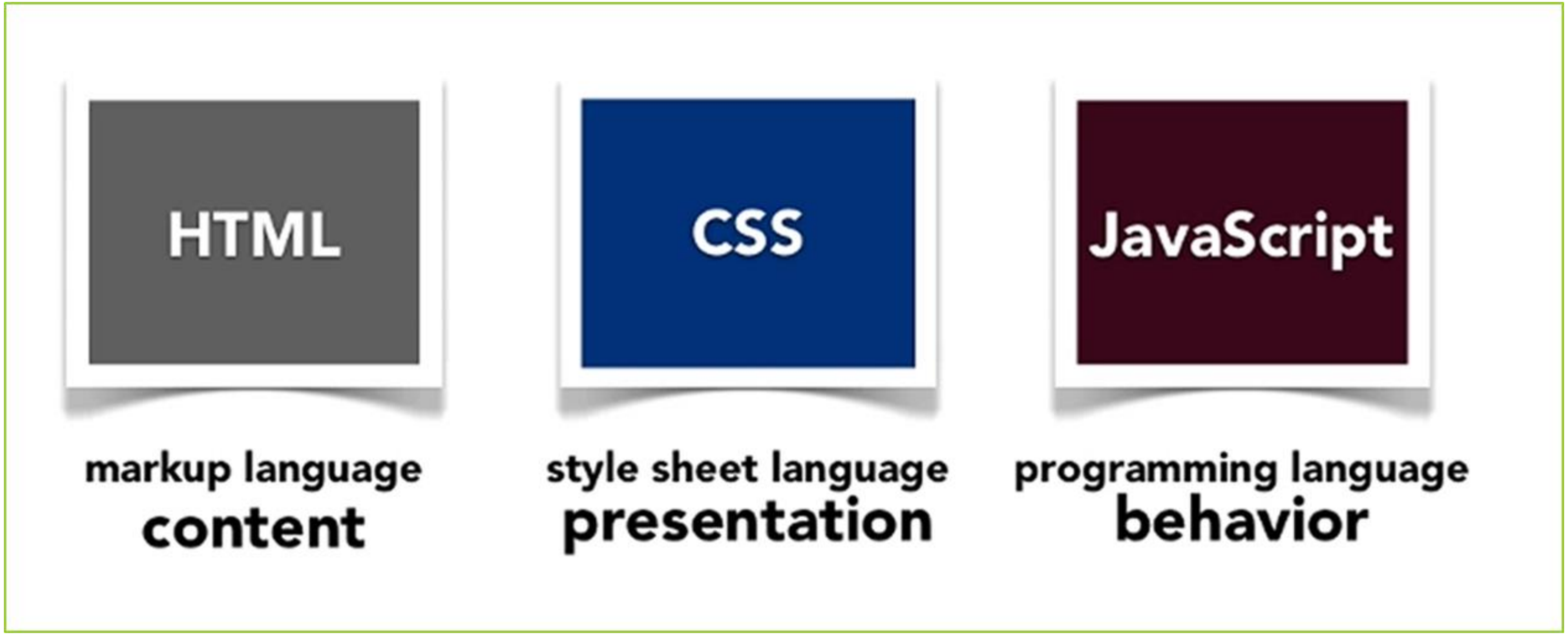
JavaScript (EN)

Justina Balsė

HTML+CSS+JavaScript



HTML

markup language
content

CSS

style sheet language
presentation

JavaScript

programming language
behavior

JavaScript ir Java

- **JavaScript** **nėra** ir neturi **nieko** bendro su Java kalba.
- **JavaScript** tai tik pavadinimas, kuris buvo suteiktas, norint išpopuliarinti skriptų kalbą.

