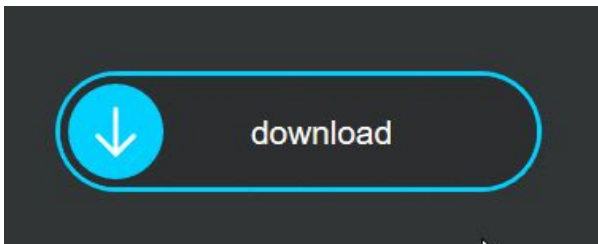


Web development basics

JS basics

Introduction to JavaScript

JavaScript is a programming language that allows you to add interactivity and dynamic behavior to websites. It can be used to create things like responsive navigation menus, image sliders, and form validation. It runs in the user's browser, allowing you to make web pages dynamic and interactive without having to refresh the page.



Syntax and language structure

```
import resolveAfter50ms from './resolve'

async function add(x) {
  if (x < 0) throw new Error('The number must be positive');
  var a = resolveAfter50ms(60);
  var b = resolveAfter50ms(30);
  return x + await a + await b;
}

add(10).then(result => {
  // res
})
```

JavaScript is a high-level, interpreted language. It uses a C-style syntax, which means that it uses curly braces, semi-colons, and similar structure. It is also an object-oriented language, which means that it is based on the concept of objects, which have properties and methods.

Variables in JavaScript

A JavaScript program typically starts with a declaration of variables, which are used to store data. Variables are declared using the keyword "var", "let" or "const" followed by the variable name and an assignment operator (=) to assign a value.

```
let name = "John";  
const age = 30;
```

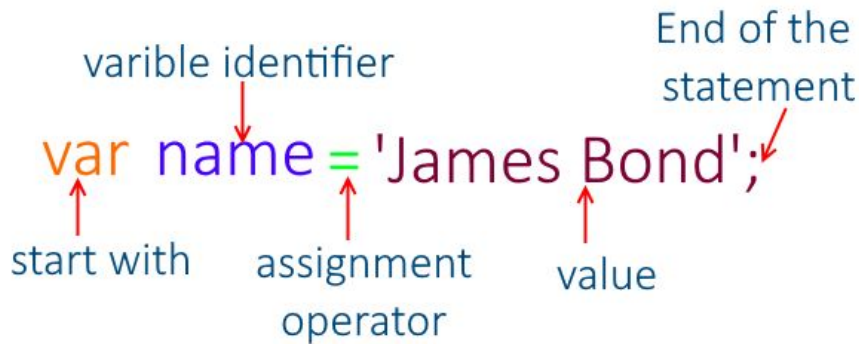


Diagram illustrating the syntax of a variable declaration statement: `var name = 'James Bond';`

- `var`: start with
- `name`: variable identifier
- `=`: assignment operator
- `'James Bond'`: value
- `;`: End of the statement


Variables in JavaScript 2

var

var apple = 



a thing in a box
named "apple"

apple = 



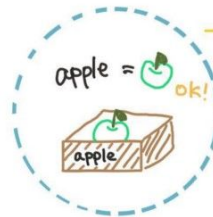
you can swap
item later

let

let apple = 



a thing in a box
named "apple" w/
protection shield



~~apple = ~~ _{NG}

ok! you can swap item
only if you ask
inside of the shield

const

const apple = 



a thing in
LOCKED cage
named "apple"



~~apple = ~~ _{NG}

you can't
swap item
later.

