



ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ:

1. HTML
2. CSS

ΠΕΡΙΕΧΟΜΕΝΑ:

1. Ιστορική Αναδρομή
2. Τέσσερις Πτυχές της JS (που θα μάθουμε)
3. Ενσωμάτωση Scripts
 1. Inline Scripts
 2. Εξωτερικά Αρχεία
 3. async και defer
4. Αλληλεπίδραση με τον Interpreter - Σχολιασμός Κώδικα
5. Έξοδος στην Κονσόλα

ΜΑΘΗΜΑ 1: Εισαγωγή στη JavaScript

1. Ιστορική Αναδρομή

- **1991:** Tim Burners-Lee: HTML + HTTP + Browser
- **1994:** Lie & Boss: Draft της CSS
- **1995 - 2001: Ο 1ος πόλεμος των Browsers:**
 - Ανταγωνιστές: Internet Explorer vs Netscape Navigator
 - IE: Microsoft
 - NN: Netscape. Το 1998, θα κάνει open-source τον browser και θα μετασηματιστεί αργότερα στον Mozilla Firefox.
 - [Για την ιστορία: Νίκησε ο IE]
- **1995: Εισάγεται η JS ως ένας τρόπος για να προσθέσουμε προγράμματα στον browser Netscape Navigator**
 - εκτοτε υιοθετήθηκε από όλους του μεγάλους browsers
 - Δεν έχει καθόλου να κάνει με τη Java (υιοθετήθηκε το όνομα για λόγους μάρκετινγκ)
- **ECMAScript:** Ο οργανισμός ECMA (European Computer Manufacturers Organization - Non Profit) έχει αναλάβει να θέτει το τι είναι στάνταρ στη γλώσσα (ECMAScript==JavaScript).
 - ECMAScript 3 (1999): Επικρατούσα στα 00s
 - ECMAScript 4: Δεν βγήκε ποτέ, πολλές φιλόδοξες αλλαγές που προτεινόταν, αποδεικνύονταν κακές στην πράξη
 - ECMAScript 5 (2009): Ελάχιστες αλλαγές
 - **ECMAScript 6 (2015): Τεράστιες αλλαγές, πολλές βελτιώσεις στη γλώσσα.**
 - Εκτοτε μία νέα βέρσιον κάθε χρόνο, ECMA2016, ECMA2017, κ.ο.κ. με μικρές γενικά αλλαγές.
 - Συντομογραφικά αναφέρονται και ως ES6, ES2017 κ.ο.κ.



Brendan Eich

Σχεδόν Ανεκδοτολογική Ιστορία:

- Η Netscape ζήτησε από τον Brendan Eich που εργαζόταν σε αυτήν, μία γλώσσα προγραμματισμού που θα ενσωματωθεί στο browser με deadline 10 ημερών.
- Ο Eich έφτιαξε ένα κράμα που πατούσε στις εξής ιδέες:
 - Η σύνταξη έμοιαζε αρκετά με τη Java (μόνον αυτό. Κατά τ'άλλα δεν έχει καμία σχέση, παρά το κοινό πρόθεμα)
 - Στοιχεία Συναρτησιακού Προγραμματισμού από τη Scheme
 - Στοιχεία Αντικειμενοστραφούς Προγραμματισμού από τη Self
 - Είναι πολύ ανεκτική σε προγραμματιστικά λάθη, ώστε ακόμη κι αν υπάρχει κάποιο bug, να προβληθεί η σελίδα, έστω και άστοχα.
- Μνημειώδης αστοχία σε πολλά σημεία της γλώσσας:
 - Η πιο ανισόρροπη δημοφιλής γλώσσα προγραμματισμού, μέχρι το 2015, που διορθώθηκαν πολλές αστοχίες της και πλέον, σήμερα, είναι μια σχεδόν «φυσιολογική» γλώσσα.

