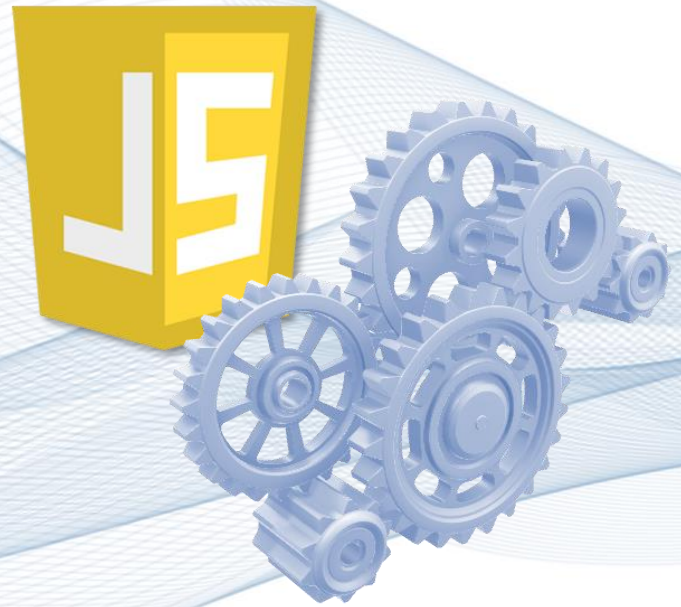
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ

Διαλέξεις μαθήματος

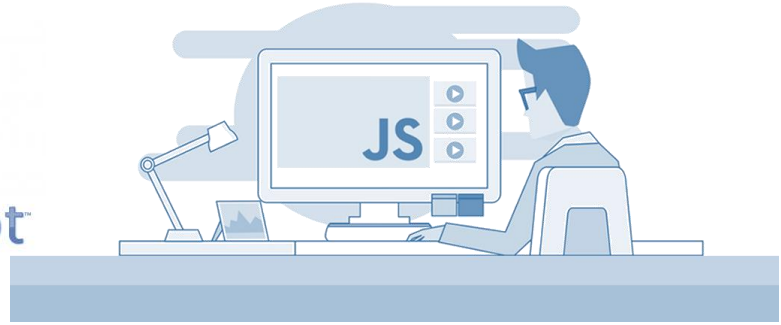
Μέρος 3ο
Μία Εισαγωγή στη Γλώσσα JavaScript



Αριστογιάννης Γαρμπής,
Καθηγητής

- Τι είναι η γλώσσα προγραμματισμού JavaScript;
- Σύνταξη: μεταβλητές, σχόλια, τύποι δεδομένων, τελεστές.
- Δομές επανάληψης.
- Συναρτήσεις: πως και πότε εκτελείται μια συνάρτηση.

Τι είναι η JavaScript;



- Είναι η πιο δημοφιλής scripting γλώσσα προγραμματισμού στον Παγκόσμιο Ιστό.
- Είναι μια interpreted γλώσσα προγραμματισμού, δηλαδή ο κώδικας JavaScript μπορεί να εκτελεστεί απευθείας από ένα φυλλομετρητή (browser), χωρίς να απαιτείται κάποιο προηγούμενο compilation.
- Έχει σχεδιαστεί και χρησιμοποιείται για να εισάγουμε την έννοια της διαδραστικότητας στις HTML σελίδες.
- Δεν έχει καμία σχέση με τη γλώσσα προγραμματισμού Java.

Γιατί χρειάζεται η JavaScript στον Παγκόσμιο Ιστό; (1/2)

- Η JavaScript χρησιμοποιείται για να εισάγουμε δυναμικά και αλληλεπιδραστικά χαρακτηριστικά στις HTML σελίδες.

- Συγκεκριμένα:
  - Η HTML καθορίζει το περιεχόμενο των HTML σελίδων.
  - Το CSS καθορίζει τη διάταξη (layout) των HTML σελίδων.
  - Η JavaScript καθορίζει τη συμπεριφορά (behavior) των HTML σελίδων.