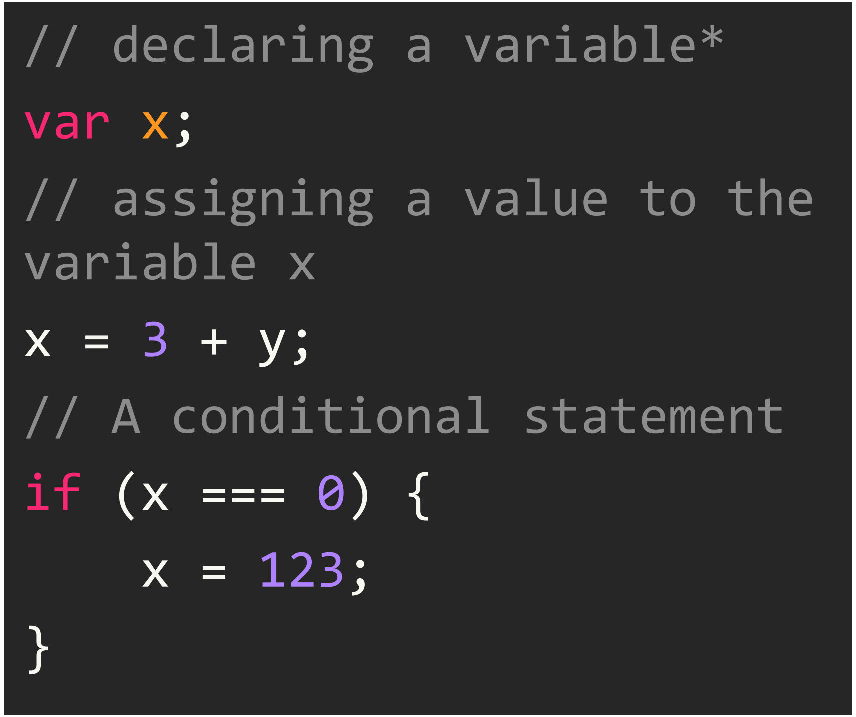
ES6, ES8, ES 2017, ECMAScript… ?

- **ECMAScript** is a standard. ECMA is an organization that standardizes information.
- **ES** is simply short for **ECMAScript**.
  - ***ES1****: June 1997 ,* ***ES2****: June 1998,* ***ES3****: Dec. 1999 ,* ***ES4****: Abandoned,* ***ES5:*** *December 2009,* ***ES6/ES2015:*** *June 2015, ...*

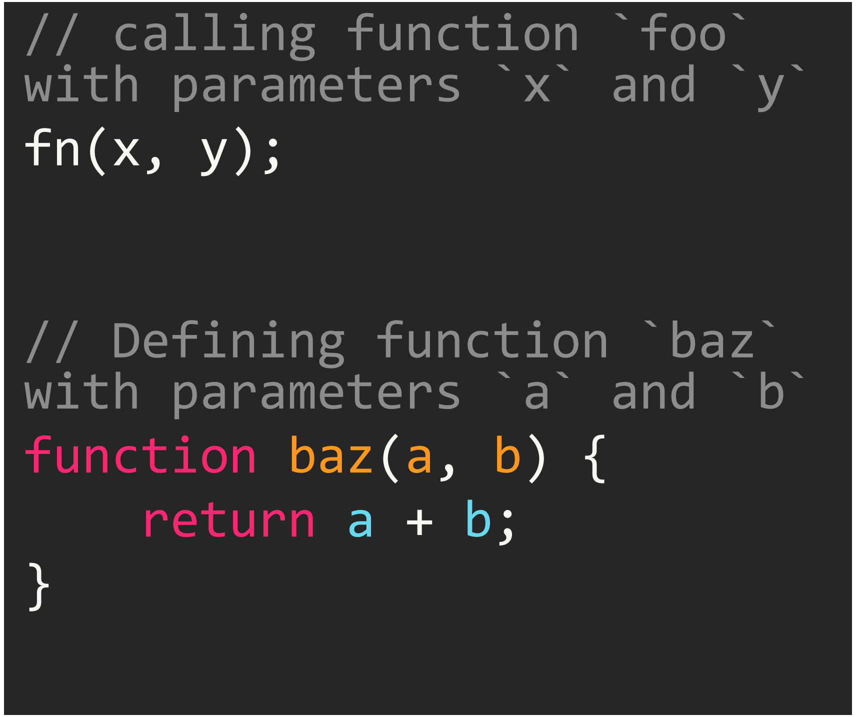
JavaScript

- Add new HTML to the page, change the existing content, modify styles.
- React to user actions, run on mouse clicks, pointer movements, key presses.
- Get and set cookies, ask questions to the visitor, show messages.
- Remember the data on the client-side (“local storage”).

