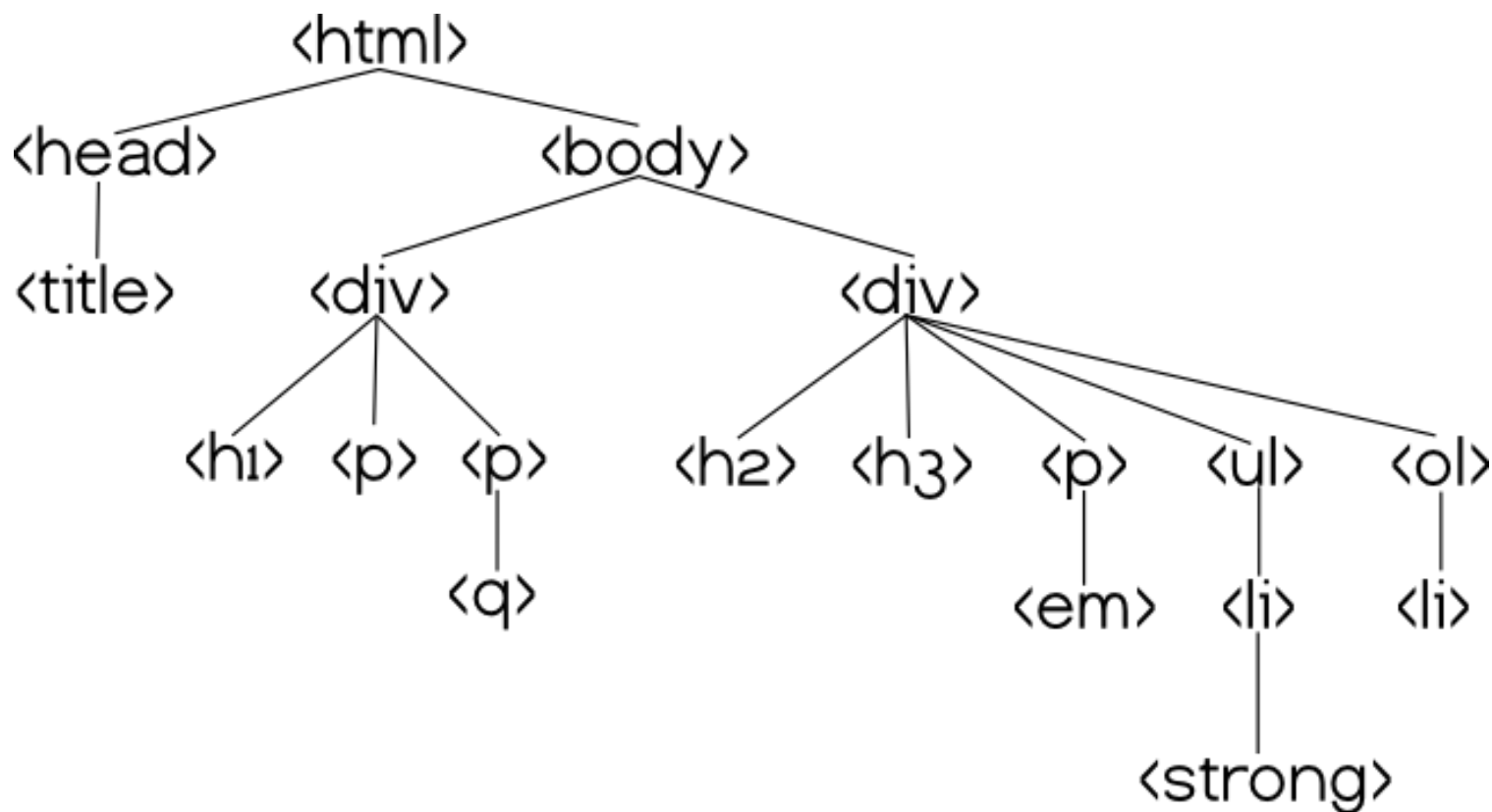


ΠΡΑΚΤΙΚΗ ΣΤΗΝ ΕΙΔΙΚΟΤΗΤΑ

HTML, CSS & JS

ΔΕΛΤΑ 360 

Ιεραρχία από Στοιχεία (elements)