1. Είναι Μία Γλώσσα Προγραμματισμού

- Συνεπώς έχει το δικό της τρόπο για να ορίζει τα συνήθη στοιχεία μια γλώσσας:
 - Μεταβλητές και Δυναμικοί Τύποι Δεδομένων, Πίνακες
 - Δομες Επιλογής και Επανάληψης
 - Συναρτήσεις
 - OOP (Αντικειμενοστρεφής) και Συναρτησιακή
 - Χωρισμός σε Modules
- Είναι τελείως ιδιαίτερη:
 - Σύνταξη: Μοιάζει με Java
 - Μεταβλητές - Τιμές: Μοιάζει με Python
 - OOP: Εντελώς Ιδιαίτερη (prototype inheritance)
 - Συναρτησιακή: Μοιάζει με τη Scheme

2. Ο τρόπος που τρέχει

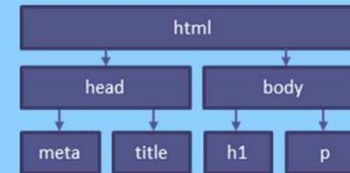
- Η HTML περιέχει/συνδέει κώδικα JS
- Ο browser φορτώνει τη σελίδα
- Τρέχει τον κώδικα σε έναν ενσωματωμένο διερμηνευτή (interpreter) του Browser. Ονομάζεται **JS Engine**.
 - Π.χ. ο Google Chrome περιέχει το διερμηνευτή V8 και ο FireFox τον SpiderMonkey.



- Γι' αυτό λέμε ότι γράφουμε scripts τα οποία θα εκτελούνται από το διερμηνευτή και θα επηρεάζουν τη σελίδα HTML

3. Αλληλεπιδρά με το DOM (Document Object Model)

- DOM: Ένα αντικείμενο που περιέχει την οργάνωση της HTML σελίδας.



- Ένα ενδεικτικό σενάριο χρήσης που μπορούμε να προγραμματίσουμε με JS επί μίας σελίδας:
 - Επίλεξε το κουμπί με id που έχει μια συγκεκριμένη τιμή και αν ο χρήστης κάνει κλικ σε αυτό, να γίνεται μία ενέργεια.

4. Έχει άπειρες βιβλιοθήκες

- JS Standard Library: Βιβλιοθήκες που είναι ενσωματωμένες για συνηθισμένες ενέργειες, π.χ. Δομές Δεδομένων, Δικτύωση, Timers, Διαχείριση Ημ/νιών κ.λπ.
- Βιβλιοθήκες 3^{ων} μερών:



- Vanilla JS: JS χωρίς βιβλιοθήκες 3ων μερών

Υπόθεση γι' αυτήν τη σειρά:

- Η σειρά αυτή δεν έχει φτιαχτεί για κάποιον που κάνει την πρώτη επαφή του με τον προγ/μό. Για πρώτη επαφή με τον προγ/μό, από τις σειρές του καναλιού, συνίσταται η C ή η Python.

Α' τρόπος: Inline Scripts

- Το στοιχείο script χρησιμοποιείται για να ενσωματώσουμε δεδομένα ή κώδικα στη σελίδα μας (στο head ή στο body)
- Έχει πολλές προχωρημένες χρήσεις.
- Στην απλούστερη χρήση του:

```
<script>
...
</script>
```

- Είναι ισοδύναμο με το:

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

- που ορίζει ότι η χρησιμοποιούμενη γλώσσα είναι η JS.

Συντακτικά Στοιχεία:

- (Αν και θα τα δούμε πολύ αναλυτικά στα επόμενα μαθήματα, εισάγουμε εδώ κάποια στοιχεία, για τη συνέχεια του μαθήματος)
- Δήλωση Μεταβλητής:

```
let variableName = value;
```

- Συμβολοσειρά:
 - Αναπαρίσταται σε μονά ή διπλά εισαγωγικά
 - Με το +, κάνουμε συνένωση συμβολοσειρών
- Η μέθοδος:

```
document.write(string);
```

- Γράφει τη συμβολοσειρά string στο σημείο που καλείται επί της ροής του εγγράφου HTML

Παράδειγμα 1: inline

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Psounis JS Tutorial</title>
  <script>
    let hello = "Hello World!";
  </script>
</head>