JS Sintaksė



```
// declaring a variable*
var x;
// assigning a value to the variable x
x = 3 + y;
// A conditional statement
if (x === 0) {
    x = 123;
}
```



```
// calling function `foo` with parameters `x` and `y`
fn(x, y);


// Defining function `baz` with parameters `a` and `b`
function baz(a, b) {
    return a + b;
}
```

Declaring variables

- `var` - this syntax can be used to declare both local and global variables.
- `let` - this syntax can be used to declare a block-scope local variable.
- `const` - is used to assign a constant value to the variable.

Declaration and initialization

```
var name;
var age;
```

```
var name = "Tom" ;
var age = "39";
var cars = ["Saab",
"Volvo", "BMW"];
```

Data Types | 1

- **Primitive data types**
  - string, number, boolean, null and undefined
- **Objects**
  - Object refers to a data structure containing data and instructions for working with the data.

Data Types | 2

Strings:

  - single or double quotes;

- String Operators
  - Strings are concatenated via the plus (+) operator
- String Methods
  - length, toUpperCase(), indexOf('b') , charAt(1)...

Data Types | 3

Booleans: true, false

- Binary logical operators: && (And), || (Or)
- Prefix logical operator: ! (Not)
- Comparison operators:
  - Equality operators: ===, !==, ==, !=
  - Ordering operators (for strings and numbers):
    >, >=, <, <=

Data Types | 4

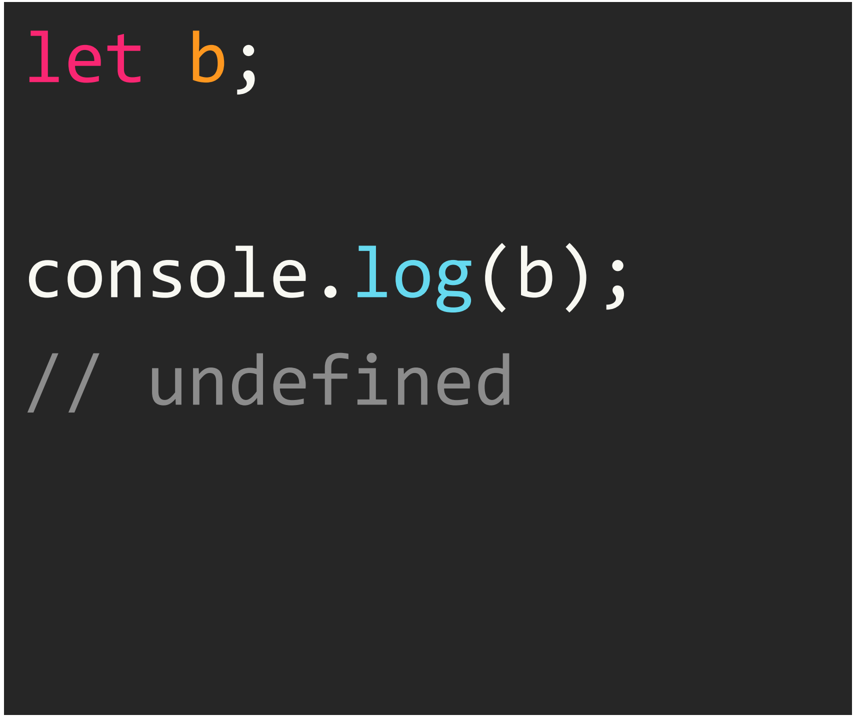
Numbers:

- All numbers in JavaScript are floating-point;
- NaN (“not a number”);
- Infinity.