Πώς χρησιμοποιείται;

- Χρήση της οδηγίας `<script>` και τοποθέτηση του κώδικα μες στο αρχείο με τον κώδικα html

```
<script>
    function printMessage() {
        alert("Message.");
    }
</script>
```

- Τοποθέτηση σε ξεχωριστό αρχείο και συμπίληψη αυτού στο αρχείο html

```
<script src="file.js"></script>
```

συμβαντα στην html

Συμβάντα στην HTML

- Ορίζονται διάφορα συμβάντα (events) στην HTML, όπως:
 - Για το πάτημα του αριστερού πλήκτρου πάνω σε ένα στοιχείο μίας φόρμας (`onclick`)
 - Για τη φόρτωση της σελίδας HTML (`onload`)
 - Για την υποβολή μίας φόρμας (`onsubmit`)
 - Για την αλλαγή της τιμής κάποιου στοιχείου μίας φόρμας (`onchange`)

Συμβάντα και JavaScript

- Δίνεται η δυνατότητα κλήσης κάποιας συνάρτησης της JavaScript, όταν γίνει κάποιο συμβάν, π.χ.

```
<form>
```

```
  <input type="checkbox" onclick="clicked();">
```

```
  <input type="submit" value="OK">
```

```
</form>
```

Μόλις ο χρήστης κάνει κλικ στο checkbox, θα καλεστεί η συνάρτηση `clicked()` που έχει ορίσει ο χρήστης.

