Variables & Constants

```
let userName = 'Max';
```

A “data container” / “data storage”

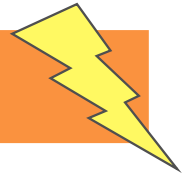
```
userName = 'Manu';
```

...where the value can change!

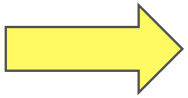
```
const totalUsers = 15;
```

A “data container” / “data storage”

```
totalUsers = 20;
```



...where the value must not change!



Use **constants as often as possible** (i.e. whenever you actually got data that never changes) to be clear about your intentions (in your code).

Operators

+	Add two numbers
-	Subtract two numbers
*	Multiply two numbers
/	Divide two numbers
%	Divide two numbers, yield remainder
**	Exponentiation (e.g. 2 ** 3 = 8)

=	Assign value to variable

Functions

"Code on Demand"

(1) Define Function

```
function greetUser(name) {
  alert('Hi ' + name);
}
```

You can (but don't have to) use ***parameters*** (e.g. *name*) and you can (but don't have to) return values (via ***return***).

(2) Call Function

```
greetUser('Max');
```

As often as you want, passing in (different) parameter values!

Every function execution then runs independent from (possible) other executions.

Operators

+	Add two numbers
-	Subtract two numbers
*	Multiply two numbers
/	Divide two numbers
%	Divide two numbers, yield remainder
**	Exponentiation (e.g. 2 ** 3 = 8)

=	Assign value to variable
+=, -=, ...	Perform calculation and re-assign result to variable
++, --	Increment / Decrement variable value + re-assign

ACADEMIND

Data Types

Numbers	2, -3, 22.956	Important for calculations and code where you need to “work with a number”
Strings (Text)	'Hi', "Hi", `Hi`	Important for outputting results, gathering input
Booleans	true / false	Important for conditional code and situations where you only have 2 options
Objects	{ name: 'Max', age: 31 }	Important for grouped/ related data, helps you with organizing data
Arrays	[1, 3, 5]	Important for list data, unknown amounts of data

null / undefined / NaN

Special Values

undefined	null	NaN (Not a type!)
Default value of uninitialized variables	Never assumed by default	Technically, it's of type number and can therefore be used in calculations
You shouldn't assign undefined as a value manually	You can assign this is a value if you want to "reset" / "clear" a variable	It yields a new NaN and it's the result of invalid calculations (e.g. 3 * 'hi')

NOT entirely equal!

Different Ways of Adding JavaScript to a Page

Use this approach!

Inline JavaScript Code	External / Imported JavaScript Files

`<script>`
`alert('Hi there!');`
`</script>`
	`<script src="someFile.js"></script>`
	`<script src="otherFile.js"></script>`
	`<script src="otherFile.js"></script>`
	Multiple imports (even of the same file) are possible and normal!

Executed directly when the browser (HTML parser) reaches this tag	Requested & loaded when browser (HTML parser) reaches this tag
Typically only used for very trivial sites/ scripts (you'd create a huge HTML file otherwise)	Execution behavior depends on configuration: **async** and **defer** allows HTML parser to continue

How To Import Your Scripts

Inline	<script> alert('Hi!'); </script>	Immediatley executed, blocks HTML parsing & rendering
External File	<script src="file.js"> </script>	Immediatley loaded & executed, blocks HTML parsing & rendering
External File (async)	<script src="file.js" async> </script>	Immediatley loaded & executed thereafter, blocks HTML parsing & rendering
External File (defer)	<script src="file.js" defer> </script>	Immediatley loaded & executed after HTML parsing & rendering

Timeline Summary