Γιατί χρειάζεται η JavaScript στον Παγκόσμιο Ιστό; (2/2)

- Ενσωματώνοντας τη JavaScript σε μια HTML σελίδα μπορούμε να:
  - Μεταβάλλουμε δυναμικά το περιεχόμενο, τα χαρακτηριστικά (attributes) και το στυλ (style) των HTML στοιχείων (elements).
  - Εκτελέσουμε κάποιες ενέργειες όταν συμβαίνει ένα γεγονός (event).
    - π.χ. όταν ο χρήστης κάνει κλικ σε ένα HTML στοιχείο, μπορούμε να ορίσουμε να εκτελείται ένα συγκεκριμένο script και να παράγονται τα επιθυμητά αποτελέσματα.
  - Επικυρώσουμε (validate) τα δεδομένα που εισάγει ένας χρήστης σε μια φόρμα.
  - Μπορούμε να εντοπίσουμε το φυλλομετρητή (browser) του επισκέπτη μιας σελίδας και στη συνέχεια να φορτώσουμε την αντίστοιχη έκδοση της σελίδας που είναι φτιαγμένη για το συγκεκριμένο φυλλομετρητή.
  - Μπορούμε να δημιουργήσουμε και να διαβάσουμε cookies (να αποθηκεύουμε πληροφορίες) στον υπολογιστή του επισκέπτη ενός Διαδικτυακού Τόπου.

Σύνταξη (1/2)

- Για να εισάγουμε κώδικα JavaScript σε μια HTML σελίδα χρησιμοποιούμε την ετικέτα (tag) <script>.

```
<script>
    …
</script>
```

Οι εντολές που περικλείονται μέσα στο script, εκτελούνται από το φυλλομετρητή Παγκόσμιου Ιστού.

- Παλιότερα χρησιμοποιούνταν το χαρακτηριστικό (attribute) *type* για να οριστεί η scripting γλώσσα που πρόκειται να χρησιμοποιηθεί.
  - Σήμερα το χαρακτηριστικό αυτό είναι προαιρετικό καθώς η JavaScript είναι η default scripting γλώσσα της HTML.

```
<script type="text/javascript">
    …
</script>
```

Σύνταξη (2/2)

- Οι εντολές σε ένα πρόγραμμα JavaScript καλούνται δηλώσεις (statements).

- Κάθε εντολή τελειώνει με το σύμβολο ' ; '.

- Ο κώδικας JavaScript εκτελείται σειριακά, όπως και σε όλες τις γλώσσες προγραμματισμού.

Τρόποι εισαγωγής JavaScript κώδικα σε ένα HTML αρχείο

- Υπάρχουν τρεις τρόποι για να εισάγουμε κώδικα JavaScript σε ένα HTML αρχείο.
  - Να τοποθετήσουμε τον κώδικα JavaScript μέσα στην ετικέτα <head> μιας HTML σελίδας.
  - Να τοποθετήσουμε τον κώδικα JavaScript μέσα στην ετικέτα <body> μιας HTML σελίδας.
  - Να τοποθετήσουμε τον κώδικα JavaScript σε εξωτερικό αρχείο JavaScript.
    - Τα αρχεία JavaScript έχουν κατάληξη ".js".

JavaScript στην ετικέτα <head> μιας HTML σελίδας

- Σε αυτό το παράδειγμα μια JavaScript συνάρτηση τοποθετείται στην ετικέτα <head>.
- Η συνάρτηση πυροδοτείται όταν ο χρήστης κάνει κλικ σε ένα κουμπί.
- Κάνοντας κλικ στο κουμπί, το περιεχόμενο του HTML στοιχείου αλλάζει.

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML
= "Aris_Garmpis - Paragraph changed.";
}
</script>
</head>

<body>
<h1>JavaScript in Head</h1>
<p id="demo">A Paragraph.</p>

<button type="button"
onclick="myFunction()">Try it</button>

</body>
</html>
```

JavaScript στην ετικέτα <body> μιας HTML σελίδας

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript in Body</h1>
<p id="demo">A Paragraph.</p>

<button type="button"
onclick="myFunction()">Try it</button>

<script>
function myFunction() {

document.getElementById("demo").innerHTM
L = " Aris_Garmpis - Paragraph
changed.";
}
</script>

</body>
</html>
```

- Ομοίως, μια JavaScript συνάρτηση τοποθετείται στην ετικέτα <body>.
- Η συνάρτηση πυροδοτείται όταν ο χρήστης κάνει κλικ σε ένα κουμπί.
  - Κάνοντας κλικ στο κουμπί, το περιεχόμενο του HTML στοιχείου αλλάζει .
- Οποιοδήποτε πλήθος scripts μπορούν να τοποθετηθούν είτε στην ετικέτα <head>, είτε στην ετικέτα <body>. Επίσης, μπορούν να τοποθετηθούν ταυτόχρονα και στα δύο.
- Μια καλή πρακτική είναι να τοποθετείται ο κώδικας JavaScript στο ίδιο σημείο.