Html dom

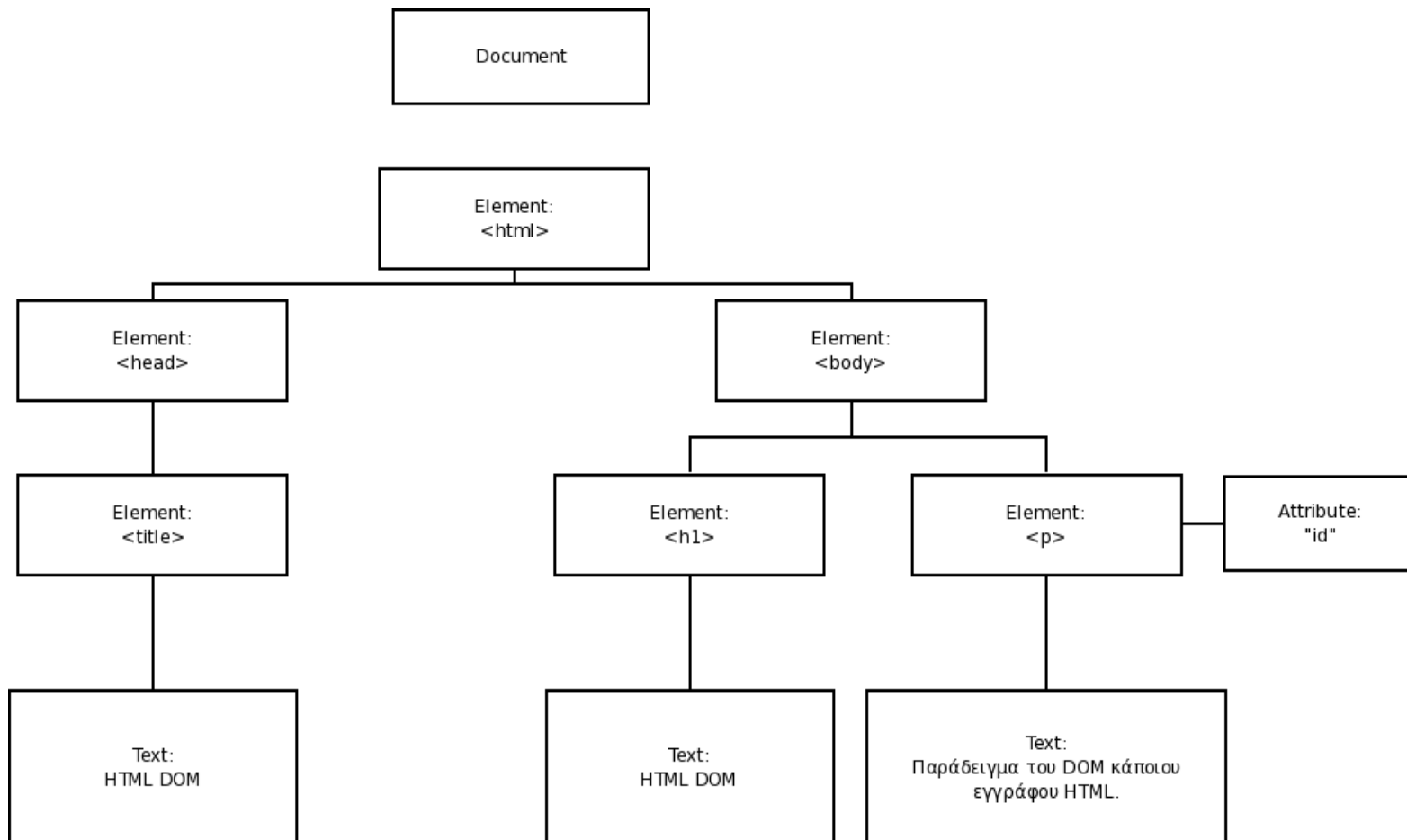
HTML DOM

- Document Object Model
- Αποτελεί μία δενδρική δομή, όπου κάθε οδηγία, παράμετρος αυτής και απλό κείμενο σε ένα έγγραφο HTML αναπαρίσταται με ένα κόμβο.
- Παιδιά του κόμβου είναι οι οδηγίες που αυτός περιέχει.

Παράδειγμα HTML

```
<html>
  <head>
    <title>HTML DOM</title>
  </head>
  <body>
    <h1>HTML DOM</h1>
    <p id="example">Παράδειγμα του DOM για κάποιο
έγγραφο HTML</p>
  </body>
</html>
```

Αντίστοιχο Διάγραμμα DOM



HTML DOM

- Κάθε κόμβος αντιστοιχεί σε κάποιο αντικείμενο, το οποίο έχει τις κατάλληλες ιδιότητες και μεθόδους για το χειρισμό της οδηγίας, όπως για το χειρισμό των παραμέτρων της, π.χ. ένα αντικείμενο που αντιστοιχεί σε checkboxes έχει (μεταξύ άλλων) τις ιδιότητες checked, name και id.

HTML DOM

- Δύο χρήσιμες συναρτήσεις είναι οι:
 - `getElementById("id")`: επιστρέφει ένα αντικείμενο που αντιστοιχεί στο στοιχείο του οποίου η παράμετρος `id` έχει τιμή "id"
 - `getElementByName("name")`: επιστρέφει ένα αντικείμενο που αντιστοιχεί στο στοιχείο του οποίου η παράμετρος `name` έχει τιμή "name"

Παράδειγμα Χειρισμού του DOM από JavaScript

- Για το προηγούμενο έγγραφο HTML θα μπορούσαμε να γράψουμε:

```
function addtext(){  
    document.getElementById("example").innerHTML =  
        "<b>Νέο</b> κείμενο";  
}
```

- `<button onclick="addtext()">clickme</button>`
- `<p id="example"></p>`

Javascript - βασικά

- Η JavaScript είναι μία διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού με ορισμένες ιδιότητες και επηρεασμούς από αντικειμενοστραφείς γλώσσες προγραμματισμού όπως η C, C++ και η Java,
- χωρίς όμως να μπορεί να χαρακτηριστεί ως μία αντικειμενοστραφής γλώσσα προγραμματισμού.
- Είναι γλώσσα υψηλού επιπέδου (high-level), δυναμική και χωρίς τύπους ουσιαστικά (untyped).

