

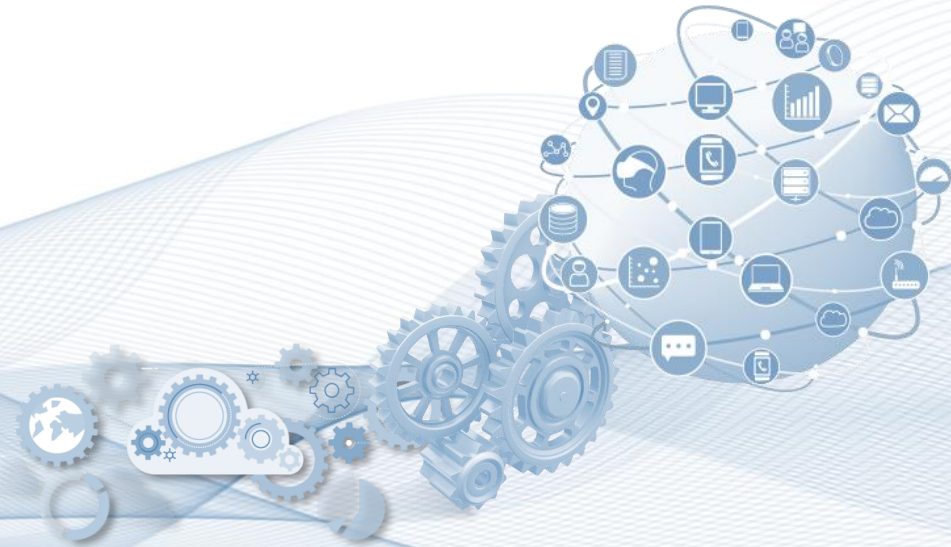
ΤΕΧΝΙΚΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΥΠΟΛΟΓΙΣΤΩΝ

Διάλεξη III

JavaScript – Μια Εισαγωγή

Γιάννης Τζήμας, Καθηγητής

JavaScript



Τι θα δούμε σήμερα...

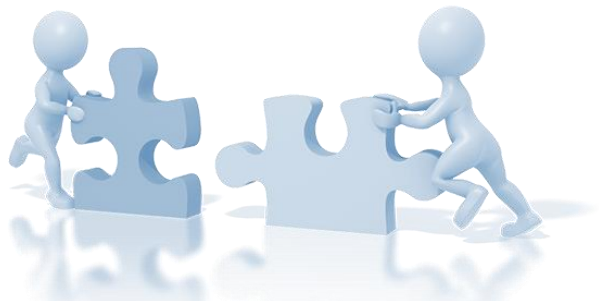
- Βασικός στόχος αυτού του δεύτερου τμήματος του μαθήματος είναι να...
 - προχωρήσουμε **βαθύτερα** στις **τεχνικές προγραμματισμού**, αξιοποιώντας αυτή τη φορά τη γλώσσα προγραμματισμού **JavaScript**.
 - Θα δούμε ότι όσα μάθαμε στο **Scratch** (μεταβλητές, τύποι, συναρτήσεις) υπάρχουν και εδώ, αλλά με άλλο συντακτικό.



Ας ξαναθυμηθούμε το γιατί είμαστε εδώ!

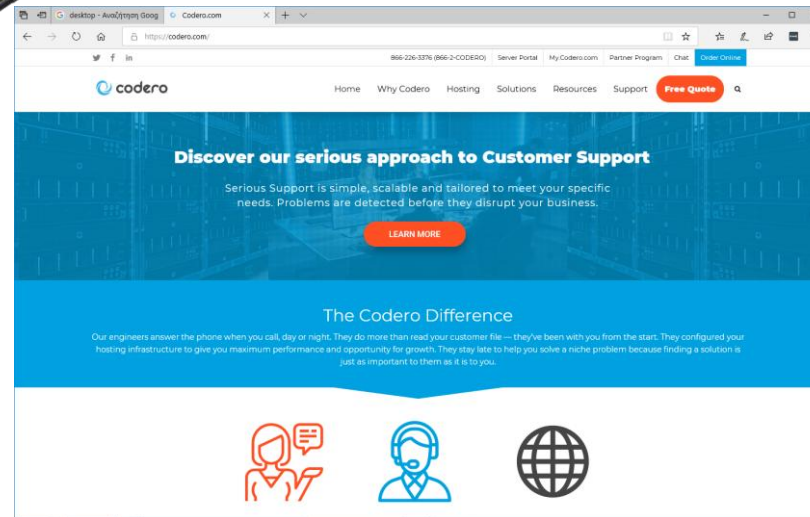
**NOT
JUST
ART**

- Όπως ξαναείπαμε ο προγραμματισμός **ανταποδίδει** όσα του δίνετε, είτε αν τον δείτε σαν
 - **χόμπι**, είτε σαν
 - **επάγγελμα**.
- Εμπεριέχει όπως είπαμε **δυνατότητες επίλυσης προβλημάτων (puzzle solving)**.
 - Αυτό πολλοί το θεωρούν **τέχνη!!!**
- Μπορείτε να μαθαίνετε κάθε μέρα και κάτι νέο σε σχέση με τον προγραμματισμό και **όσο περισσότερα προγράμματα γράφετε, συνεχώς θα βελτιώνεστε!**



Ένα από τα πολύ καλά στον προγραμματισμό σήμερα είναι...

- ...το γεγονός ότι υπάρχουν πάρα **πολλοί τύποι υπολογιστών** και πάρα **πολλοί τύποι προγραμμάτων**...



Όπως ξαναείπαμε...



- Όταν δύο άνθρωποι επιλέγουν να **επικοινωνήσουν** διαλέγουν μία **γλώσσα** που και οι δύο καταλαβαίνουν.
- Ο προγραμματισμός είναι κάτι παρόμοιο:
 - Ένα **πρόγραμμα** είναι ένα σύνολο **εντολών** που ο **υπολογιστής** εκτελεί.
 - Οι **εντολές** είναι σε κάποια **γλώσσα** που **καταλαβαίνει** ο υπολογιστής.
 - Καλούμε αυτές τις γλώσσες, **γλώσσες προγραμματισμού**.



Γλώσσες Προγραμματισμού...