JavaScript σε εξωτερικό αρχείο JavaScript

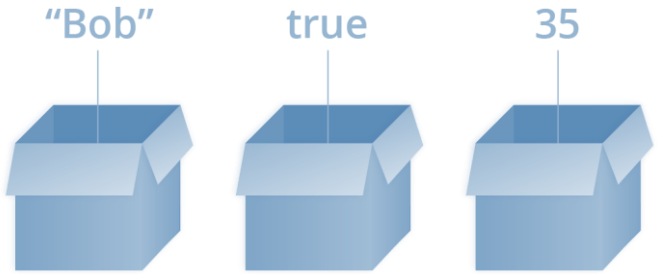
```
<!DOCTYPE html>
<html>
<body>

<h1>External JavaScript</h1>
<p id="demo">A Paragraph.</p>

<button type="button"
onclick="myFunction()">Try it</button>

<p><strong>Note:</strong> myFunction is
stored in an external file called
"myScript.js".</p>

<script src="myScript.js"></script>

</body>
</html>
```

- Τα JavaScript scripts μπορούν επίσης να τοποθετηθούν σε ένα εξωτερικό αρχείο.
- Η χρήση εξωτερικών αρχείων είναι καλή πρακτική όταν ο ίδιος JavaScript κώδικας χρησιμοποιείται σε πολλές HTML σελίδες.
  - Τα αρχεία JavaScript έχουν κατάληξη ".js".
- Για τη χρήση εξωτερικού αρχείου, απλά ορίστε το όνομα του αρχείου στο χαρακτηριστικό *src* της ετικέτας <script>.

```
function myFunction() {
 document.getElementById("demo").innerHTML = "
Aris_Garmpis - Paragraph changed.";
}
```

myScript.js

Μεταβλητές (1/2)



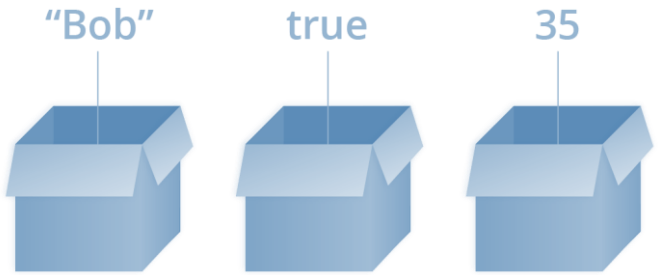
- Οι μεταβλητές στη JavaScript χρησιμοποιούνται για την αποθήκευση δεδομένων.

- Η δήλωση μεταβλητών γίνεται με τη χρήση της δεσμευμένης λέξης (keyword) *var*.

```
var x = 5;
var y = 6;
var z = x + y;
var pi = 3.14;
var person = "John Doe", carName = "Volvo";
```

Η ανάθεση τιμής σε μια μεταβλητή γίνεται με τον τελεστή ανάθεσης '='.

Μεταβλητές (2/2)



- Η JavaScript είναι case sensitive.

```
var lastName = "Doe";
var lastname = "Peterson";
```

Οι μεταβλητές lastName και lastname είναι δυο διαφορετικές μεταβλητές.

```
VAR lastName = "Doe";
Var lastname = "Peterson";
```

Τα Var και VAR δεν ερμηνεύονται ως το keyword var που χρησιμοποιείται για τη δήλωση μεταβλητών.

- Τα ονόματα μεταβλητών:
  - Μπορούν να περιέχουν γράμματα, αριθμούς, _ και $.
  - Πρέπει να ξεκινάνε με γράμμα.
  - Δεν μπορούν να έχουν σαν όνομα δεσμευμένες λέξεις της JavaScript όπως το var.

Σχόλια

- Για την εισαγωγή σχολίων σε μια γραμμή (single line) χρησιμοποιείται το //.