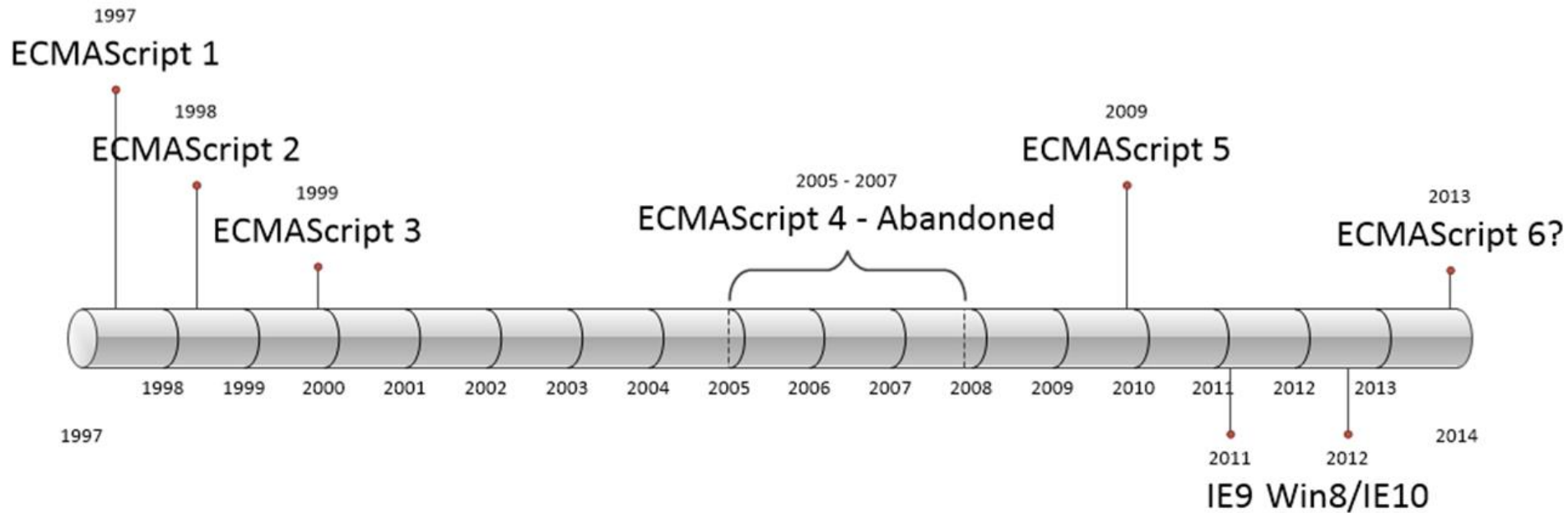
JavaScript and Java

- JavaScript and Java are completely different languages, both in concept and design. JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997. ECMA-262 is the official name. ECMAScript 6 (released in June 2015) is the latest official version of JavaScript.

JavaScript and Java

JavaScript	Java
Ερμηνευμένη (Interpreted) από τον client	Μεταγλωττισμένη (Compiled) από τον προγραμματιστή, τρέχει στον client
Κώδικας ενσωματωμένος σε HTML σελίδες	τα java Applets δεν είναι ενσωματωμένα στα HTML κείμενα
Χαλαροί τύποι δεδομένων (loose typing of data types)	Αυστηροί τύποι δεδομένων (strong typing of data types)
Δυναμική διασύνδεση (dynamic binding)	Στατική διασύνδεση (Static binding)
Περιορισμός σε λειτουργίες του client browser	Εφαρμογές αυτόνομες και πλήρης (Stand-alone applications)
Σχετίζεται/λειτουργεί με τα HTML αντικείμενα (HTML elements)	Ξεπερνάει τον περιορισμένο ορίζοντα της HTML (για παράδειγμα multimedia)
Προσπελαύνει αντικείμενα του browser και τη λειτουργικότητα του	Δεν έχει προσπέλαση στα αντικείμενα του browser και τις λειτουργίες του