Data Types | 5

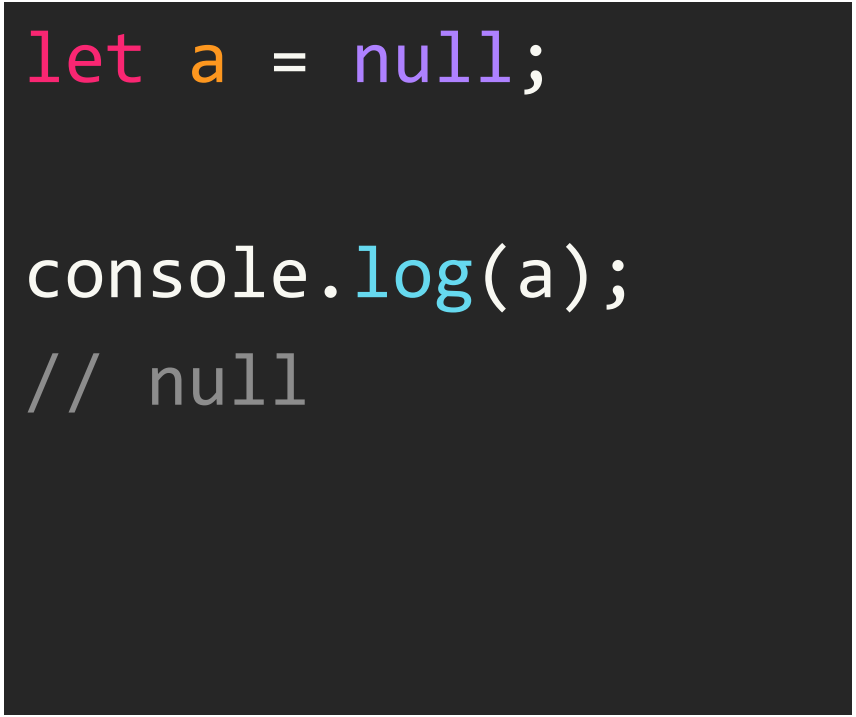
- **undefined and null**
  - **undefined** most typically means a variable has been declared, but not defined.
  - **null** - is an empty or non-existent value. JavaScript never sets a value to null. That must be done programmatically.

undefined and null



```
let b;

console.log(b);
// undefined
```

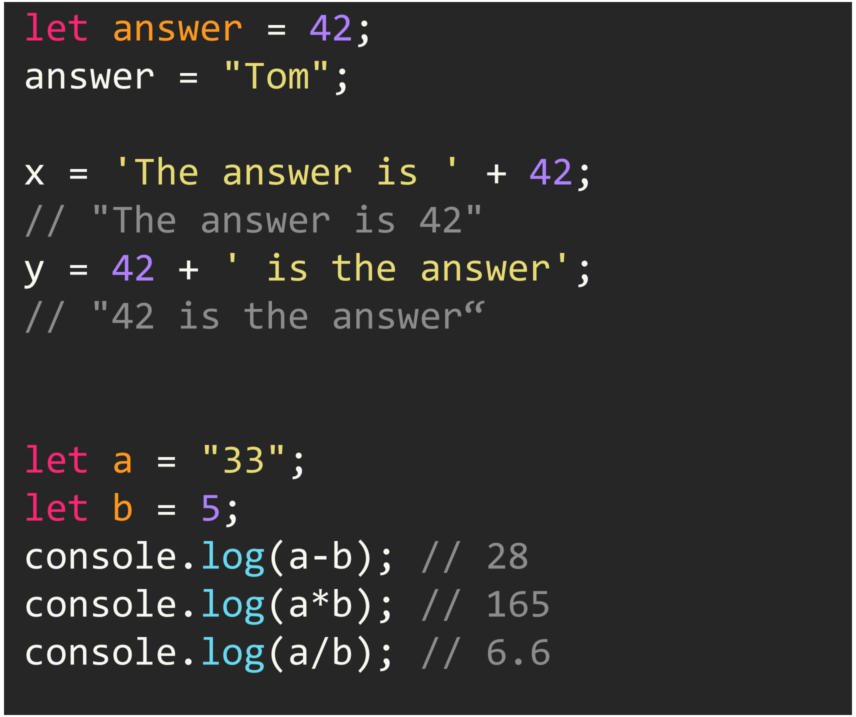


```
let a = null;

console.log(a);
// null
```

Data type conversion

JavaScript is a dynamically typed language. That means you don't have to specify the data type of a variable when you declare it, and data types are converted automatically as needed during script execution.