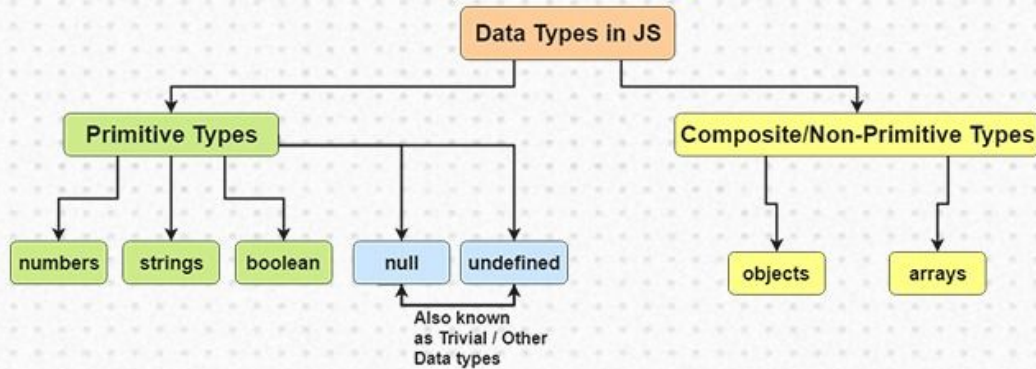


apple.multiply(3) _{ok!}


... but you can ask
the item to change itself
(if the item has method
to do that)

Datatypes

Variables & Datatypes **in JavaScript**



Type conversion



TYPE CONVERSION TABLE

JavaScript / CheatSheet

	false	true	0	1	NaN	Infinity	-Infinity	null	undefined	{}
Number	0	1	0	1	NaN	Infinity	-Infinity	0	NaN	NaN
String	"false"	"true"	"0"	"1"	"NaN"	"Infinity"	"-Infinity"	"null"	"undefined"	"[object Object]"
Boolean	false	true	false	true	false	true	true	false	false	true

	""	"0"	"1"	"Ten"	[]	[10]	[10,20]	["ten"]	["one","two"]
Number	0	0	1	NaN	0	10	NaN	NaN	NaN
String	""	"0"	"1"	"10"	""	"10"	"10,20"	"ten"	"one,two"
Boolean	false	true	true	true	true	true	true	true	true

For more JS related infographics, follow [@swapnaka](#)

Logical operands

Relational Operators

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True
===	Equal value and same type	5 === 5	True
		5 === "5"	False
!==	Not Equal value or Not same type	5 !== 5	False
		5 !== "5"	True

↳

Classwork - variables

Task: Declare a variable in JavaScript called "name" and assign it the value of your first name. Then, declare a variable called "age" and assign it your current age. Finally, use console.log to print out a sentence that says "My name is [name] and I am [age] years old."

Expected Output: "My name is John and I am 30 years old."

Classwork - variables (solution)

```
let name = "John";  
let age = 30;  
console.log("My name is " + name + " and I am " + age + " years old.");
```

Arrays in JavaScript

```
let array = [1, 12, 2.5, null, 'John', true, 100]
```

	int	int	float	Null	string	bool	number
Elements: →	1	12	2.5	null	'John'	true	100
Index : → (position)	0	1	2	3	4	5	6

Javascript Array

JavaScript **arrays** and **objects** are both used to store **collections of data**.

An array is a list of values that are organized **in a specific order**. Each value in an array is called an element, and each element has a numerical index that can be used to access it. For example, if you have an array called "fruits" that contains the elements "apple", "banana", and "orange", you could access the "banana" element by referring to fruits[1], since it is the second element in the array (**indexes start at 0**).

Objects in JavaScript

An **object** is a collection of **key-value** pairs, where each **key** is a string and each **value** can be any type of data. Objects are similar to arrays in that they can store **collections of data**, but the **data is organized differently**. Instead of using numerical indexes to access the data, you use the keys. For example, if you have an object called "person" that has properties like "name", "age", and "address", you could access the "name" **property** by referring to person.name