History of JavaScript

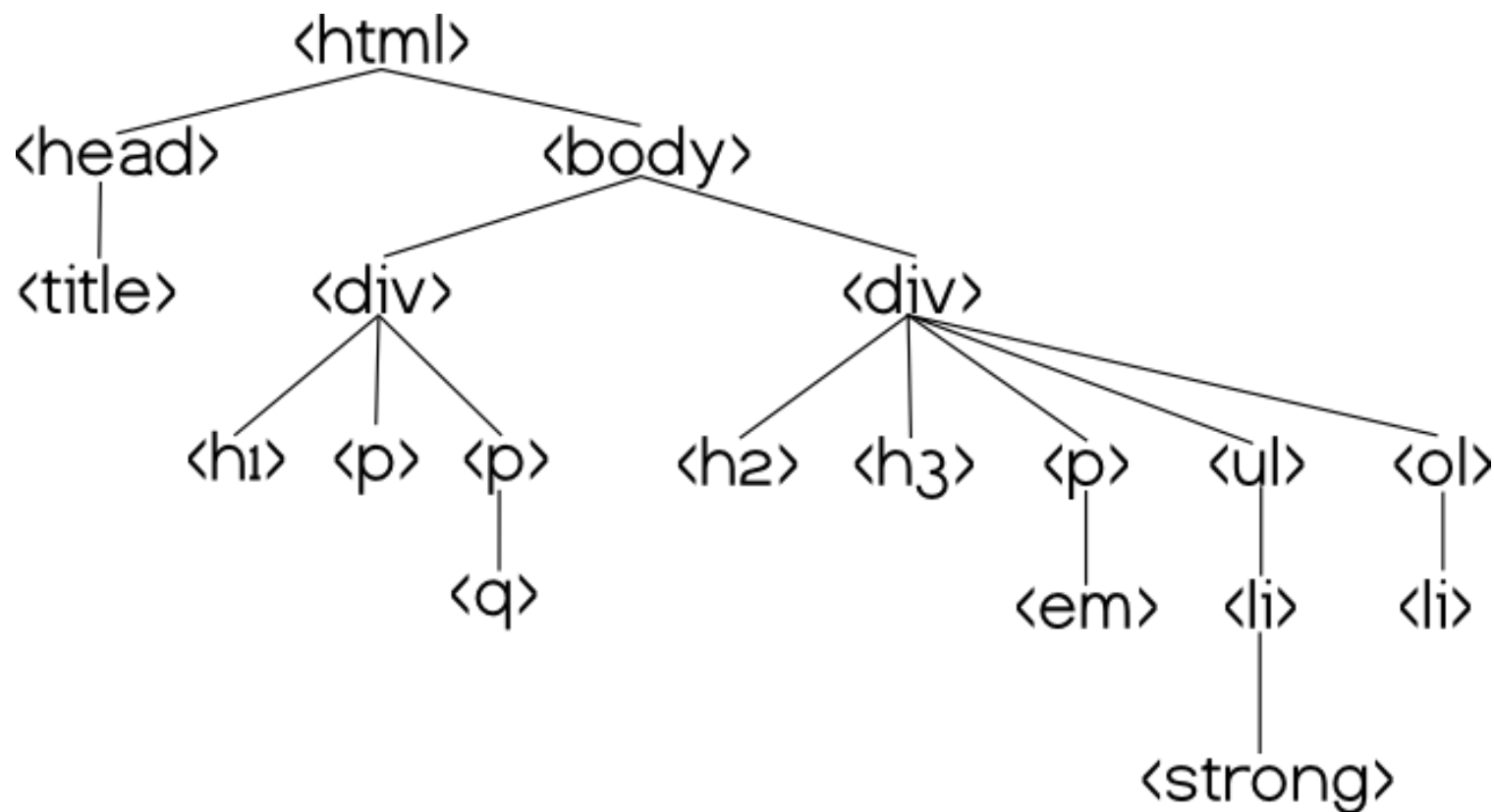


JavaScript is one of the **4 languages** and technologies all web developers **must** learn:

1. **HTML** to define the content of web pages
2. **CSS** to specify the layout of web pages
3. **JavaScript** to program the behavior of web pages
4. **Document Object Model (DOM)** to model and access HTML document

DOM –

Ιεραρχία από Στοιχεία (elements)



Δημιουργία απλών scripts

- Για να γίνει η εισαγωγή ενός client-side script (είτε είναι γραμμένο με JavaScript ή με οποιαδήποτε άλλη scripting γλώσσα) θα πρέπει με κάποιο τρόπο να "ενσωματωθούν" οι εντολές του μέσα σε κάποιο αρχείο HTML.
- Για να γίνει αυτό χρησιμοποιείται το HTML στοιχείο (σήμανση) `<SCRIPT>`.