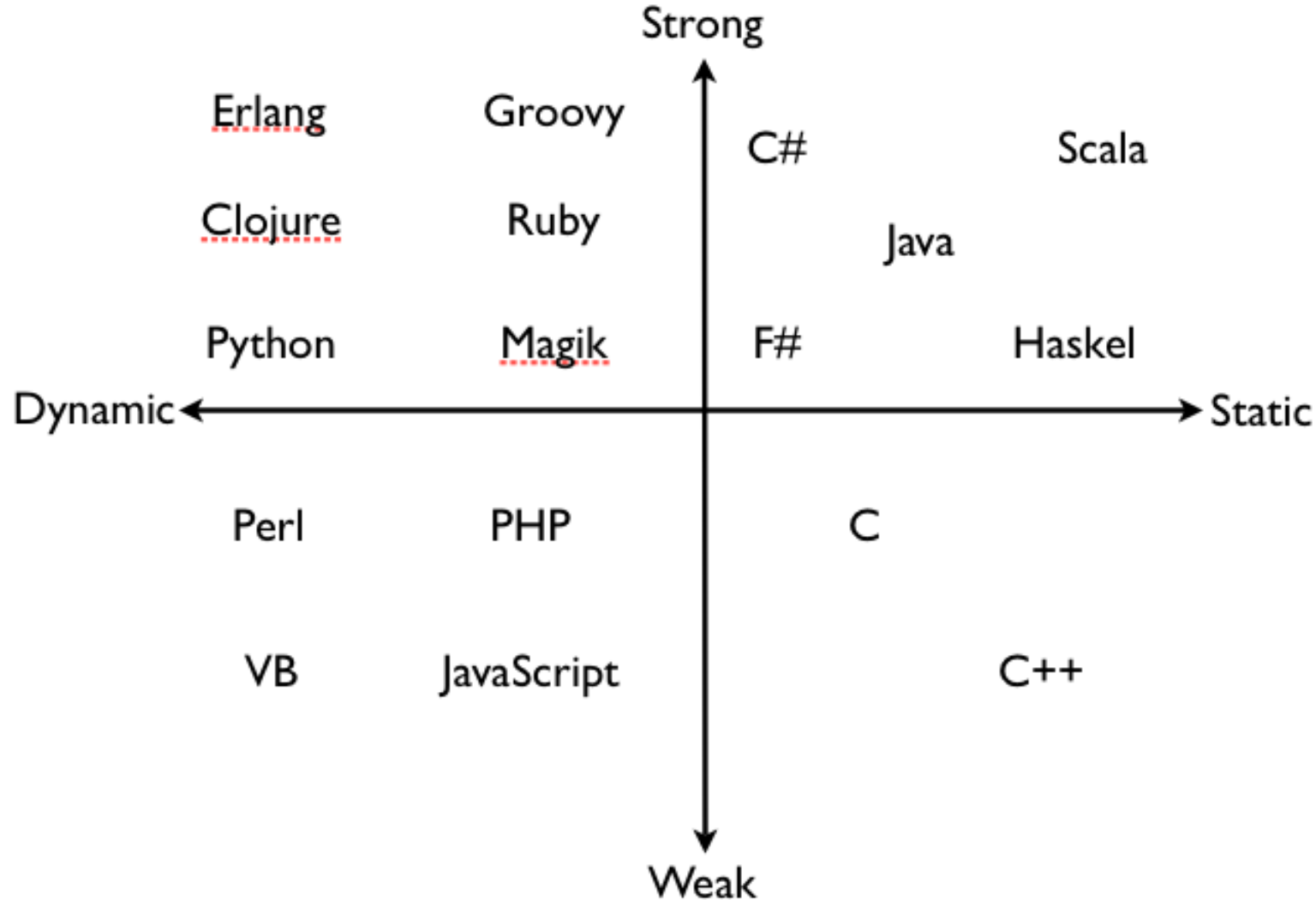


감사합니다 Natick
 Danke Ευχαριστίες Dalu
 Grazie Thank You Köszönöm
 Tack
 Спасибо Dank Gracias
 谢谢 Merci See
 ありがとう



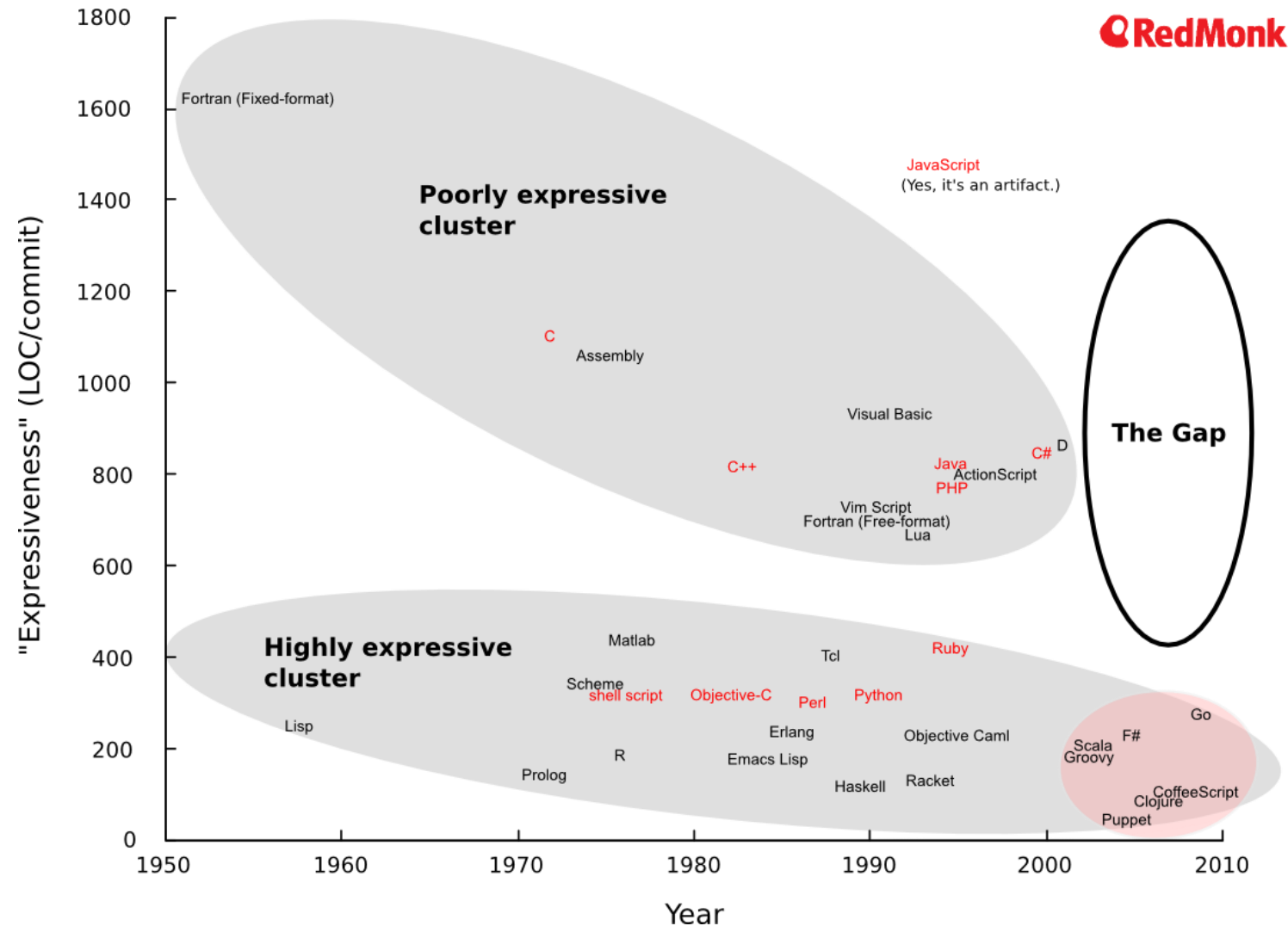
Γλώσσες Προγραμματισμού...

Πολλά κριτήρια ανάλογα με το τι θέλουμε να πετύχουμε...



Γλώσσες Προγραμματισμού...

