2.3 Arithmetic Operators



Arithmetic operators



Arithmetic operators are used to perform calculations on numbers and variables.

e.g. $x = 10 + 20$, $y = 20 - 10$

Arithmetic operators



Operator	Function
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder
**	Exponentiation
++	Increment
--	Decrement

Arithmetic operator exercise



$$a^{**2} + b^{**2} = c^{**2}$$

What famous equation is this?

Arithmetic operator exercise



```
var x = 30;  
var y = 2;  
y++;  
x = x / y;  
x = x ** 2;
```

What does x equal?

More arithmetic syntax



You can use parentheses to indicate which arithmetic operators to execute first.

e.g.

$x = 1 + 2 * 3$, x equals 7

$x = 1 + (2 * 3)$, x equals 7

$x = (1 + 2) * 3$, x equals 9

More arithmetic syntax



Parentheses can be stacked on top of each other.

e.g.

$$x = ((1 + 2) * (3 + 4)) / 3, \text{ } x \text{ equals } 7$$

2.4 Comments



Comments



- Comments can be used to explain code to others, or as a reminder for yourself.
- It can also be used to prevent code from being executed when testing a program.

Comments



Single line comments:

```
// This is a comment
```

Multi line comments:

```
/*
```

```
This is a comment
```

```
This is also a comment
```

```
*/
```

Comment example



```
// Declare and assign the variable x  
var x = 10;
```

Comments



Get into the habit of leaving comments in your code, not just for JavaScript, but for HTML and CSS too, as well as any other languages you may learn in the future.

Practice: Adding comments to old code



Go back to some of your old files and add come comments to describe their functions. Then show your friends and see if they can understand it.



The End



Reference 1: W3 Schools JavaScript Tutorial <https://www.w3schools.com/js/default.asp>