Δημιουργία απλών scripts

- Χρησιμοποιώντας το HTML στοιχείο `<SCRIPT>` με τον όρο `client-side script` αναφερόμαστε με μία λέξη σε κάθε διερμηνεύσιμο πρόγραμμα (`script`) το οποίο εκτελείται (διερμηνεύεται γραμμή-γραμμή) στον `web browser`.
- Ο όρος αυτός είναι σε αντίθεση με τον όρο `server-side script` με τον οποίο αναφερόμαστε σε κάθε διερμηνεύσιμο πρόγραμμα (`script`) το οποίο εκτελείται (διερμηνεύεται γραμμή-γραμμή) στον `web server`.

Τι κάνω με τη Javascript σε μία ιστοσελίδα

- Τα πάντα !!!
- Να αλλάξω περιεχόμενο
- Να αλλάξω μορφοποίηση
- Να μετακινήσω/εμφανίσω/εξαφανίσω περιεχόμενο
- Να κάνω input validation
- Να δημιουργήσω web 2.0 εφαρμογές
- Γενικά τα πάντα **απαιτούν** τη χρήση Javascript

ενσωμάτωση script κώδικα

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title>Ένα HTML αρχείο που ενσωματώνει script!</title>
```

```
</head>
```

```
<body>
```

```
<script language="JavaScript" src="myscript.js"  
type="text/javascript">
```

```
</script>
```

```
</body>
```

```
</html>
```

Πώς, Πού και Πότε ?

- **Πώς;**
 - Το είδαμε στις προηγούμενες διαφάνειες
- **Που πρέπει να τοποθετούνται τα scripts;**
- **Πότε ένα script εκτελείται;**

Τα θεμελιώδη

- Statements
- Variables
- Operators
- Functions

Statements

- if...else
- switch
- while
- do...while
- for
- for...in
 - It is used to enumerate the properties of an object.
(cf. All object has a method `propertyIsEnumerable()`.)
 - for (*property in expression*) *statement*
- with
 - A very slow segment. It is best to avoid using it.
- Label, break and continue
- try...catch...finally
- throw

Τα θεμελιώδη

- Statements
- Variables
- Operators
- Functions

Keywords & Reserved Words

- The keywords and reserved words cannot be used as variables or function names.

- Keywords

break, case, catch, continue, default, delete, do, else, finally, for, function, if, in, instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with

Τα θεμελιώδη

- Reserved Words

abstract, boolean, byte, char, class, const, debugger, double, enum, export, extends, final, float, goto, implements, import, int, interface, long, native, package, private, protected, public, short, static, super, synchronized, throws, transient, volatile

Μεταβλητές

- Το όνομα μιας μεταβλητής αποτελείται από λατινικούς χαρακτήρες, ψηφία και το χαρακτήρα «_», π.χ.

```
myVariable, num1, my_car
```

- Μία μεταβλητή δημιουργείται, όταν της ανατεθεί πρώτη φορά μία τιμή, π.χ.

```
form = document.forms[0];
```

- Οι μεταβλητές είναι καθολικές, εκτός εάν χρησιμοποιηθεί η λέξη-κλειδί `var`, π.χ.