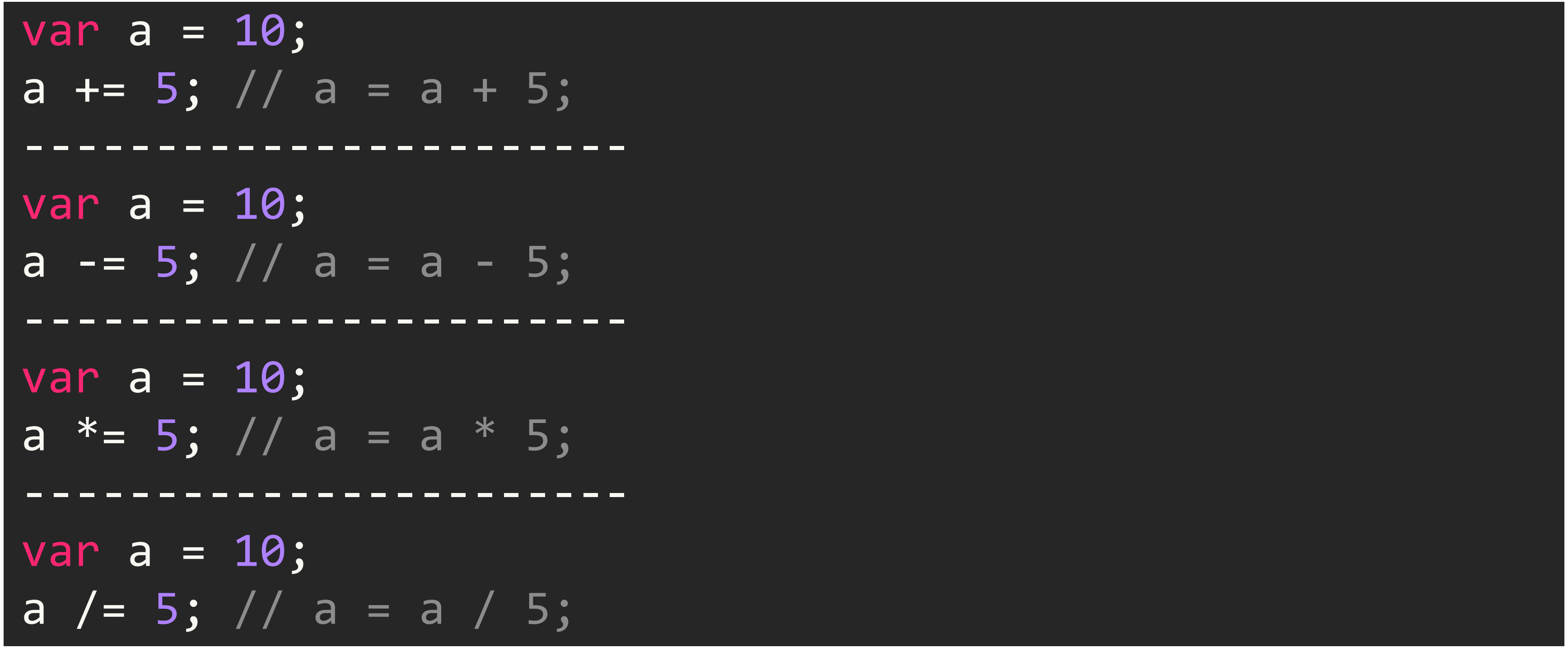


```
let answer = 42;
answer = "Tom";

x = 'The answer is ' + 42;
// "The answer is 42"
y = 42 + ' is the answer';
// "42 is the answer“


let a = "33";
let b = 5;
console.log(a-b); // 28
console.log(a*b); // 165
console.log(a/b); // 6.6
```