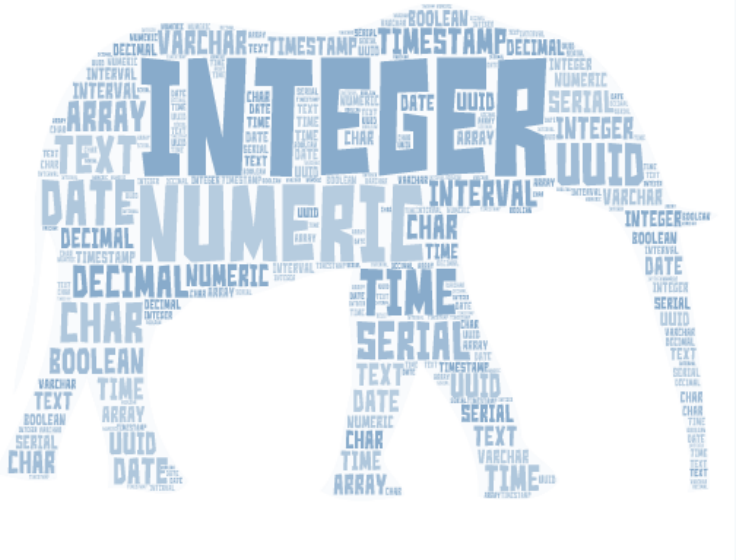
```
var x = 5;
// var x = 6;    Comment
```

- Για σχόλια που εκτείνονται σε περισσότερες από μια γραμμές (multi-line) χρησιμοποιείται το /* ... */.

```
/*
The code below is
declaring a variable.
*/
var x=5.1;
```

Τύποι Δεδομένων

- Οι βασικοί τύποι δεδομένων είναι:
  - Αριθμητικός (number).
  - Αλφαριθμητικός (string).
  - Λογικός (boolean).
  - Πίνακας (array).
  - Αντικείμενο (object).

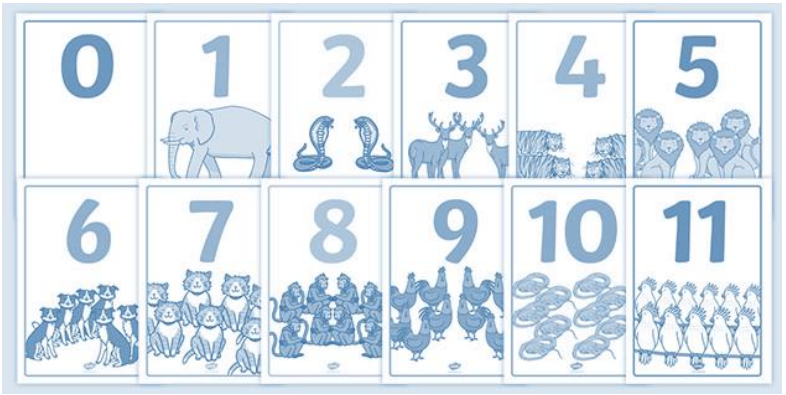


Αριθμητικός Τύπος Δεδομένων

- Αριθμητικός τύπος δεδομένων: οι αριθμοί γράφονται με ή χωρίς δεκαδικά ψηφία.

```
var x = 15;
var pi = 3.14;

var y = 123e5          // 12300000
var y = 123e-5;        // 0.00123
```

Για πολύ μεγάλους ή πολύ μικρούς αριθμούς χρησιμοποιείται η επιστημονική σημειογραφία.



Αλφαριθμητικός Τύπος Δεδομένων

- Αλφαριθμητικός τύπος δεδομένων

```
var carName = "Volvo XC60";
var carName = 'Volvo XC60';
```

Τα αλφαριθμητικά τοποθετούνται μέσα σε εισαγωγικά (quotes), διπλά ή μονά.

```
var answer = "He is called 'Johnny'";
var answer = 'He is called "Johnny"';
```

Όταν η τιμή του αλφαριθμητικού περιέχει διπλά ή μονά εισαγωγικά, μπορώ να τα ορίσω κανονικά, αρκεί αυτά να μην είναι ίδια με τα περιβάλλοντα εισαγωγικά.

```
var x = 'It\'s alright';
var y = "We are the \"Vikings\"."
```

Ωστόσο, σε περίπτωση που θέλω να χρησιμοποιήσω εισαγωγικά ανεξαρτήτως των περιβαλλόντων εισαγωγικών, μπορώ να χρησιμοποιήσω το χαρακτήρα διαφυγής '\'.