```
var form = document.forms[0];
```

Τύποι Δεδομένων

- Αριθμοί (ακέραιοι και κινητής υποδιαστολής)
- Booleans
- Συμβολοσειρές
- Arrays
- Object. Μεταβλητές για αναφορά σε αντικείμενα
 - Function
 - Null
 - Undefined

Τύποι Δεδομένων

- Δεν είναι απαραίτητη η δήλωση του τύπου κάποιας μεταβλητής. Αντίθετα, αυτός βρίσκεται αυτόματα.
- Ο τύπος της μεταβλητής αλλάζει βάσει της τιμής την οποία αποθηκεύει.

Primitive and Reference Values

■ Primitive Values

- Primitive values are simple pieces of data that are stored on the *stack*,
- which is to say that their value is stored directly in the location that the variable accesses.
- The value is one of the JavaScript primitive types:
 - *Undefined, Null, Boolean, Number, or String.*
- Many languages consider strings as a reference type and not a primitive type, but JavaScript breaks from this tradition.

■ Reference Values

- Reference values are objects that are stored in the *heap*,
- meaning that the value stored in the variable location is a pointer to a location in memory where the object is stored.

Primitive Types

- JavaScript has five primitive types:
 - Undefined
 - The Undefined type has only one value, *undefined*.
 - Null
 - The Null type has only one value, *null*.
 - Boolean
 - The Boolean type has two values, *true* and *false*.
 - Number
 - 32-bit integer and 64-bit floating-point values.
 - Infinity → `isFinite()`
 - NaN (Not a Number) → `isNaN()`
 - String
 - Using either double quotes("") or single quote(').
 - JavaScript has no character type.

Ζωή μίας μεταβλητής (scope)

- Οι μεταβλητές στη JavaScript μπορούν να είναι είτε local όταν ορίζεται μέσα σε μία συνάρτηση, είτε global όταν ορίζεται οπουδήποτε μέσα σε κάποιο <SCRIPT> στοιχείο έξω όμως από συναρτήσεις. Μία local μεταβλητή που ορίζεται μέσα σε μία συνάρτηση είναι "ορατή" μόνο μέσα σε αυτή.
- Μία global μεταβλητή που ορίζεται έξω από μία συνάρτηση είναι "ορατή" από οπουδήποτε μέσα στο HTML αρχείο που φιλοξενεί το JavaScript.
- Μία local μεταβλητή με το ίδιο όνομα με μία global μεταβλητή, "καλύπτει" (overrides) την global μεταβλητή για τη διάρκεια ζωής της (δηλ. μέσα στη συνάρτηση).

Τα θεμελιώδη

- Operators
- Υποστηρίζονται οι κλασικοί τελεστές που υποστηρίζονται από τις σύγχρονες γλώσσες προγραμματισμού.
- Για τη συνένωση συμβολοσειρών χρησιμοποιείται ο τελεστής «+».

Τα θεμελιώδη

- Functions
- Για τον ορισμό των συναρτήσεων χρησιμοποιείται η λέξη-κλειδί `function` ακολουθούμενη από το όνομα της συνάρτησης και τα ορίσματα αυτής σε παρενθέσεις. Το σώμα της συνάρτησης περικλείεται μεταξύ αγκίστρων.

Βασικά Native Objects

- String
- Date
- Array
- Math
-

Strings

- [String variables and String objects](#)
- Better do not use String Objects, slows down execution
- [String methods](#)

Numbers

- [JavaScript numbers](#) are always stored as double precision floating point numbers, following the international IEEE 754 standard.
- [Number methods](#)

The Math Object

- Το αντικείμενο Math δίνει πρόσβαση σε μία σειρά από μαθηματικές ρουτίνες χρήσιμες σε περιπτώσεις που χρειάζεστε ανάλογες δυνατότητες για μαθηματικά προβλήματα.

Ιδιότητες (χρήσιμες μαθηματικές σταθερές)	Μέθοδοι (χρήσιμες μαθηματικές συναρτήσεις)
E	abs acos asin atan atan2 ceil cos exp
LN2	floor log max min
LN10	pow
LOG2E	random
LOG10E	round
PI	sin
SQRT1_2	sqrt
SQRT2	

Το αντικείμενο Date

- Με το αντικείμενο Date έχουμε τη δυνατότητα να δουλεύουμε με ημερομηνίες & ώρες. Με αυτά τα αντικείμενα και τις πολλές μεθόδους που υποστηρίζουν μπορούμε να πάρουμε οποιοδήποτε μέρος μιας ημερομηνίας & ώρας.

Πίνακες

Εντολές Ελέγχου

- Υπάρχουν οι γνωστές εντολές ελέγχου
 - `if...else`
 - `switch`

if...else

- Η πλήρης μορφή είναι:

```
if (condition1) {  
    statements1;  
} else if (condition2) {  
    statements2;  
} else {  
    statements4;  
}
```

- Τα τμήματα `else if` και `else` είναι προαιρετικά.
- Μπορεί να υπάρχουν όσα τμήματα `else if` επιθυμεί ο προγραμματιστής.

switch

- Η πλήρης μορφή της εντολής είναι:

```
switch (expression) {  
    case value1:  
        statements1;  
        break;  
    case value2:  
        statements2;  
        break;  
    default:  
        statements3;  
        break;  
}
```

- Μπορεί να υπάρχουν όσα cases απαιτούνται.

Εντολές Βρόχων

- Υποστηρίζονται οι βρόχοι με τις εντολές επανάληψης `for`, `do...while` και `while`, καθώς και οι εντολές `break` και `continue`.
- Υπάρχει επίσης η εντολή βρόχου `for ... in`

for

- Η γενική μορφή της εντολής είναι:

```
for (initial-statement; test; increment) {  
    statements;  
}
```

- Π.χ.

