DOCENTE	Shadi Lahham
Corso	Web Developer
Unità Formativa	Programmazione - Javascript e Typescript
Argomento	Specificato nel titolo della slide successiva

Variables, types and operators

Foundation

Shadi Lahham - Web development

Variables

Statements

Each instruction in JS is a "statement", like:

```
console.log('Hello World!');

document.getElementById("demo").innerHTML = "Hello Dolly.";
```

More details:
JavaScript Statements

Variables

Use variables to store values

Declare, then initialize in 2 statements:

```
let x;
x = 5;
console.log(x);
```

Or declare and initialize in one statement:

```
let y = 2;
console.log(y);
```

Re-assign the value later:

```
let x = 5;
x = 1;
```

Primitive Data Types

```
string: an immutable string of characters:
let greeting = 'Hello Kitty';
let restaurant = "Paul's Place";

number: whole (6, -102) or floating point (5.8737):
let myAge = 28;
let pi = 3.14;

boolean: Represents logical values true or false:
let catsAreBest = true;
let dogsRule = false;

undefined: Represents a value that hasn't been defined.
let notDefinedYet;

null: Represents an explicitly empty value.
let goodPickupLines = null;
```

Strings

```
A string holds an ordered list of characters:
let alphabet = "abcdefghijklmnopqrstuvwxyz";

The length property reports the size of the string:
console.log(alphabet.length); // 26

Each character has an index.
The first character is always at index 0.
The last character is always at index length-1:
console.log(alphabet[0]); // 'a'
console.log(alphabet[1]); // 'b'
console.log(alphabet[2]); // 'c'
console.log(alphabet[alphabet.length]); // undefined
console.log(alphabet[alphabet.length-1]); // 'z'
console.log(alphabet[alphabet.length-2]); // 'y'
```

Variable Names

- Begin with letters, $ or _
- Only contain letters, numbers, $ and _
- Case sensitive
- Avoid reserved words
- Choose clarity and meaning
- Prefer **camelCase** for multiple words (instead of under_score)
- Pick a naming convention and stick with it

Camel case - MDN
Camel case - Wikipedia

note: $ is usually used by libraries such as jQuery so it’s best to avoid in variable names

Variable Names

OK:

```
let numPeople, $mainHeader, _num, _Num;
```

Not OK:

```
let 2coolForSchool, soHappy!
```

Meaningful Names in English

```
// always choose meaningful variable names; code is easier to understand and maintain
let x = 100; // bad
let totalPrice = 100; // good

// always use English for variable names, even for values in a different language
let liczbaProduktow = 5; // bad
let liczbaProduktow = 'pięć'; // bad
let nazwaUzytkownika = 'Jan Kowalski'; // bad
```

Meaningful Names in English

```
// always choose meaningful variable names; code is easier to understand and maintain
let x = 100; // bad
let totalPrice = 100; // good

// always use English for variable names, even for values in a different language
let numberOfProducts = 5; // good
let numberOfProducts = 'pięć'; // good
let userName = 'Jan Kowalski'; // good
```

Expressions

Variables can also store the result of any "expression":

```
let x = 2 + 2;
let y = x * 3;
let myName = 'Gina';
let greeting = 'Hello ' + myName;
let title = 'Baroness';
let formalGreeting = greeting + ', ' + title
```

Loose Typing

JS figures out the type based on value, and the type can change:

```
let x;
x = 2;
x = 'Hi';
```

A variable can only be of one type:

```
let y = 2 + ' cats';
console.log(typeof y);
```

Operators

Arithmetic Operators

```
let a = 12 + 5;    // 17
let b = 12 - 5;    // 7
let c = 12 * 5;    // 60
let d = 12 / 5;    // 2.4 - division results in floating point numbers.
let e = 12 % 5;    // 2 - the remainder of 12/5 in integer math is 2.

let a = "1";
let b = a;     // b = "1": a string
let c = +a;    // c = 1: a number
let d = -a;    // d = -1: a number
```

Assignment Operators

Assignment:

```
x = y
x += y
x -= y
x *= y
x /= y
x %= y
```

Same as:

```
x = y
x = x + y
x = x - y
x = x * y
x = x / y
x = x % y
```

note:
x has to be already declared

Increment Operators

```
let a = 1;
a = a + 1;
a += 1;
a++;
++a;

// increment occurs before a is assigned to b
let a = 1;
let b = ++a; // a = 2, b = 2;

// increment occurs to c after c is assigned to d
let c = 1;
let d = c++; // c = 2, d = 1;
```