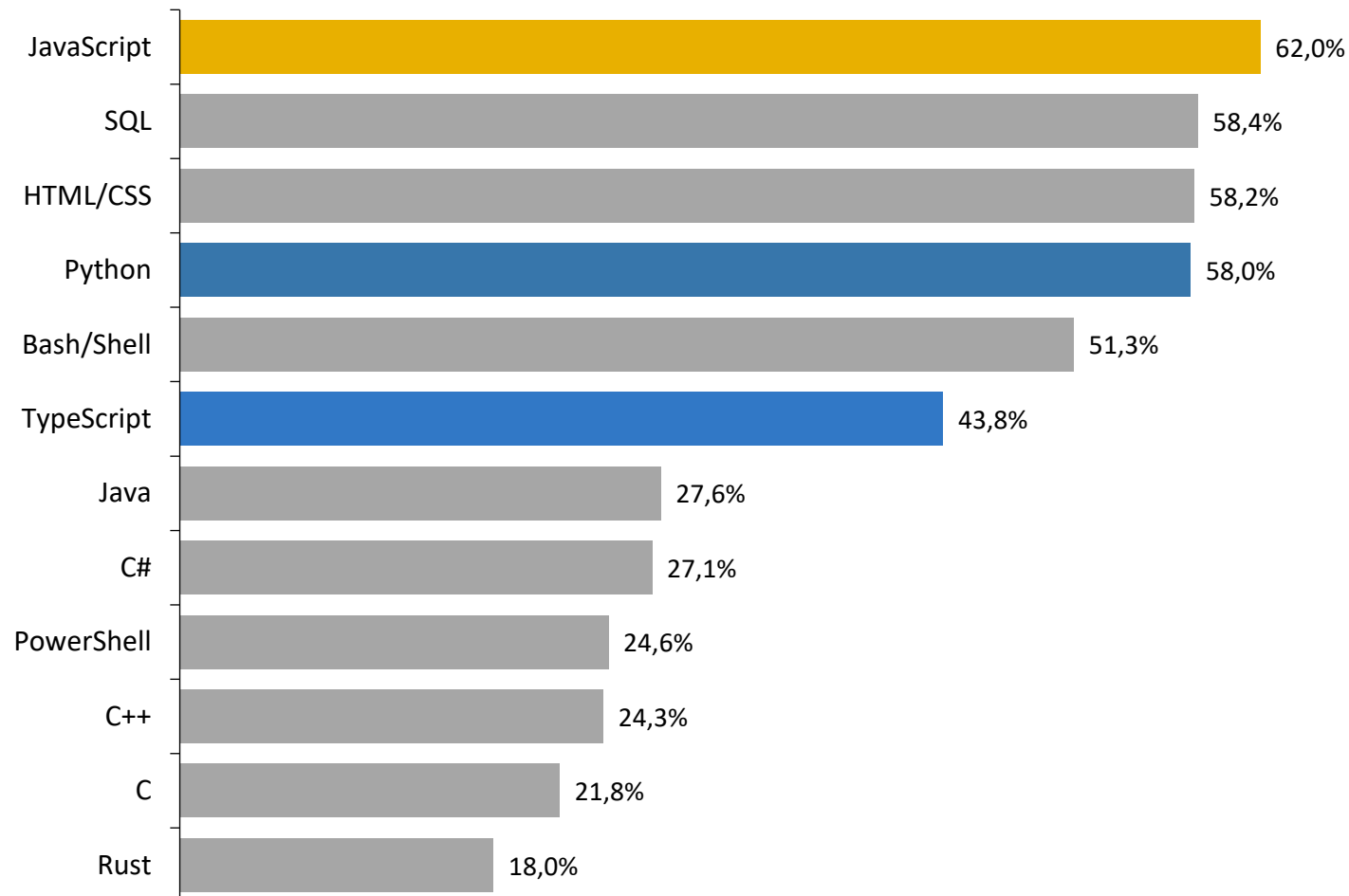
Πολλά κριτήρια ανάλογα με το τι θέλουμε να πετύχουμε...



Γλώσσες Προγραμματισμού...

Πόσο Δημοφιλείς είναι;

Ποιες γλώσσες χρησιμοποιούν οι προγραμματιστές; Stack Overflow Developer Survey 2026



Η **JavaScript** είναι η γλώσσα που χρησιμοποιούν **οι περισσότεροι** προγραμματιστές (σχεδόν 2 στους 3).

Η **TypeScript** είναι JavaScript με τύπους δεδομένων (θα δούμε τι είναι οι τύποι σε λίγο).

Οκτ 2026	Οκτ 2025	Γλώσσα	Rating	Μεταβολή
1	1	Python	19.23%	-5.23%
2	2	C	11.33%	+2.04%
3	3	C++	8.87%	+0.04%
4	4	Java	8.41%	+0.06%
5	5	C#	3.79%	-3.15%
6	6	JavaScript	2.77%	-0.64%
7	7	Visual Basic	2.20%	-1.02%
8	13	R	2.06%	+0.55%
9	10	SQL	1.97%	+0.20%
10	16	Rust	1.44%	+0.24%
11	8	Go	1.39%	-0.53%
12	14	PHP	1.26%	-0.12%
13	11	Fortran	1.12%	-0.58%
14	22	Swift	1.03%	+0.09%
15	18	Scratch	0.97%	-0.18%
16	23	COBOL	0.89%	+0.03%
17	12	Perl	0.88%	-0.78%
18	30	Julia	0.88%	+0.42%
19	9	Delphi/Object Pascal	0.83%	-1.03%
20	17	MATLAB	0.83%	-0.33%

Γλώσσες Προγραμματισμού... Πόσο Δημοφιλείς είναι; (TIOBE)

- Ο δείκτης **TIOBE** μετράει πόσο «ακούγεται» μια γλώσσα: πόσοι μηχανικοί, μαθήματα και εταιρείες σχετίζονται με αυτή, μέσα από αναζητήσεις στο Διαδίκτυο.
- **Δεν** μετράει ποια γλώσσα είναι η «καλύτερη».
- Οκτώβριος 2026: η **JavaScript** είναι 6η – και το **Scratch** 15ο!

Γλώσσες Προγραμματισμού...

Πόσο Δημοφιλείς είναι; (TIOBE)

Γλώσσα	2026	2021	2016	2011	2006
Python	1	2	5	6	8
C	2	1	2	2	2
C++	3	4	3	3	3
Java	4	3	1	1	1
C#	5	5	4	5	7
JavaScript	6	7	7	10	9
Visual Basic	7	6	13	-	-
SQL	8	10	-	-	-
R	9	11	17	26	-
Delphi/Object Pascal	10	19	12	12	10

Μέση θέση κάθε γλώσσας στη διάρκεια του έτους («-»: εκτός λίστας). Πηγή: [tiobe.com](https://www.tiobe.com), Οκτώβριος 2026. Η Python ανέβηκε από την 8η θέση (2006) στην 1η.

Γλώσσες Προγραμματισμού...

Πόσο δημοφιλείς; Εξαρτάται τι μετράμε!

Μέτρηση	Τι μετράει	JavaScript	Python	1η θέση
TIOBE (Οκτ. 2026)	Αναζητήσεις στο Διαδίκτυο: μηχανικοί, μαθήματα, εταιρείες	6η	1η	Python
Stack Overflow Survey 2026	Τι χρησιμοποιούν οι ίδιοι οι προγραμματιστές	1η (62%)	4η (58%)	JavaScript
GitHub Octoverse 2025	Πόσοι γράφουν κώδικα σε αποθετήρια του GitHub	3η	2η	TypeScript (= JavaScript με τύπους)

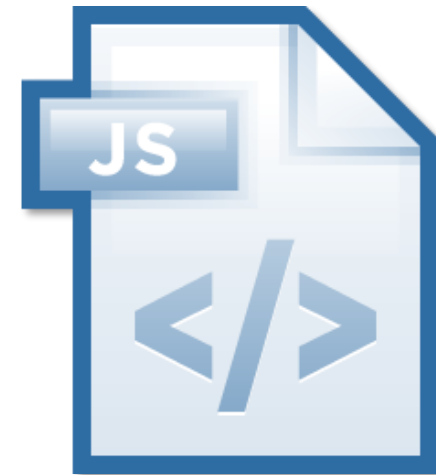
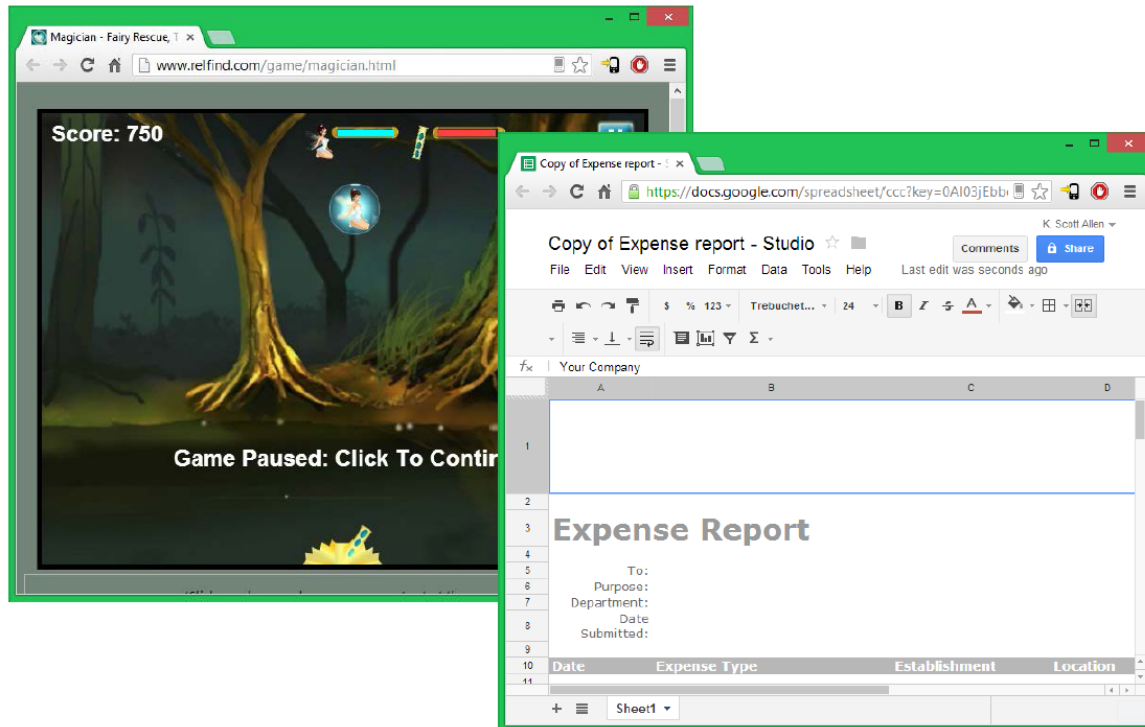
Κάθε μέτρηση δίνει διαφορετική «κατάταξη».