<body>
  <h1>Our First JS Page</h1>
  <script>
    document.write("<p>" + hello + "</p>");
  </script>
</body>
</html>
```

Our First JS Page

Hello World!

Πως τρέχει στον Browser:

- Γίνεται το HTTP αίτημα και κατεβαίνει η σελίδα HTML
- Ο browser αρχίζει το parsing της σελίδας
 - Όταν συναντά script:
 - Διακόπτει το parsing και εκτελεί τον κώδικα του script
- Ο τρόπος που εκτελούνται τα scripts μπορεί να εξειδικευτεί περαιτέρω με τα *char*/*ca* *async* και *defer* (επόμεν. διαφάνεια)

Παρατηρήσεις:

- Σε πραγματικές σελίδες, σπάνια υπάρχει inline κώδικας JS.
- Ειδικά μάλιστα η document.write(..), είναι χρήσιμη μόνο για toy examples και δεν συνίσταται να χρησιμοποιείται στην πράξη.
- Συνήθως ο κώδικας θα ενσωματώνεται σε ξεχωριστά αρχεία.

Β' τρόπος: Σε ξεχωριστό αρχείο

- Σύνταξη στοιχείου script για εξωτερικά αρχεία:

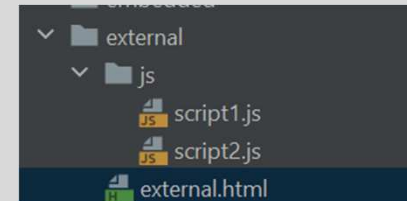
```
<script src="..."></script>
```

- όπου src είναι σύνδεσμος (άρα μπορεί να είναι απόλυτος, σχετικός, ή ακόμη και url), π.χ.:
 - js/script.js
 - σχετικό μονοπάτι προς αρχείο με κώδικα JS (Η επέκταση του αρχείου πρέπει να είναι .js)
 - /js/script.js
 - Σε σχέση με τον κατάλογο ρίζας (βλέπε και HTML - Μάθημα 4)
 - <https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js>
 - url με τη διεύθυνση της βιβλιοθήκης jQuery.

Παρατήρηση:

- Η συνηθισμένη οργάνωση των αρχείων είναι αυτή που φαίνεται στο παράδειγμα (φάκελος js που περιέχει αρχεία JS scripts με επέκταση .js)

Παράδειγμα 2: external



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Psounis JS Tutorial</title>

  <script src="js/script1.js"></script>
</head>

<body>
  <h1>Our First JS Page</h1>

  <script src="js/script2.js"></script>
</body>
</html>
```

```
let hello = "Hello World!";
```

```
document.write("<p>" + hello + "</p>");
```

Our First JS Page

Hello World!

Default Τρόπος:

- Ενώσω γίνεται parsing της σελίδας, όταν διαβάζεται ένα στοιχείο script, το διάβασμα της σελίδας διακόπτεται, κατεβαίνει το script και εκτελείται και έπειτα συνεχίζεται το parsing της σελίδας

defer (< deferred: αναβαλλόμενο)

- Χαρακτηριστικό του στοιχείου script:

```
<script defer src="..."></script>
```
- Όταν ο parser συναντά στοιχείο script που έχει μαρκारιστεί με defer:
 - Συνεχίζει το parsing ενώ παράλληλα κατεβάζει το script.
 - Το script που κατέβηκε, θα τρέξει όταν τελειώσει το parsing της σελίδας

Παράδειγμα 3: defer (βλ. βίντεο)

```
<head>
...
<script defer src="js/script1.js" ></script>
</head>
<body>
...
<script defer src="js/script2.js" ></script>
</body>
```

Παρατήρηση για το defer:

- Με το defer, εγγυημένα, τα scripts θα τρέξουν με τη σειρά που έχουν ενσωματωθεί στο HTML έγγραφο.

async (< asynchronous: ασύγχρονο):

- Χαρακτηριστικό του στοιχείου script:

```
<script async src="..."></script>
```
- Όταν ο parser συναντά στοιχείο script που έχει μαρκарιστεί με async:
 - Συνεχίζει το parsing ενώ παράλληλα κατεβάζει το script.
 - Το script θα τρέξει αμέσως μόλις κατέβει (διακόπτοντας το parsing)

Παράδειγμα 4: async (βλ. βίντεο)

```
<head>
...
<script async src="js/script1.js" ></script>
</head>
<body>
...
<script async src="js/script2.js" ></script>
</body>
```

Παρατήρηση για το async:

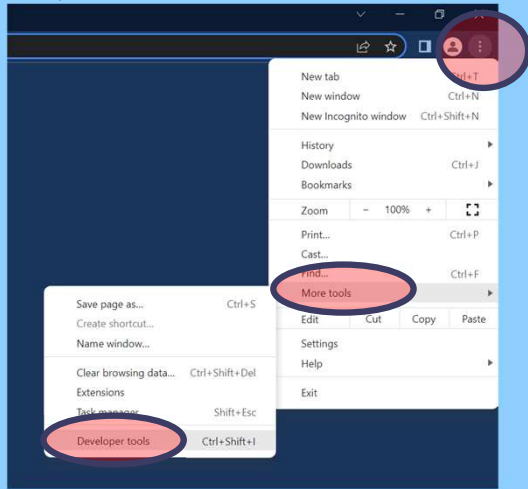
- Με το async, δεν υπάρχει εγγύηση για τη σειρά με την οποία θα τρέξουν τα scripts.