Λογικός Τύπος Δεδομένων

- Λογικός (boolean) τύπος δεδομένων: παίρνει δυο τιμές, αληθές ή ψευδές.

```
var x = true;
var y = false;
var q = (10 < 9);
var z = Boolean(11 > 9);
```

Επιστρέφει true ή false ανάλογα με την αποτίμηση του τελεστή που χρησιμοποιείται στην έκφραση.

Ομοίως, η συνάρτηση Boolean(..) επιστρέφει true ή false ανάλογα με το αποτέλεσμα της έκφρασης που της δίνεται ως όρισμα.

Τύπος Δεδομένων Πίνακα (1/2)

- Πίνακας (array): χρησιμοποιείται για να αποθηκεύσει πολλαπλές τιμές σε μια μόνο μεταβλητή.
  - Η δήλωση ενός πίνακα γίνεται με τη χρήση των […].

```
var array_name = [item1, item2, … , itemn];
```

  - Παραδείγματα:

```
var cars = ["Saab", "Volvo", "BMW"];
cars[0] = "Opel";
var name = cars[0];

var x = cars.length;
```

Η πρόσβαση στα στοιχεία ενός πίνακα γίνεται με τη χρήση **αριθμητικών** δεικτών (indexes).
Η αρίθμηση ξεκινάει από το μηδέν.

Η built-in ιδιότητα length επιστρέφει το πλήθος των στοιχείων ενός πίνακα.

Τύπος Δεδομένων Πίνακα (2/2)

- Οι τιμές των στοιχείων ενός πίνακα μεταξύ τους μπορεί να είναι και διαφορετικού τύπου δεδομένων.

```
var person = []
person[0] = "John";
person[1] = "Doe";
person[2] = 46;
var x = person.length; // person.length will return 3
var y = person[0];
```

Τύπος Δεδομένων Αντικειμένου (1/3)

- Αντικείμενο (object): χρησιμοποιείται για να αποθηκεύσει πολλαπλές τιμές σε μια μόνο μεταβλητή.
  - Σε ένα αντικείμενο μπορούμε να ορίσουμε (user-defined) ιδιότητες (properties).
  - Σε ένα αντικείμενο μπορούμε να ορίσουμε (user-defined) μεθόδους (methods) οι οποίες επιτελούν μια εργασία πάνω στα αντικείμενα.

- Η δήλωση ενός αντικειμένου γίνεται με τη χρήση των {…}.

```
var object_name = {property1:value1,…, propertyn:valuen};
```

Τύπος Δεδομένων Αντικειμένου (2/3)

- Παράδειγμα αντικειμένου (object).

```
var person = {
    firstName:"John",
    lastName:"Doe",
    age:50,
    eyeColor:"blue"
    fullName : function() {
       return this.firstName + " " +
this.lastName;
};

var x = person["lastName"];
var y = person.lastName;

name = person.fullName();
```

Δημιουργία ενός αντικειμένου person με:
- Τέσσερις ιδιότητες: firstName, lastName, age, eyeColor.
- Μία μέθοδο fullname που επιστρέφει το ονοματεπώνυμο.

Η πρόσβαση στις ιδιότητες ενός αντικειμένου γίνεται ως εξής:
- `objectName["propertyName"]`
  ή
- `objectName.propertyName`

Η πρόσβαση σε μια μέθοδο ενός αντικειμένου γίνεται ως εξής:
- `objectName.methodName()`

Τύπος Δεδομένων Αντικειμένου (3/3)

- Όλοι οι διαθέσιμοι τύποι δεδομένων είναι στην πραγματικότητα αντικείμενα.

- Ο πίνακας είναι ένας ειδικός τύπος αντικειμένου.
  - Ο πίνακας χρησιμοποιεί αριθμούς ως δείκτες για την πρόσβαση στα στοιχεία του.
  - Το αντικείμενο χρησιμοποιεί ονόματα ιδιοτήτων (αλφαριθμητικά) ως δείκτες για την πρόσβαση στα στοιχεία του.