Όλες όμως συμφωνούν ότι η **JavaScript** και η **Python** είναι από τις πιο διαδεδομένες γλώσσες – γι' αυτό τις διαλέξαμε!

Για να συνεχίσουμε το μάθημά μας θα επιλέξουμε τη...

- JavaScript

- Γλώσσα δημοφιλής στον **Παγκόσμιο Ιστό** (World Wide Web - WWW).
- **Εύκολη** στην εκμάθηση και στη χρήση της.
- Τα JavaScript προγράμματα «**τρέχουν**» **οπουδήποτε!**
- Με το **Node.js** τρέχει και έξω από τον browser: servers, εφαρμογές, εργαλεία.



Μία πρώτη γεύση...

My First JavaScript

Fri Oct 16 2026 13:02:39 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

Display Date

Η σελίδα στον browser, αφού πατήσουμε το κουμπί

[Demos_2026-2027/example1.html](#)

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <script>
6  function displayDate() {
7      document.getElementById("demo").innerHTML = Date();
8  }
9  </script>
10 </head>
11 <body>
12
13 <h1>My First JavaScript</h1>
14 <p id="demo">This is a paragraph.</p>
15
16 <button type="button" onclick="displayDate()">Display Date</button>
17
18 </body>
19 </html>
```

Και μία ακόμα...



Click the light bulb to turn on/off the light

Κλικ στη λάμπα: η συνάρτηση αλλάζει την εικόνα

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4  <script>
5  function changeImage() {
6      const element = document.getElementById("myimage");
7      if (element.src.includes("bulbon")) {
8          element.src = "pic_bulboff.gif";
9      } else {
10         element.src = "pic_bulbon.gif";
11     }
12 }
13 </script>
14
15 
17
18 <p>Click the light bulb to turn on/off the light</p>
19
20 </body>
21 </html>
```

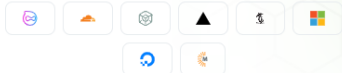
Node – Εγκαταστήστε το...

Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

Get Node.js®

Get security support
for EOL Node.js versions



Node.js is proudly supported by the partners above and more.

Create an HTTP Server Write Tests Read and Hash a File Streams Pipeline Work with Threads

```
1 // server.mjs
2 import { createServer } from 'node:http';
3
4 const server = createServer((req, res) => {
5   res.writeHead(200, { 'Content-Type': 'text/plain' });
6   res.end('Hello World!\n');
7 });
8
9 // starts a simple http server locally on port 3000
10 server.listen(3000, '127.0.0.1', () => {
11   console.log('Listening on 127.0.0.1:3000');
12 });
13
14 // run with `node server.mjs`
```

JavaScript

</> Copy to clipboard

Learn more what Node.js is able to offer with our Learning materials.

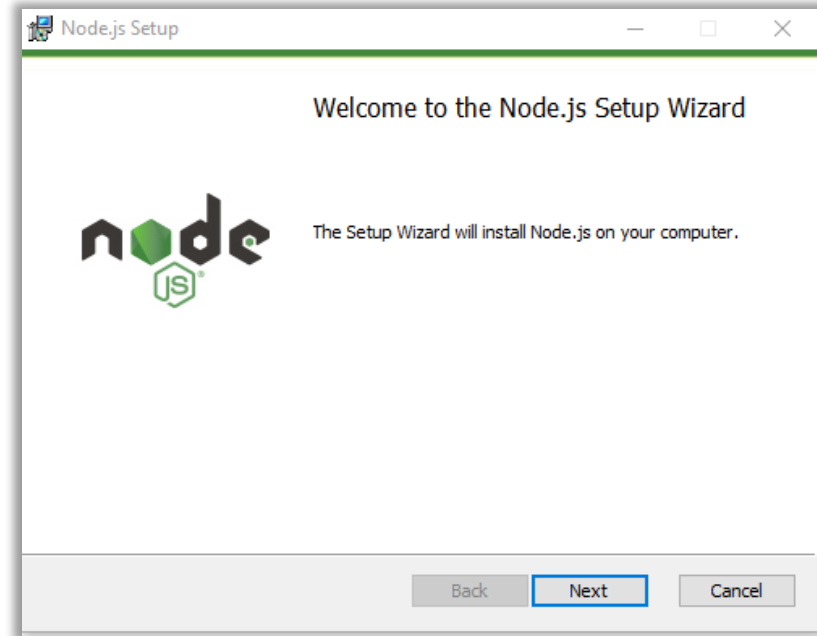
Download Node.js®

Get Node.js® v24.22.0 LTS for Windows using Docker with npm

Or get a prebuilt Node.js® for Windows running a x64 architecture.

Windows Installer (.msi)

Standalone Binary (.zip)



```
C:\examples>node
```

```
Welcome to Node.js v24.22.0.
Type ".help" for more information.
```

```
> 2+2
```

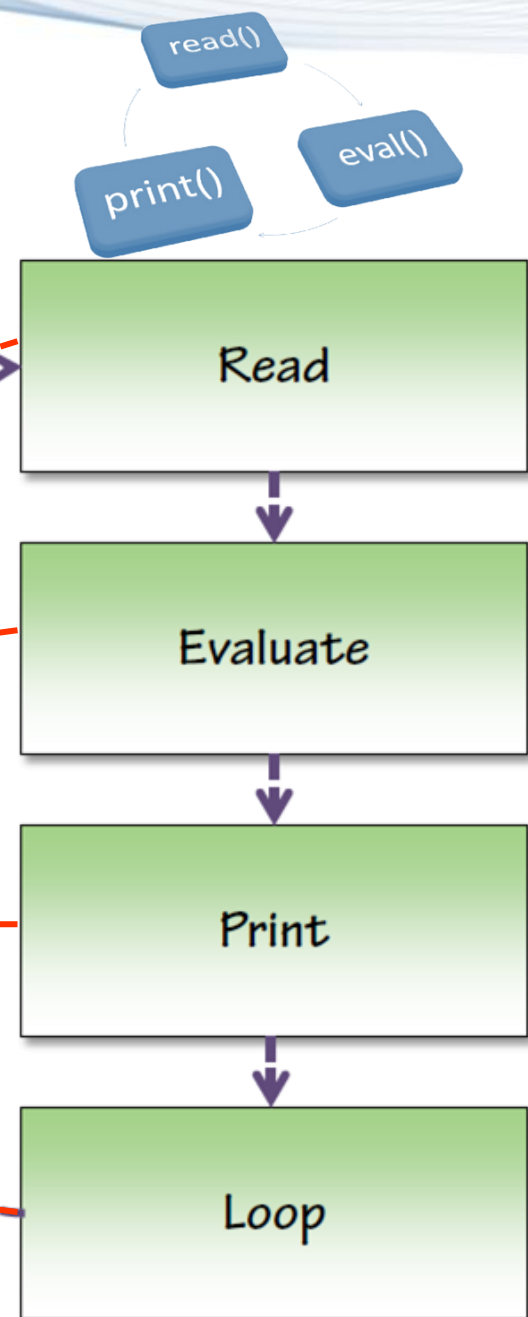
```
4
```

```
>
```

REPL (Read-Evaluate-Print-Loop)

- Η συμπεριφορά είναι ίδια με την πραγματική ζωή...

```
C:\examples>node
Welcome to Node.js v24.22.0.
Type ".help" for more information.
> 2+2
4
>
```



Μερικά πράγματα που μπορούμε να κάνουμε...

```
C:\examples>node
Welcome to Node.js v24.22.0.
Type ".help" for more information.
> 2+2
4
> 2+3
5
> 2/3
0.6666666666666666
> 2-3
-1
> 3>5
false
> 3<5
true
> 3 == 5
false
> 3 === 5
false
> '5' == 5
true
> '5' === 5
false
>
```

Μπορούμε να κάνουμε πράξεις και συγκρίσεις.

