



Εισαγωγή στη γλώσσα Javascript

Τμήμα Πληροφορικής Εισαγωγή στην
Επιστήμη Υπολογιστών



Εισαγωγή στη γλώσσα Javascript



Εισαγωγή στη γλώσσα Javascript

1^ο μέρος: εισαγωγικές έννοιες

Γιατί να μάθουμε Javascript;

- HTML to define the content of web pages
- CSS to specify the layout of web pages
- JavaScript to program the behavior of web pages
 - Δίνει δυνατότητα αλλαγής του HTML περιεχομένου
`document.getElementById("demo").innerHTML = "Hello JavaScript";`
 - Δίνει δυνατότητα αλλαγής των τιμών των HTML attributes
`<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">`
Turn on the light`</button>`
``
`<button`
`onclick="document.getElementById('myImage').src='pic_bulboff.gif'">`Turn off
the light`</button>`

Javascript και ο ρόλος της στην ανάπτυξη web εφαρμογών

- Η JavaScript είναι μια απλή στη σύνταξη, δυναμική, διερμηνευόμενη γλώσσα η οποία:
 - υποστηρίζει διάφορα υποδείγματα προγραμματισμού, πχ. διαδικασιακό, αντικειμενοστρεφή, συναρτησιακό, event-driven,
 - διαθέτει υποστήριξη για αντικείμενα, κλάσεις καθώς και για χρήση συναρτήσεων ως τιμών σε μεταβλητές (first class functions)
- Η JavaScript είναι ευρέως γνωστή για τη χρήση της στα πλαίσια ιστοσελίδων
 - εκτός από την Web Browser πλατφόρμα, η εκτέλεσή της υποστηρίζεται και σε επίπεδο Διακομιστή (πχ. Node.js, Apache CouchDB)

Προετοιμασία περιβάλλοντος ανάπτυξης

- Δεν χρειάζεται να κατεβάσουμε τη Javascript, τρέχει ήδη στους browsers της συσκευής μας
- Text editor (Visual Studio Code)
- Browser
- Online JavaScript Playground: <https://runjs.app/play>

Javascript & HTML

- Μέσα στο HTML αρχείο
 - tag `<script>` javascript code `</script>`
 - μέσα στο `<head>` ή `<body>` section
- Χωριστό αρχείο με Javascript κώδικα (`.js`), έστω `myCode.js`, για να το συμπεριλάβω σε μία σελίδα HTML:
 - `<script src="myCode.js"></script>` μέσα στο `<head>` ή `<body>` section
 - `<script src="/scripts/myCode.js"></script>`
 - `<script src="https://www.someplace.gr/js/oneScript.js"></script>`

Πλεονεκτήματα χωριστού αρχείου:

- Διαχωρισμός HTML περιεχομένου και κώδικα javascript
- Πιο εύκολη η συντήρηση περιεχομένου και κώδικα
- Cached αρχεία javascript → πιο γρήγορη η φόρτωση των web σελίδων

Έξοδος (output) κώδικα

- **Τροποποίηση** ενός HTML element

```
<p id="demo"></p>
```

```
<script> document.getElementById("demo").innerHTML = 5 + 6; </script>
```

- Εμφάνιση της εξόδου **μέσα στην HTML σελίδα**

```
<script>
```

```
    document.write(5 + 6);
```

```
</script>
```

- Εμφάνιση σε ένα **pop-up alert box**

```
<script>
```

```
    window.alert(5 + 6);
```

```
</script>
```

Βασικά θέματα σύνταξης Javascript κώδικα

- **Αναγνωριστικά** (identifiers) για μεταβλητές, keywords και συναρτήσεις
 - Ξεκινούν με λατινικό γράμμα (A-Z ή a-z), (ή με \$ _)
 - Περιλαμβάνουν γράμματα, αριθμούς, \$, _
 - Όχι δεσμευμένες λέξεις της γλώσσας
- Είναι **case sensitive**, π.χ. FirstName διαφορετικό από firstname
- Αγνοεί τα πολλαπλά κενά
- Διαχωρισμός εντολών (statements) με ; (περισσότερες εντολές στην ίδια γραμμή με ;)
- **Σχόλια** (comments): //single line /* multiple lines */
- Block εντολών { ... }

Μεταβλητές (variables)

- Χρησιμοποιούνται για την αποθήκευση δεδομένων.