Comparison Operators

```
==  Is equal to
=== Is identical (is equal to and is of the same type)
!=  Is not equal to
!== Is not identical
>   Greater than
>=  Greater than or equal to
<   Less than
<=  Less than or equal to
```

```
let x = 5;
x == 5;     //true
x === "5";  //false
```

Logical Operators

Operators:

```
&&  and
||  or
!   not
```

Examples:

```
(x < 10 && y > 1)
(x === 5 || y === 5)
!(x === y)
```

String Operators

`+`
`+=`

Examples:

```
text3 = text1 + text2;
text1 += text2;
```

Operator Classification

Operator Classification

operator: an entity (such as a symbol or keyword) that performs an action on operands
operand: an entity (such as a variable or value) on which an operator performs an action

```
// unary (1 operand)
let x = 5;
console.log(++x); // unary operator

// binary (2 operands)
let y = 5; // binary operator
let z = 3; // binary operator
console.log(y + z); // binary operator

// ternary (3 operands)
let a = 10;
let b = 5;
let result = a > b ? 'yes' : 'no'; // ternary operator
```

Operator Precedence

Operator Precedence

```
// multiplication has higher precedence than addition
let result = 10 + 5 * 2;
console.log(result); // Output: 20

// parentheses change the precedence, so addition is done first
let result1 = (10 + 5) * 2;
console.log(result1); // Output: 30
```

Operator Precedence

```
// addition has higher precedence than assignment
let x = 5;
x *= 2 + 3;
console.log(x); // Output: 25

// logical AND has lower precedence than comparison
let comparison = 10 > 5 && 5 <= 3;
console.log(comparison); // Output: false

// logical NOT has higher precedence than both logical AND and logical OR
let logical = true || false && !false;
console.log(logical); // Output: true
```

JavaScript Operator Precedence
Operator precedence | MDN

Let & const

Let

```
let x = 88;
console.log('value of x', x);

for (let i = 0; i < 10; i++) {
  let t = i;
  console.log('inside i = ', i);
  console.log('inside t = ', t);
}

console.log('outside i = ', i); // i not defined
console.log('outside t = ', t); // t not defined
```

let: Block-scoped

Access restricted to nearest enclosing block

Const

```
let x = 88;
const y = 77;
x = 9;
console.log('x = ', x);
y = 17;  // TypeError: Assignment to constant variable.
console.log('y = ', y);
const y = 55; // SyntaxError: Identifier 'y' has already been declared
```

const: Block-scoped, like **let**

Values of const variables cannot be reassignment
Const variables cannot be redeclared

Your turn

1.Tell my fortune

- Store the following into variables: number of children, partner's name, geographic location, job title.
- Output your fortune to the console like so: "You will be a X in Y, and married to Z with N kids."

note: remember to create an **index.html** file and a **main.js** file
Do this for all future exercises
Open your browser's devtools and go to the console

Open Chrome DevTools
Open Firefox DevTools

2.Calculate age

- Store your birth year in a variable.
- Store a future year in a variable.
- Calculate your 2 possible ages for that year based on the stored values.
- For example, if you were born in 1988, then in 2026 you'll be either 37 or 38, depending on what month it is in 2026.

- Output them to the console like so: "I will be either NN or NN in YYYY", substituting the values.

3.Free coffee

- Store your current age into a variable.
- Store a maximum age into a variable.
- Store the amount of coffee you drink per day (as a number).
- Calculate how much coffee you would drink for the rest of your life.

- Output the result to the console like so: "You will need NN cups of coffee to last you until the ripe old age of X".

Bonus

4.Easy geometry

Calculate properties of a circle, using the definitions here.

- Store a radius into a variable.
- Calculate the circumference based on the radius, and output "The circumference is NN".
- Calculate the area based on the radius, and output "The area is NN".

Reference:

JavaScript Math Object

Circles

5.Convert temperature

- Store a celsius temperature into a variable.
- Convert it to fahrenheit and output "NN°C is NN°F".
- Now store a fahrenheit temperature into a variable.
- Convert it to celsius and output "NN°F is NN°C."

References

JavaScript Operators Reference

Expressions and operators

JavaScript data types and data structures

Values, Types, and Operators

Operator Precedence

JavaScript Operator Precedence
Operator precedence | MDN

Javascript validation

Code quality tools

ESLint

JSHint

© Copyright & Attribution