Διαφορά Πίνακα - Αντικειμένου

- Διαφορά πίνακα - αντικειμένου.

```
var person = []
person[0] = "John";
person[1] = "Doe";
person[2] = 46;

var x = person.length; // person.length will return 3

var y = person[0];     // person[0] will return "John"
```

Δημιουργία ενός πίνακα – γίνεται χρήση αριθμών ως δεικτών για την πρόσβαση στα στοιχεία του.

```
var person = [];
person["firstName"] = "John";
person["lastName"] = "Doe";
person["age"] = 46;

var x = person.length; // person.length will return 0

var y = person[0]; // person[0] will return undefined
```

Αν γίνει χρήση αλφαριθμητικών ως δεικτών κατά τη δήλωση των στοιχείων του πίνακα, η JavaScript το θεωρεί ως αντικείμενο και η ιδιότητα length καθώς και η πρόσβαση στα στοιχεία του πίνακα με δείκτη αριθμό, δεν θα επιστρέφει αποτέλεσμα.

Τελεστές (Operators) (1/4)

- Αριθμητικοί:

Τελεστής	Περιγραφή
=	Ανάθεση τιμής σε μεταβλητή
+	Πρόσθεση τιμών
-	Αφαίρεση τιμών
*	Πολλαπλασιασμός τιμών
/	Διαίρεση τιμών
%	Υπόλοιπο ακέραιας διαίρεσης
++	Μοναδιαία αύξηση
--	Μοναδιαία μείωση

Τελεστές (Operators) (2/4)

- Αλφαριθμητικοί:

Τελεστής	Περιγραφή
+	Συνένωση αλφαριθμητικών

```
var sMonth = "August";
var nYear = 2001;
var nDay = 1;
var sDate = sMonth + " " + nDay + ", " + nYear


var y = "4" + 5;
var z = "Hello" + 5;
```

Η τιμή της sDate είναι ‘August 1, 2001’.

Η πρόσθεση αλφαριθμητικού με αριθμό έχει ως αποτέλεσμα ένα αλφαριθμητικό.

Η τιμή του y είναι ’45’ και του z είναι ‘Hello5’

Τελεστές (Operators) (3/4)

- Σύγκρισης: χρησιμοποιούνται σε λογικές δηλώσεις και επιστρέφουν αληθές (true) ή ψευδές (false) ανάλογα με το αποτέλεσμα της λογικής έκφρασης.

Τελεστής	Περιγραφή
==	Ισότητα με βάση την τιμή
===	Ισότητα με βάση τον τύπο και την τιμή
!=	Διαφορετική τιμή από
!==	Διαφορετική τιμή ή τύπο από
>	Μεγαλύτερο από
>=	Μεγαλύτερο ή ίσο από
<	Μικρότερο από
<=	Μικρότερο ή ίσο από

Έστω var x = 5;

Οι δηλώσεις
var z = (x===5)
var z = (x==5)
θα επιστρέψουν true.

Η δήλωση
var z = (x==“5”)
θα επιστρέψει true ενώ η δήλωση
var z = (x===“5”)
θα επιστρέψει false.

Τελεστές (Operators) (4/4)

- Λογικοί: χρησιμοποιούνται σε λογικές δηλώσεις και επιστρέφουν αληθές (true) ή ψευδές (false) ανάλογα με το αποτέλεσμα της έκφρασης.

Τελεστής	Περιγραφή
&&	Λογικό AND
||	Λογικό OR
!	Λογικό NOT

Δομή If-Else

- Δομή If…else (conditional statement).

```
if (condition)
{
   block of code to be executed if the condition is true
}
else
{
   block of code to be executed if the condition is false
}
```

- Παράδειγμα:

```
if (time < 16)
{
    greeting = "Good day";
}
else
{
    greeting = "Good evening";
}
```

Δομή else-if

- Δομή else…if (conditional statement).

```
if (condition1)
{
    block of code to be executed if condition1 is true
}
else if (condition2)
{
    block of code to be executed if the condition1 is false and
condition2 is true
}
else
{
    block of code to be executed if all the above conditions are false
}
```

- Παράδειγμα:

```
if (time < 12) {
    greeting = "Good morning";
} else if (time < 16) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

Δομή Switch (1/2)

- Δομή Switch: πραγματοποιούνται διαφορετικές ενέργειες με βάση διαφορετικές συνθήκες.