```
for (i = 0; i < 10; ++i) {  
    document.write(i);  
}
```

do...while

- Η γενική μορφή της εντολής είναι:

```
do {  
    statements;  
} while (condition);
```

- Π.χ.

```
i = 0;  
do {  
    document.write(i);  
} while (i < 10);
```


while

- Η γενική μορφή του της εντολής είναι:

```
while (condition) {  
    statements;  
}
```

- Π.χ.

```
i = 0;  
while (i < 10) {  
    document.write(i);  
    i++;  
}
```

for ... in

- Η γενική μορφή της εντολής είναι:

```
for variable in object {  
    statements;  
}
```

- Π.χ.

```
var i = 0;  
for form in document.forms {  
    i++;  
}
```

Σχόλια

- Δύο τρόποι εισαγωγής σχολίων
 - `// this is a comment`
 - `/* this is a comment */`

Αντικείμενα

- Κάθε αντικείμενο στη JavaScript έχει
 - Ιδιότητες: μεταβλητές που ανήκουν στο αντικείμενο, π.χ.

```
checkbox.checked = true;  
myCar.color = "red";
```
 - Μεθόδους: συναρτήσεις που ανήκουν στο αντικείμενο, π.χ.

```
document.write("A message");  
button.click();
```

Δημιουργία Αντικειμένων

- Για τη δημιουργία κάποιου αντικειμένου που ανήκει σε κάποια κλάση χρησιμοποιείται η λέξη-κλειδί `new`, π.χ.

```
fruits = new Array();
```

```
today = new Date();
```

- Η JavaScript παρέχει πολλούς χρήσιμους τύπους, όπως για δημιουργία πινάκων (`Array`), επεξεργασία ημερομηνιών (`Date`), συμβολοσειρές (`String`)

Δημιουργία Νέων Τύπων

- Δημιουργία ενός αντικειμένου της κλάσης Object και προσθήκη ιδιοτήτων σε αυτό, π.χ.

```
car = new Object();  
car.color = "red";
```

- Δημιουργία συνάρτησης που επιστρέφει αντικείμενο, π.χ.

```
function Contact(name, surname) {  
    this.name = name;  
    this.surname = surname;  
}  
contact = new Contact("John", "Doe");
```

Παραδείγματα για Αντικείμενα

```
function printContact() {  
    return this.name + " " + this.surname;  
}  
function Contact(name, surname, phone) {  
    this.name = name;  
    this.surname = surname;  
    this.phone = phone;  
    this.print = printContact;  
}  
contact = new Contact("John", "Doe", "5551234");  
document.write(contact.print());
```

JavaScript Debugging

- Για να κάνουμε την ανάπτυξη web εφαρμογών πιο αποτελεσματική χρησιμοποιούμε μία σειρά από εργαλεία (web development tools). Τα εργαλεία αυτά είναι διαφορετικά από τους HTML editors που είδαμε νωρίτερα σε αυτό το βιβλίο. Δεν βοηθούν στην άμεση δημιουργία μιας ιστοσελίδας όπως οι editors παρά είναι εργαλεία που χρησιμοποιούνται για τον έλεγχο της χρήσης και των σφαλμάτων που πιθανόν να αντιμετωπίζει η διεπαφή μιας ιστοσελίδας ή μιας web εφαρμογής.
- https://en.wikipedia.org/wiki/Web_development_tools

JavaScript Debugging

[Google Chrome](#): Επιλογή *Web Developer Tools*

[Internet Explorer](#): Επιλογή *F12 Web Developer Tools*
(από την έκδοση 8 και μετά)

[Safari](#): *Safari Web Development Tools*^[7] (από την έκδοση 3)

[Firefox](#): *F12 Web Console / Browser Console* (από την έκδοση Firefox 4)

Microsoft Edge: Εργαλεία προγραμματισμού (F12)

[Opera](#) – επιλογή *Opera Dragonfly*