Για να ελέγξουμε αν δύο τιμές είναι ίσες χρησιμοποιούμε το `===` (αυστηρή ισότητα). Το `==` προσπαθεί πρώτα να «εξομοιώσει» τους τύπους: γι' αυτό το `'5' == 5` δίνει `true`!

Θα δούμε γιατί στους τύπους δεδομένων.

...και που δε μπορούμε...

```
> 3>5
false
> 3<5
true
> lunch
Uncaught ReferenceError: lunch is not defined
>
```

```
> {
|  asdasd
|  123123
|  .break
>
```

Το | σημαίνει ότι η Node περιμένει να ολοκληρωθεί η εντολή. Με **.break** βγαίνουμε.

```
> .help
.break    Sometimes you get stuck, this gets you out
.clear    Alias for .break
.editor   Enter editor mode
.exit     Exit the REPL
.help     Print this help message
.load     Load JS from a file into the REPL session
.save     Save all evaluated commands in this REPL session to a file

Press Ctrl+C to abort current expression, Ctrl+D to exit the REPL
>
```

Ας κάνουμε ένα βήμα ακόμα...

```
> let lunch = 'pizza'
undefined
> lunch
'pizza'
> lunch = 'πίτα με γύρο'
'πίτα με γύρο'
> luch
Uncaught ReferenceError: luch is not defined
> 'abc'
'abc'
> "abc"
'abc'
> 'hello world'
'hello world'
>
```

assign
αναθέτω

← η δήλωση με let δεν «επιστρέφει» τιμή

ορθογραφικό λάθος (luch)!

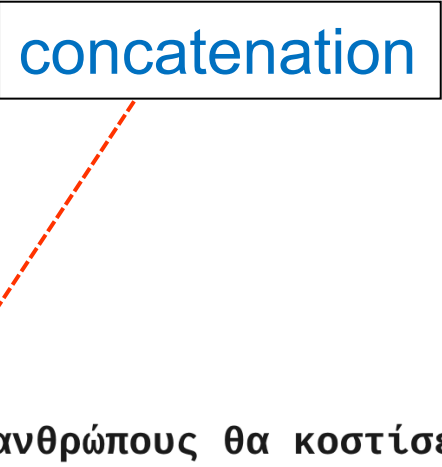
string



Και ας προϋπολογίσουμε ένα βράδυ μας...

```
> let lunch = 'πίτσα με 12 κομμάτια'
undefined
> let sizeOfParty = 4
undefined
> let totalCost = 14.99
undefined
> totalCost / sizeOfParty
3.7475
> let averageCost = totalCost / sizeOfParty
undefined
> averageCost
3.7475
> 'Το γεύμα με ' + lunch + ' για ' + sizeOfParty + ' ανθρώπους θα κοστίσει ' + averageCost
'Tο γεύμα με πίτσα με 12 κομμάτια για 4 ανθρώπους θα κοστίσει 3.7475'
>
```

concatenation



Μη επιτρεπτό
average cost

Επιτρεπτό
averageCost
lunchOrder
averageBalance

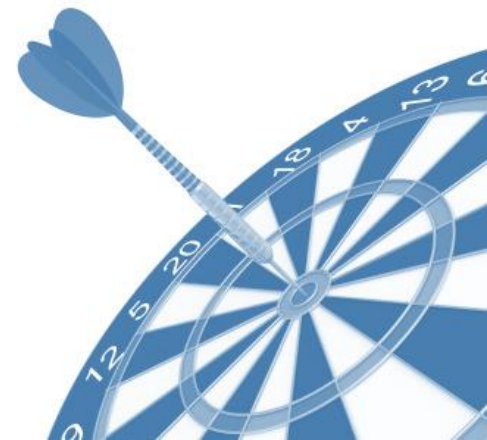
Προσοχή!
Η JavaScript είναι **Case Sensitive**.
Προσοχή επίσης στις **συμβάσεις!**

Τρεις μικρές συμβουλές...

- Δεν υπάρχει δυνατότητα να **μάθετε** προγραμματισμό απλά **διαβάζοντας**.

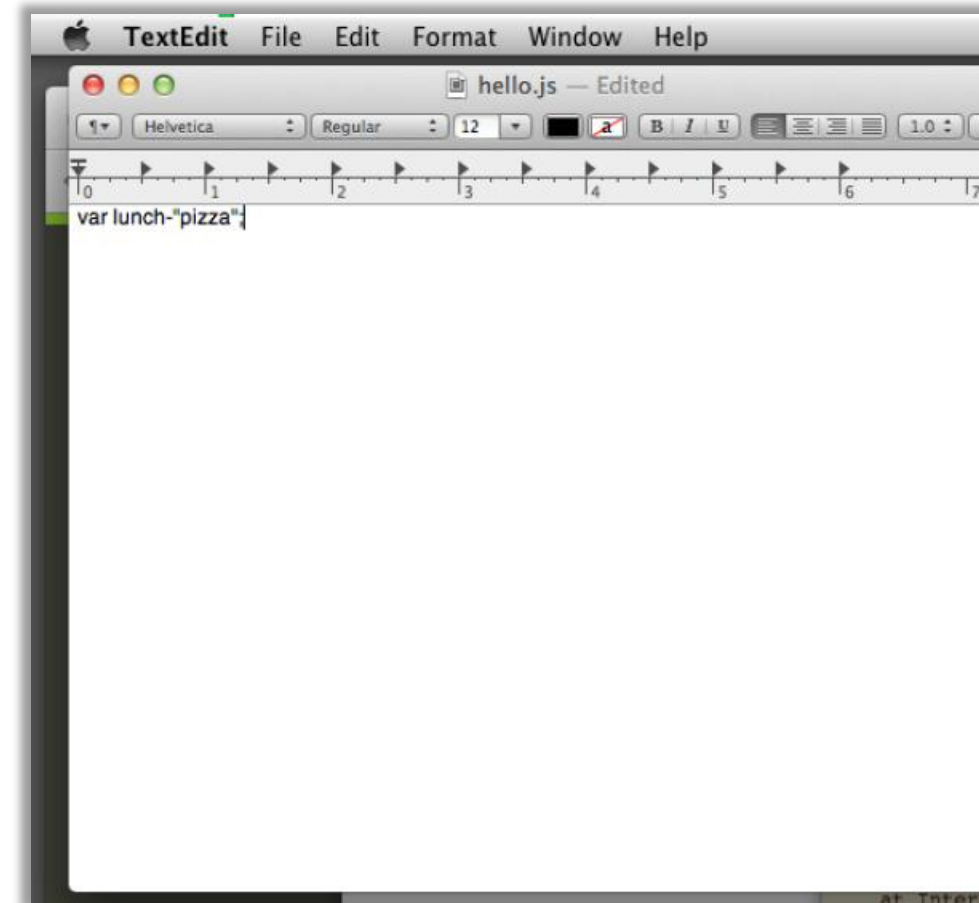
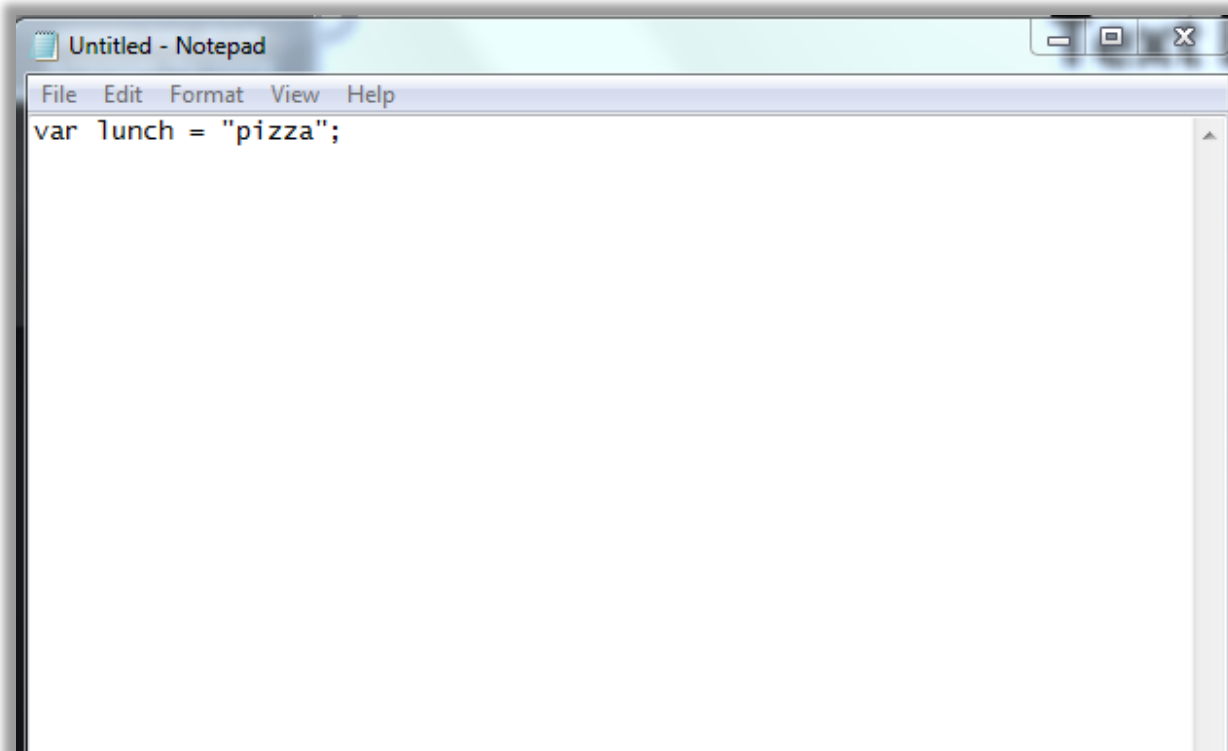
Πάντα **δοκιμάστε μόνοι σας** όσα βλέπουμε μαζί.