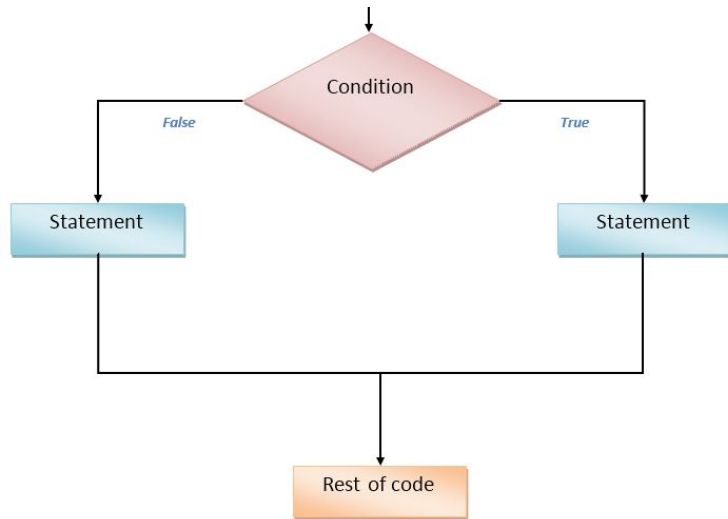
```
1  const list = [  
2    {  
3      name: 'Michael Scott',  
4      company: 'Dunder Mufflin',  
5      designation: 'Regional Manager',  
6      show: 'The Office'  
7    },  
8    {  
9      name: 'Barney Stinson',  
10     company: 'Golaith National Bank',  
11     designation: 'Please',  
12     show: 'How I met your mother'  
13   },  
14   {  
15     name: 'Jake Peralta',  
16     company: 'NYPD',  
17     designation: 'Detective',  
18     show: 'Brooklyn 99'  
19   },  
20 ]
```

Classwork - objects

Task: Create an object, that describes you or anything else you can imagine. Object should be nested and contain another object inside.

Conditional Statements (if-else)

A conditional statement in JavaScript is used to make decisions in code. It allows you to perform different actions based on different conditions. The most common type of conditional statement is the if-else statement.

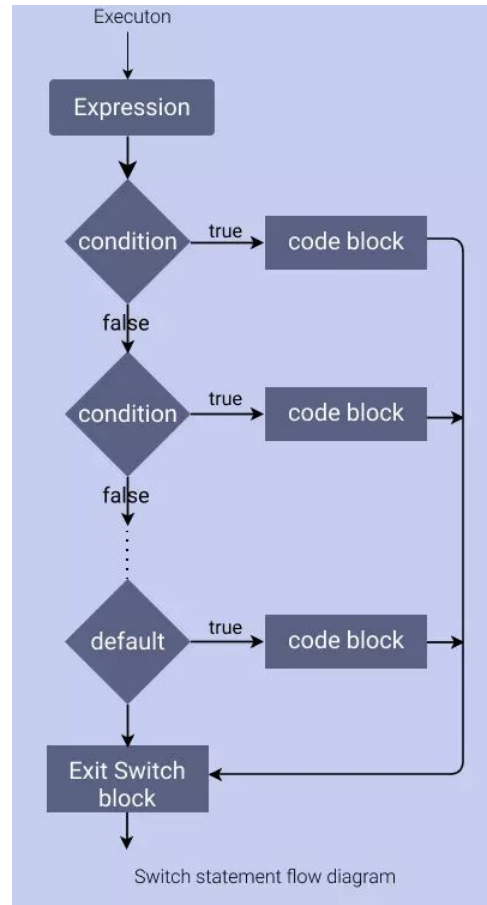


```
if (condition1) {  
    // code to be executed if condition1 is true  
} else if (condition2) {  
    // code to be executed if condition1 is false and condition2 is true  
} else {  
    // code to be executed if both condition1 and condition2 are false  
}
```

Conditional Statements (switch)

You can also use a switch statement, it's used when you want to test a value against multiple conditions:

```
switch (expression) {  
  case value1:  
    // code to be executed if expression === value1  
    break;  
  case value2:  
    // code to be executed if expression === value2  
    break;  
  default:  
    // code to be executed if expression is not equal to any of the  
    values  
}
```



Homework

Create condition statements (using both, if and switch), that based on the variable “day” prints out day of the week.

Example - if day = 5, console output = Friday