Σημείωση:

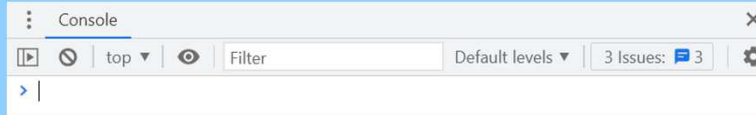
- Το defer, αντικαθιστά την παλιά πολιτική, τα scripts να ενσωματώνονται στο τέλος της σελίδας (πριν το </body>)

Ο Interpreter στον Google Chrome:

- Ανοίγουμε τα Developer Tools (Menu->More Tools->Developer Tools) ή πατάμε Ctrl+Shift+I.



- Ο interpreter είναι έτοιμος να υποδεχτεί τις εντολές μας στο prompt:



Άσκηση 1:

- Γράψτε στον interpreter διαδοχικά τις εντολές (πατώντας Enter μετά από κάθε μία)
 - > let x=1;
 - > let y=2;
 - > let z=x+y;
 - > let w=z*z;
 - > w;

Σημειώσεις:

- Κάθε φορά που πατάμε Enter επιστρέφεται (με αχνό γκρι) και η επιστρεφόμενη τιμή της εντολής (συνήθως undefined - βλ. επόμενο μάθημα)

Άσκηση 2:

- Γράψτε στον interpreter διαδοχικά τις εντολές (πατώντας Enter μετά από κάθε μία)
 - > let openingTag="<p>";
 - > let closingTag="</p>";
 - > let text = "Hello World!";
 - > let element = openingTag + text + closingTag;
 - > element;

Σχολιασμός Κώδικα

- Υπάρχουν δύο είδη σχολίων στη JavaScript:
 - Σχόλια γραμμής: **// A comment...**
 - Σχόλια πολλών γραμμών **/* A multi-line comment */**

Παράδειγμα 7: comments

```
/* This is a multi-line comment
   This is another line
*/
let x = 1; // a single-line comment
//another single-line comment
```

ΜΑΘΗΜΑ 1: Εισαγωγή στη JavaScript

5. Έξοδος στην Κονσόλα

Η μέθοδος console.log(...):

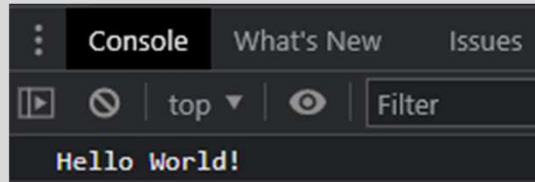
- Τυπώνει το όρισμά της στην κονσόλα.

Παράδειγμα 8: log

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Psounis JS Tutorial</title>

<script>
  console.log("Hello World!");
</script>
</head>

<body>
<h1>JS Tutorial</h1>
</body>
</html>
```



Εναλλακτική σύνταξη της console.log(...):

- Εναλλακτικά, η console.log συντάσσεται ως:
 - **console.log(x1, x2, ..., xn)**
 - Θα τυπώσει όλα τα ορίσματά της στην κονσόλα, αφήνοντας ένα κενό μεταξύ των διαδοχικών ορισμάτων.

Παράδειγμα 9: log2

- Το \n είναι ο χαρακτήρας αλλαγής γραμμής της JS

```
<script>
  console.log("Hello World!", 2, 3, "object");
  console.log("line1\nline2");
</script>
```

```
Hello World! 2 3 object
line1
line2
```

Σημειώσεις για τη συνέχεια της σειράς:

- Χρησιμοποιούμε το ATOM
 - **Οδηγίες Εγκατάστασης στο «HTML - Μάθημα 1»**
 - (και προσοχή στην εγκατάσταση των πακέτων atom-live-server)
- Βλέπουμε το αποτέλεσμα των script μας με εκτυπώσεις με την console.log(..) στην κονσόλα των Developer Tools του Chrome.

Άσκηση 3:

- Προβλέψτε τι θα εκτυπώσει το παρακάτω script:

```
<script>
  console.log(
    "H\nell\no" +
    "W\nor\nld!\n"
  );
</script>
```