Δήλωση μεταβλητών:

- αυτόματα με την ανάθεση τιμής

```
x = 5;  
y = 6;  
z = x + y;
```

- **var** ...

```
var x = 5;  
var y = 6;  
var z = x + y;
```

- **let** ...

```
let x = 5;  
let y = 6;  
let z = x + y;
```

Ανάθεση τιμής με =

- **const** ... (τιμή σταθερή που δεν μεταβάλλεται)

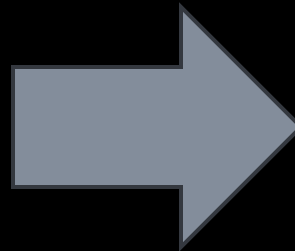
```
const x = 5;  
const y = 6;  
let z = x + y;
```

```
let person = "John Doe", carName  
= "Volvo", price = 200;
```

Μεταβλητές (2)

- Dynamically Typed: οι μεταβλητές JavaScript δεν δεσμεύονται με συγκεκριμένους τύπους δεδομένων. Ο τύπος δεδομένων ελέγχεται και χρησιμοποιείται κατά την εκτέλεση (run time) βάση της τιμής.

```
let x = 42;  
console.log(x)  
x = "hello";  
console.log(x)  
x = [1, 2, 3]  
console.log(x)
```



Output:

- 42
- hello
- [1, 2, 3]

Μεταβλητές (variables) | εμβέλεια (scope)

- **var** μεταβλητές

- εμβέλεια μέσα στη συνάρτηση που δηλώνονται ή καθολική (global) αν δηλωθούν εκτός συνάρτησης
- Μπορούν να δηλωθούν οπουδήποτε στην περιοχή εμβέλειάς τους

```
const x = 5;
```

```
const y = 6;
```

```
z = x + y;
```

```
var z;
```

- **const** και **let** μεταβλητές

- εμβέλεια επιπέδου block
- η δήλωσή τους πρέπει να γίνει πριν χρησιμοποιηθούν

```
let x = 5;
```

```
{
```

```
    let x = 1;
```

```
}
```

```
console.log(x);
```

```
// it will print 5
```

- Οι **const** μεταβλητές πρέπει να αρχικοποιηθούν κατά τη δήλωσή τους (δεν αλλάζει η τιμή)

Τύποι δεδομένων (data types)

- String: αλφαριθμητικοί χαρακτήρες μέσα σε ‘ ‘ ή “ “
- Number: αριθμοί ακέραιοι ή κινητής υποδιαστολής
- BigInt: 64-bit αριθμοί
- Boolean: true, false
- Object: σύνθετα δεδομένα, περιλαμβάνουν ιδιότητες (ζεύγη όνομα-τιμή)
 - Built-in: object, array, date, map, set, ...
 - User-defined

Τύποι δεδομένων | παραδείγματα

- // Numbers:
 - let length = 16;
 - let weight = 7.5;
- // Strings:
 - let color = "Yellow";
 - let lastName = "Johnson";
- // Booleans
 - let x = true;
 - let y = false;
- // Object:
 - const person = {firstName:"John", lastName:"Doe"};
- // Array object:
 - const cars = ["Saab", "Volvo", "BMW"];
- // Date object:
 - const date = new Date("2024-11-25");

Τύποι δεδομένων (data types)

- Οι τύποι δεδομένων είναι δυναμικοί, ο τύπος μεταβάλλεται ανάλογα με την ανάθεση τιμής

```
let x;    // Now x is undefined
```

```
x = 5;    // Now x is a Number
```

```
x = "John"; // Now x is a String
```