- Πρέπει να είστε πάντα **ακριβείς**.
Κάνετε διπλούς ελέγχους σε όσα πληκτρολογείτε.
- **Πειραματιστείτε!** Η εμπειρία θα σας κάνει καλύτερους προγραμματιστές!



Και πού γράφουμε κώδικα;

- Text Editors
 - Notepad (Windows),
 - TextEdit (Mac),
 - αλλά υπάρχουν και πολύ καλύτερες λύσεις...



Δηλαδή...

Administrator: Command Prompt

Microsoft Windows [Version 10.0.17134.1069]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>notepad

C:\Users\Administrator>cd documents

C:\Users\Administrator\Documents>mkdir examples

C:\Users\Administrator\Documents>cd examples

C:\Users\Administrator\Documents\examples>dir
Volume in drive C has no label.
Volume Serial Number is F4AF-F997

Directory of C:\Users\Administrator\Documents\examples

08/11/2019	11:11 πμ	<DIR>	.
08/11/2019	11:11 πμ	<DIR>	..
		0 File(s)	0 bytes
		2 Dir(s)	171.558.150.144 bytes free

C:\Users\Administrator\Documents\examples>

C:\Users\Administrator\Documents\examples>node program.js
pizza

C:\Users\Administrator\Documents\examples>

Untitled - Notepad

File Edit Format View Help

```
lunch = "pizza"  
console.log(lunch)
```

Save As



<< Documents >> My Documents >> examples

Search examples

Organize

New folder

★ Favorites

Desktop

Downloads

Recent Places

Libraries

Documents

Music

Pictures

Videos

Documents library

examples

Arrange by: Folder

Name

Date modified

Type

No items match your search.

File name: program.js

Save as type: All Files (*.*)

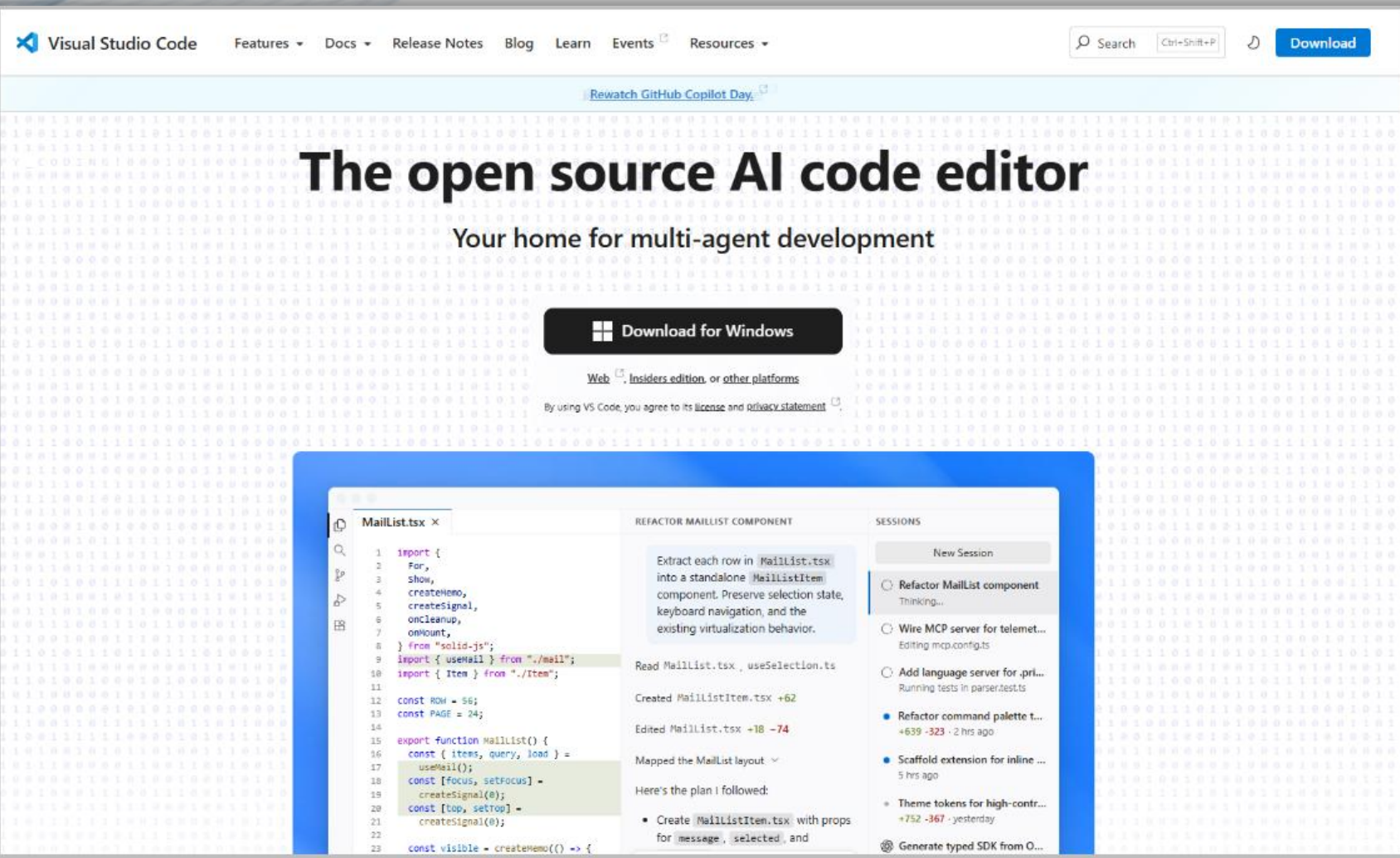
Hide Folders

Encoding: ANSI

Save

Cancel

Πιο έξυπνες λύσεις....