```
switch(expression)
{
    case 1:
        code block
        break;
    case 2:
        code block
        break;
        …
    case n:
        code block
        break;
    default:
        default code block
}
```

Αρχικά, γίνεται η αποτίμηση της λογικής έκφρασης.

Στη συνέχεια, η τιμή της λογικής έκφρασης συγκρίνεται με τις τιμές κάθε περίπτωσης (case).

Αν υπάρχει ταίριασμα, εκτελείται το αντίστοιχο τμήμα κώδικα αλλιώς εκτελείται το τμήμα κώδικα της default περίπτωσης.

Η δεσμευμένη λέξη break οδηγεί στη διακοπή της εκτέλεσης από το συγκεκριμένο switch block.

Δομή Switch (2/2)

- Παράδειγμα εφαρμογής της δομής Switch.

Η μέθοδος Date().getDay() επιστρέφει την τρέχουσα ημέρα σαν έναν αριθμό από 0-6.

```
var day;
switch (new Date().getDay())
{
        case 0:
            day = "Sunday";
            break;
        case 1:
            day = "Monday";
            break;
        case 2:
            day = "Tuesday";
            break;
        case 3:
            day = "Wednesday";
            break;
        case 4:
            day = "Thursday";
            break;
        case 5:
            day = "Friday";
            break;
        case  6:
            day = "Saturday";
            break;
}
```

Δομές Επανάληψης

- Υποστηρίζονται οι ακόλουθες δομές επανάληψης:
  - for loop: επανάληψη ενός μπλοκ κώδικα σε συγκεκριμένο αριθμό βημάτων.
  - for / in: επανάληψη ενός μπλοκ κώδικα με βάση το πλήθος των ιδιοτήτων ενός αντικειμένου.
  - while: επανάληψη ενός μπλοκ κώδικα όσο μια συνθήκη είναι αληθής.
  - do … while: επανάληψη ενός μπλοκ κώδικα όσο μια συνθήκη είναι αληθής.

For loop

- Δομή επανάληψης for loop:

```
for(init; bool_condition; incr)
{
    //code block
}
```

- Παράδειγμα:

```
for (i = 0; i < 5; i++)
{
    text += "The number is " + i + ".";
}
```

For/in loop

- Δομή επανάληψης for/in loop:

```
var x;
for(x in object_name)
{
    //code block
}
```

- Παράδειγμα:

```
var person = {fname:"John", lname:"Doe", age:25};

var text = "";
var x;
for (x in person) {
    text += person[x];
}
```

Η τιμή της μεταβλητής text είναι
'John Doe 25'.

While loop

- Δομή επανάληψης while loop:

```
while (condition)
{
    code block to be executed
}
```

- Παράδειγμα:

```
var i=0;
while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

Η έξοδος θα είναι:

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

Do...while loop

- Δομή επανάληψης do...while loop: το τμήμα κώδικα θα εκτελεστεί τουλάχιστον μία φορά.

```
do
{
    code block to be executed
}
while (condition);
```

- Παράδειγμα:

```
var i=0;
do {
    text += "The number is " + i;
    i++;
}
while (i < 10);
```

Η έξοδος θα είναι:

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

Συναρτήσεις (Functions) (1/3)

- Μια συνάρτηση (function) είναι ένα σύνολο εντολών (block) οι οποίες υλοποιούν μία συγκεκριμένη λειτουργία.

- Μια συνάρτηση εκτελείται όταν συμβαίνει ένα γεγονός (event) το οποίο την πυροδοτεί.

```
function functionName(parameter1, parameter2, …, parametern)
{
    code to be executed
}
```

Συναρτήσεις (Functions) (2/3)

- Σύνταξη συνάρτησης: με τη δεσμευμένη λέξη function.

```
function functionName(parameter1, parameter2, …, parametern)
{
    code to be executed
}
```

Μια συνάρτηση μπορεί να έχει παραμέτρους.
Στο σώμα της συνάρτησης, οι παράμετροι χρησιμοποιούνται ως τοπικές μεταβλητές.

Συναρτήσεις (Functions) (3/3)

- Σύνταξη συνάρτησης: επιστρεφόμενη τιμή.

```
function myFunction(a, b)
{
    return a * b;
}

var x = myFunction(4, 3);
```

Μια συνάρτηση μπορεί να επιστρέφει ένα αποτέλεσμα μέσω της εντολής return.

Συναρτήσεις και Εμβέλεια Μεταβλητών

- Υπάρχουν 2 ειδών μεταβλητές, οι τοπικές και οι καθολικές μεταβλητές.
  - Οι τοπικές (local) μεταβλητές δηλώνονται μέσα στη συνάρτηση και μπορούν να είναι προσβάσιμες μόνο από αυτή.
  - Οι καθολικές (global) μεταβλητές δηλώνονται έξω από τις συναρτήσεις και μπορούν να είναι προσβάσιμες από όλα τα scripts και τις συναρτήσεις της HTML σελίδας.