- Μπορεί να χρησιμοποιηθούν quotes μέσα σε string

```
// Single quote inside double quotes:
```

```
let answer1 = "It's alright";
```

```
// Single quotes inside double quotes:
```

```
let answer2 = "He is called 'Johnny'";
```

- Εκθετική σήμανση: let y = 123e5; // 12300000

```
let z = 123e-5; // 0.00123
```

Boolean

- `let x = 5;`
- `let y = 5;`
- `let z = 6;`
- `(x == y)` `// Returns true`
- `(x == z)` `// Returns false`

Πίνακες (Arrays)

- η δήλωση των στοιχείων ενός JavaScript array γράφονται μέσα σε `[ ]`, διαχωριζόμενα με `,`
- στο ακόλουθο παράδειγμα δηλώνεται ένα array (μία λίστα) car που περιέχει 3 στοιχεία (μάρκες αυτοκινήτων):
`const cars = ["Saab", "Volvo", "BMW"];`

Αντικείμενα (objects)

- η δήλωση των ιδιοτήτων (properties) ενός JavaScript object γράφονται μέσα σε `{ }` ως ζεύγη `name:value`, διαχωριζόμενα με `,`
- `const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};`

Τελεστές

- Μαθηματικές και λογικές πράξεις

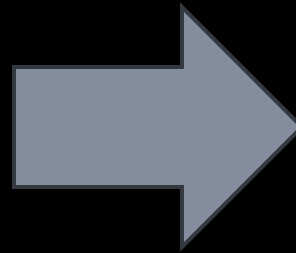
Τελεστής	Περιγραφή
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
**	Ύψωση σε δύναμη
/	Διαίρεση
%	Modulus (Υπόλοιπο)
++	Αύξηση
--	Μείωση

Τελεστής	Περιγραφή
==	ίσο
===	ίση τιμή & ίσος τύπος
!=	όχι ίσο
!==	όχι ίση τιμή & ίσος τύπος
>	μεγαλύτερο
<	μικρότερο
>=	μεγαλύτερο-ίσο
<=	μικρότερο-ίσο
?	ternary (υπό συνθήκη τελεστής)

Τελεστές

- Πρόσθεση strings & numbers

```
let x = 5 + 5; //πρόσθεση  
let y = "5" + 5; //συνένωση  
let z = "Hello" + 5; //συνένωση  
console.log(x, y, z);
```



Output:

- 10 '55' 'Hello5'

Τελεστές λογικών πράξεων

Operator	Description
&&	logical and
	logical or
!	logical not

Λογικοί τελεστές για 32-bit αριθμούς

Operator	Description	Example	Same as	Result	Decimal
&	AND	5 & 1	0101 & 0001	0001	1
	OR	5 1	0101 0001	0101	5
~	NOT	~ 5	~0101 1010	10	
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	left shift	5 << 1	0101 << 1	1010	10
>>	right shift	5 >> 1	0101 >> 1	0010	2
>>>	unsigned right shift	5 >>> 1	0101 >>> 1	0010	2

Τελεστής `typeof`, `instanceof`

- Τελεστής `typeof`: επιστρέφει τον τύπο δεδομένων

`typeof ""` // επιστρέφει "string"

`typeof "John"` // επιστρέφει "string"

`typeof "John Doe"` // επιστρέφει "string«

`typeof 3.14` // επιστρέφει "number"

`typeof (3)` // επιστρέφει "number"

`typeof (3 + 4)` // επιστρέφει "number«

- Τελεστής `instanceof`: επιστρέφει `true` όταν ένα `object` είναι instance κάποιου `object type`

Keywords (δεσμευμένες λέξεις)

Keyword	Description
var	Declares a variable
let	Declares a block variable
const	Declares a block constant
if	Marks a block of statements to be executed on a condition
switch	Marks a block of statements to be executed in different cases
for	Marks a block of statements to be executed in a loop
function	Declares a function
return	Exits a function
try	Implements error handling to a block of statements