Operatoriai

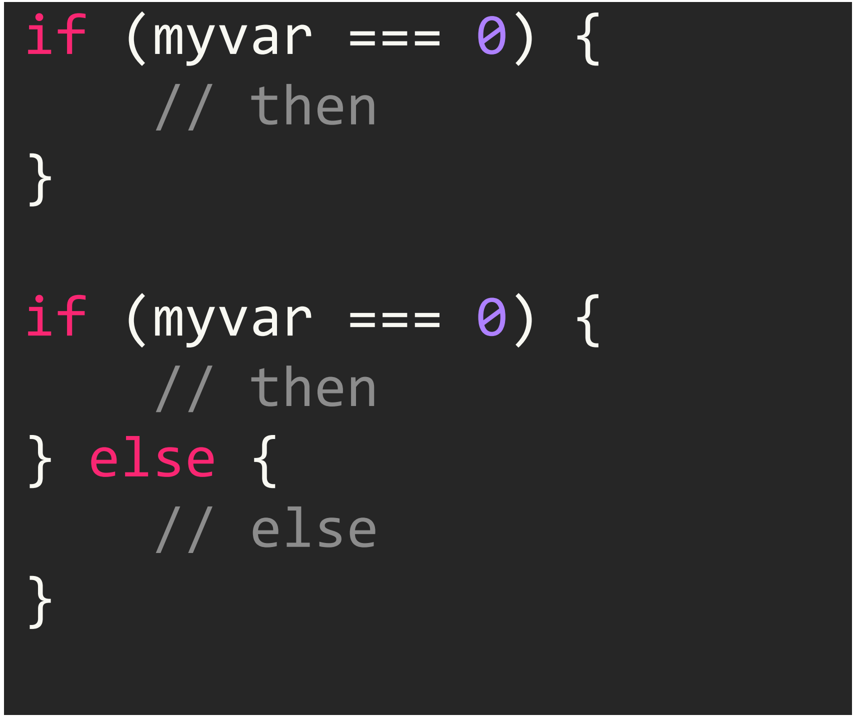


```
var a = 10;
a += 5; // a = a + 5;
-----------------------
var a = 10;
a -= 5; // a = a - 5;
-----------------------
var a = 10;
a *= 5; // a = a * 5;
-----------------------
var a = 10;
a /= 5; // a = a / 5;
```

The Math object

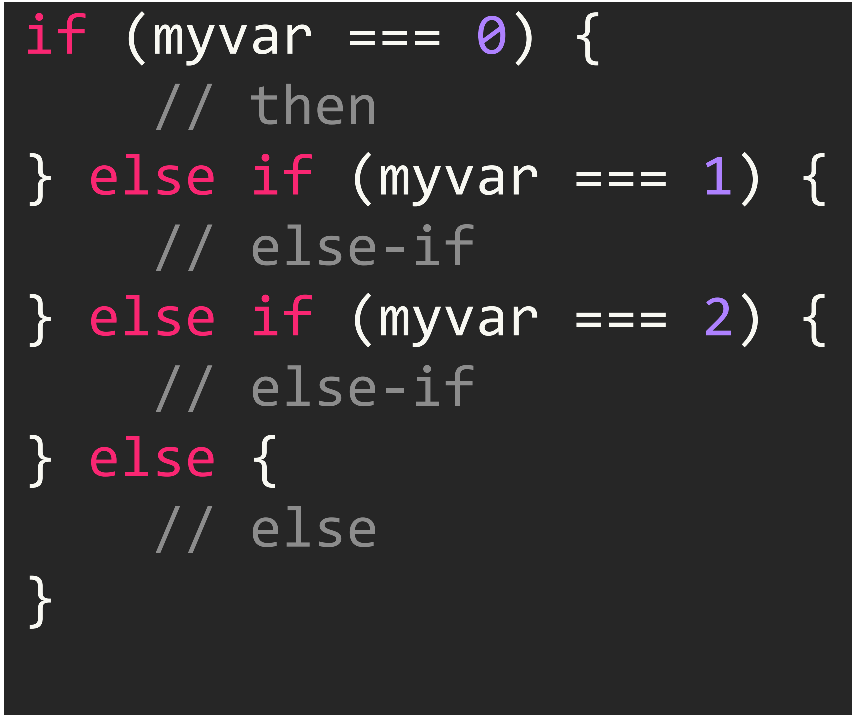
- Math.abs(x)
  - Returns the absolute value of x.
- Math.pow(x, y)
  - Returns x raised to the power of y.
- Math.sqrt(x)
  - Returns the square root of x.
- Math.random()
  - Returns a pseudorandom number $0 \leq r < 1$
  - `Math.floor((Math.random() * 10) + 1);`