- **Visual Studio Code:** δωρεάν και ένα από τα πιο δημοφιλή εργαλεία σήμερα.
- Χρωματισμός και αυτόματη συμπλήρωση κώδικα, εντοπισμός λαθών, ενσωματωμένο terminal για να τρέχουμε το **node**.
- Έχει και βοηθούς **AI**: χρησιμοποιήστε τους για να **μάθετε** και όχι για να αποφύγετε να σκεφτείτε!
- <https://code.visualstudio.com/>

Πιο έξυπνες λύσεις....



Current Version 8.9.8.1

Home

Download

News

Online Help

Resources

RSS

Donate

Author

What is Notepad++

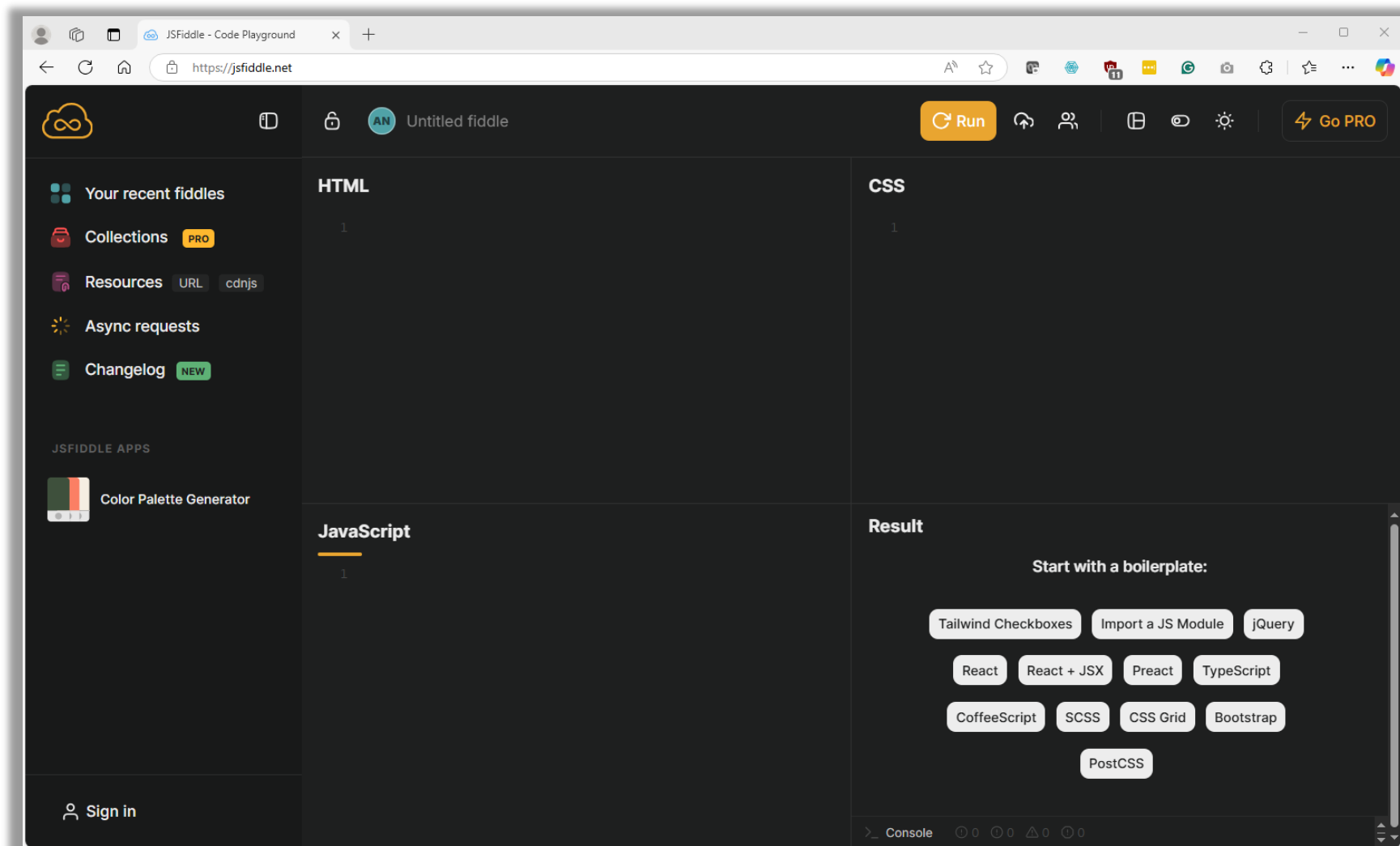
Notepad++ is a free (as in “free speech” and also as in “free beer”) source code editor and Notepad replacement that supports several programming languages. Running in the MS Windows environment, its use is governed by [GNU General Public License](#).

Based on the powerful editing component [Scintilla](#), Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment.

```
*C:\sources\notepad4ever.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X
Notepad_plus.cpp notepad4ever.cpp new 1
1 #include <GPL>
2 #include <free_software>
3
4 void Notepad4ever()
5 {
6     while (true)
7     {
8         Notepad++ ;
9     }
10 }
length: 108 line:Ln: 8 Col: 21 Pos: 102 Windows (CR LF) UTF-8 INS
```

Πιο έξυπνες λύσεις.... (online)

- Υπάρχουν **πολλές ακόμα** online (π.χ. JSFiddle) και κάθε browser έχει ενσωματωμένη **κονσόλα JavaScript** (F12 → Console)!



Syntax Rules – Συντακτικοί Κανόνες

- Κάθε γλώσσα προγραμματισμού έχει ένα καθορισμένο **συντακτικό** (**syntax**).
- Σε **διαφορετικές γλώσσες** το συντακτικό μπορεί να **διαφέρει αρκετά**.
- Το συντακτικό έχει σχέση με το πώς **διαμορφώνεις** τον κώδικα, ώστε η γλώσσα να μπορεί να τον καταλάβει.

```
let lunch = "pizza"
```

```
console.log(lunch)  
console.log(lunch)  
console.log(4)
```

```
let lunch = "pizza";
```

```
console.log(lunch);  
console.log(lunch);  
console.log(4);
```

```
let name = "John";  
name = "Alex";
```

Keyword: πρόκειται για μία λέξη με ξεχωριστή σημασία για τη γλώσσα.

Βέλτιστη πρακτική: δηλώνουμε **πάντα** κάθε μεταβλητή με **let** (ή **const**). Χωρίς δήλωση η JavaScript δημιουργεί «σιωπηλά» μια καθολική μεταβλητή – πηγή λαθών.

Προαιρετικό επίσης...

Ευκολία στην ανάγνωση...

```
let name = "John";  
name = "Alex";
```

Semicolon (;)

- Καλό θα είναι να **χρησιμοποιείται**, γιατί η μη χρήση της μοιάζει με χρήση της ακόλουθης πρότασης:
- *Η Μαρία χώρισε με το αγόρι της Ο Γιώργος δεν είχε ιδέα ότι ήταν στενοχωρημένη*

αντί για...

- *Η Μαρία χώρισε με το αγόρι της. Ο Γιώργος δεν είχε ιδέα ότι ήταν στενοχωρημένη*

```
let lunch = "pizza"; console.log(lunch);
```

Αν πρόκειται να κάνετε λάθος, καλύτερα να γίνει νωρίτερα!