Πότε εκτελείται μια συνάρτηση;

- Μία συνάρτηση για να εκτελεστεί πρέπει:
  - Να πυροδοτηθεί από ένα γεγονός (event).
    - Π.χ. όταν ο χρήστης κάνει κλικ σε ένα κουμπί.
  - Να γίνει αναφορά σε αυτή (δηλαδή να κληθεί) μέσα στον κώδικα JavaScript.
- Μια συνάρτηση μπορεί να δηλωθεί έτσι ώστε να εκτελείται αυτόματα (self-invoked).

Συναρτήσεις και Γεγονότα

- Πυροδότηση συνάρτησης από ένα γεγονός (event).
  - Π.χ. όταν ο χρήστης κάνει κλικ σε ένα κουμπί.

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to display the date.</p>

<button onclick="displayDate()">The time is?</button>

<script>
function displayDate() {
    document.getElementById("demo").innerHTML = Date();
}
</script>

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

</body>
</html>
```

Όταν ο χρήστης κάνει κλικ στο κουμπί, εμφανίζεται η τρέχουσα ημερομηνία.

Η συνάρτηση **`document.getElementById("demo")`** βρίσκει το HTML στοιχείο με id="demo" και αλλάζει το περιεχόμενο του στην τρέχουσα ημερομηνία.

Τι είναι το γεγονός (events) (1/2)

- Κάθε HTML στοιχείο έχει κάποια γεγονότα (events) που ενεργοποιούνται από ενέργειες του χρήστη.
  - Τα events εμφανίζονται σαν attributes των HTML tags.

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to display the date.</p>

<button onclick="...">The time is?</button>

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

</body>
</html>
```

Γεγονότα (Events) (2/2)

- Μερικά από τα πιο γνωστά events είναι:

Event	Περιγραφή
onclick	Ο χρήστης κάνει κλικ σε ένα HTML στοιχείο
onmouseover	Ο χρήστης μετακινεί το ποντίκι πάνω σε ένα HTML στοιχείο
onmouseout	Ο χρήστης μετακινεί το ποντίκι έξω από ένα HTML στοιχείο
onchange	Ένα HTML στοιχείο αλλάζει
onload	Ο browser έχει τελειώσει με τη φόρτωση της σελίδας

- Μπορούμε να ορίσουμε συναρτήσεις JavaScript, οι οποίες θα εκτελούνται μόλις ενεργοποιηθεί ένα συγκεκριμένο event.

Πυροδότηση συνάρτησης από τον κώδικα JavaScript

- Πυροδότηση συνάρτησης με αναφορά σε αυτή μέσα στον κώδικα JavaScript.

```
<!DOCTYPE html>
<html>
<body>

<p>
myFunction returns the product of the arguments (a ,b):
</p>
<p id="demo"></p>

<script>
function myFunction(a, b) {
    return a * b;
}
document.getElementById("demo").innerHTML =
myFunction(10, 2);
</script>

</body>
</html>
```

Όταν ο browser θα φορτώσει την σελίδα θα εκτυπώσει το αποτέλεσμα της συνάρτησης, δηλαδή το 20.

Η συνάρτηση **`document.getElementById("demo")`** βρίσκει το HTML στοιχείο με id="demo" και αλλάζει το περιεχόμενο του σε 20.

Αυτόματη εκτέλεση συνάρτησης

- Μια συνάρτηση μπορεί να δηλωθεί έτσι ώστε να εκτελείται αυτόματα (self-invoked).

```
<!DOCTYPE html>
<html>
<body>

<p>Function invoked automatically </p>

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

<script>
(function testFunction()
{
    document.getElementById("demo").innerHTML = "Hello!
    I called myself";
})();
</script>

</body>
</html>
```

Για να οριστεί η αυτόματη εκτέλεση μιας συνάρτησης, χρησιμοποιούνται οι παρενθέσεις.

Όταν ο browser θα φορτώσει τη σελίδα, η συνάρτηση **`document.getElementById("demo")`** βρίσκει το HTML στοιχείο με id="demo" και αλλάζει το περιεχόμενο του σε "Hello! I called myself".

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Ερωτήσεις;