Statements. Conditionals



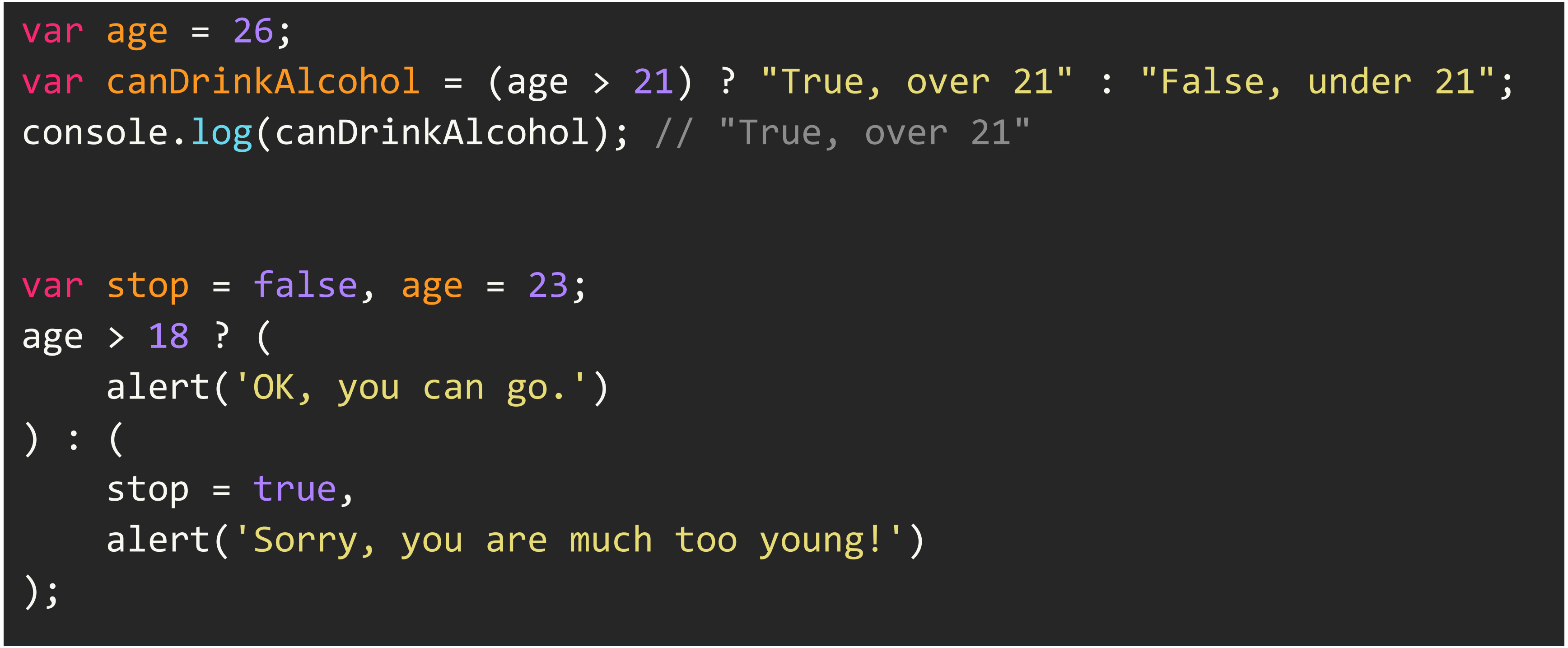
```
if (myvar === 0) {
    // then
}

if (myvar === 0) {
    // then
} else {
    // else
}
```



```
if (myvar === 0) {
    // then
} else if (myvar === 1) {
    // else-if
} else if (myvar === 2) {
    // else-if
} else {
    // else
}
```

The conditional (ternary) operator



```
var age = 26;
var canDrinkAlcohol = (age > 21) ? "True, over 21" : "False, under 21";
console.log(canDrinkAlcohol); // "True, over 21"


var stop = false, age = 23;
age > 18 ? (
    alert('OK, you can go.')
) : (
    stop = true,
    alert('Sorry, you are much too young!')
);
```

Statements. Conditionals

```
switch (fruit) {
    case 'banana':
        // ...
        break;
    case 'apple':
        // ...
        break;
    default:  // all other cases
        // ...
}
```

Statments. Loops

```
for (var i=0; i < arr.length; i++) {
    console.log(arr[i]);
}


// Same as for loop above:
var i = 0;
while (i < arr.length) {
    console.log(arr[i]);
    i++;
}
```

```
do {
    // ...
} while (condition);
```

In all loops:

- `break` leaves the loop.
- `continue` starts a new loop iteration.

Praktika

- Uždavinių sprendimas konsolėje.