- Η **εμπειρία** που θα αποκτήσετε στη διάγνωση λαθών είναι **ανεκτίμητη**, ιδιαίτερα όταν οι γραμμές του κώδικα γίνουν πολλές!

```
let lunch = "pizza" console.log(lunch);
```

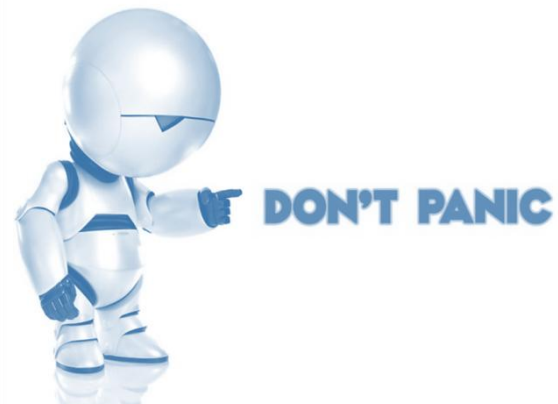
```
let lunch = "pizza";  
console.log(lunch);  
console.log(lunch);  
console.log(4);
```

```
C:\examples>node syntax_error.js  
C:\examples\syntax_error.js:1  
let lunch = "pizza" console.log(lunch);  
                      ^^^^^^^
```

```
SyntaxError: Unexpected identifier 'console'  
    at wrapSafe (node:internal/modules/cjs/loader:1861:18)  
    at Module._compile (node:internal/modules/cjs/loader:1903:20)  
    at Object..js (node:internal/modules/cjs/loader:2060:10)  
    at Module.load (node:internal/modules/cjs/loader:1651:32)  
    at Module._load (node:internal/modules/cjs/loader:1443:12)  
    at wrapModuleLoad (node:internal/modules/cjs/loader:261:19)  
    at Module.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:154:5)  
    at node:internal/main/run_main_module:33:47
```

Node.js v24.22.0

```
C:\examples>node typo.js  
C:\examples\typo.js:3  
console.log(lunch);  
          ^^
```



DON'T PANIC



Case Sensitive

- Στη JavaScript έχει σημασία η χρήση μικρών και κεφαλαίων γραμμάτων!

```
let Lunch = "pizza";
```

```
console.log(lunch);  
console.log(lunch);  
console.log(4);
```

```
let name = "John";  
name = "Alex";  
console.log(name);
```

Challenge Round:

Προπαθήστε να δημιουργήσετε λάθη στο πρόγραμμα!

```
C:\examples>node case_sensitive.js  
C:\examples\case_sensitive.js:3  
console.log(lunch);  
          ^
```

ReferenceError: lunch is not defined

```
at Object.<anonymous> (C:\examples\case_sensitive.js:3:13)  
at Module._compile (node:internal/modules/cjs/loader:1929:14)  
at Object..js (node:internal/modules/cjs/loader:2060:10)  
at Module.load (node:internal/modules/cjs/loader:1651:32)  
at Module._load (node:internal/modules/cjs/loader:1443:12)  
at wrapModuleLoad (node:internal/modules/cjs/loader:261:19)  
at Module.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:154:5)  
at node:internal/main/run_main_module:33:47
```

Node.js v24.22.0

Ας προϋπολογίσουμε ξανά το βράδυ μας...

- `>process.`
(και πατήστε το πλήκτρο **Tab**)

Το **στιγμιότυπο** ενός προγράμματος που τρέχει.

```
> process.  
process.abort  
process.allowedNodeEnvironmentFlags  
process.arch  
process.argv  
process.argv0  
process.availableMemory  
process.binding  
process.chdir  
process.config  
process.constrainedMemory  
process.cpuUsage  
process.cwd  
process.debugPort  
process.dlopen  
process.domain  
process.emitWarning  
process.env  
process.execArgv  
process.execPath  
  
process.execve  
process.exit  
process.exitCode  
process.features  
process.finalization  
process.getActiveResourcesInfo  
process.getBuiltinModule  
process.hasUncaughtExceptionCaptureCallback  
process.hrtime  
process.kill  
process.loadEnvFile  
process.memoryUsage  
process.moduleLoadList  
process.nextTick  
process.openStdin  
process.pid  
process.platform  
process.ppid  
process.reallyExit  
  
process.ref  
process.release  
process.report  
process.resourceUsage  
process.setSourceMapsEnabled  
process.setUncaughtExceptionCaptureCallback  
process.sourceMapsEnabled  
process.stderr  
process.stdin  
process.stdout  
process.threadCpuUsage  
process.title  
process.umask  
process.unref  
process.uptime  
process.version  
process.versions  
  
> process.version  
'v24.22.0'  
> process.platform  
'win32'  
> process.argv  
[ 'C:\\Program Files\\nodejs\\node.exe' ]  
>
```

Ας πάρουμε είσοδο από το χρήστη...

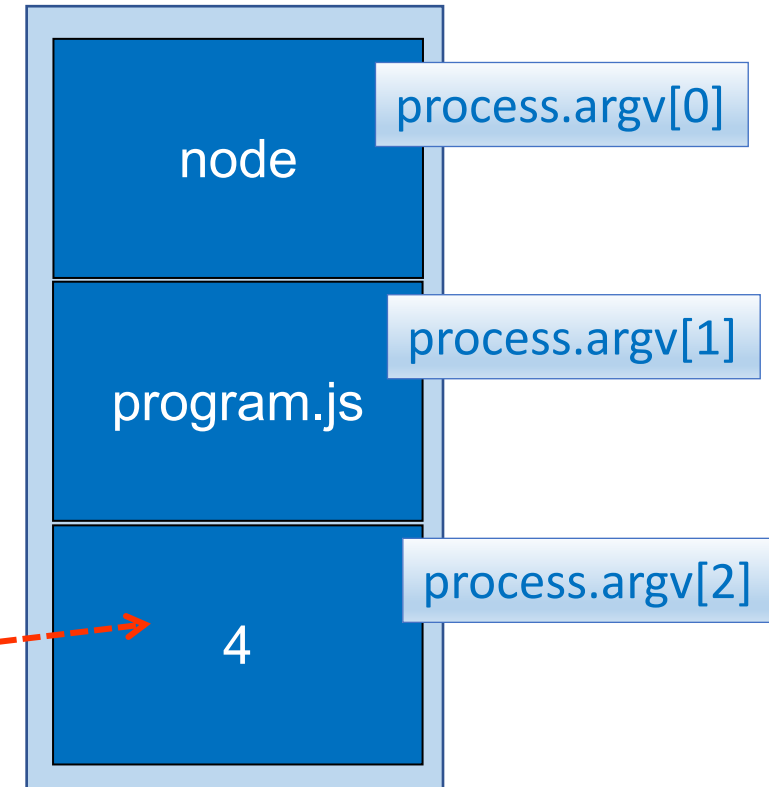
```
1 const totalCost = 14.99;  
2  
3 console.log(process.argv);
```

```
C:\examples>node argv.js pizza orange onion  
[  
  'C:\\Program Files\\nodejs\\node.exe',  
  'C:\\examples\\argv.js',  
  'pizza',  
  'orange',  
  'onion'  
]
```

```
C:\examples>node argv.js 4  
[  
  'C:\\Program Files\\nodejs\\node.exe',  
  'C:\\examples\\argv.js',  
  '4'  
]
```

```
1 const totalCost = 14.99;  
2  
3 const sizeOfParty = process.argv[2];
```

To process.argv είναι
ένα array



Και σας δούμε το αποτέλεσμα...

```
1  const totalCost = 14.99;  
2  
3  const sizeOfParty = process.argv[2];  
4  
5  const averageCost = totalCost / sizeOfParty;  
6  
7  console.log("$" + averageCost);
```



```
C:\examples>node program.js 4  
$3.7475
```

```
C:\examples>node program.js 3  
$4.996666666666667
```

```
C:\examples>node program.js 36  
$0.416388888888889
```

```
C:\examples>node program.js 3610  
$0.004152354570637119
```

Σκεφτείτε πριν τρέξετε! – Είσοδος από τον χρήστη

```
1  const totalCost = 14.99;  
2  
3  const sizeOfParty = process.argv[2];  
4  
5  const averageCost = totalCost / sizeOfParty;  
6  
7  console.log("$" + averageCost);
```

```
C:\examples>node program.js  
$NaN
```

```
C:\examples>node program.js 0  
$Infinity
```

```
C:\examples>node program.js four  
$NaN
```

Τι θα τυπώσει το πρόγραμμα αν το τρέξουμε:

α) χωρίς είσοδο: `node program.js`

β) `node program.js 0`

γ) `node program.js four`

α) **\$NaN**: χωρίς είσοδο το `process.argv[2]` είναι `undefined` – και `14.99 / undefined` δεν είναι αριθμός.

β) **\$Infinity**: διαίρεση με το 0, ακριβώς όπως στο Scratch (Διάλεξη 2)!

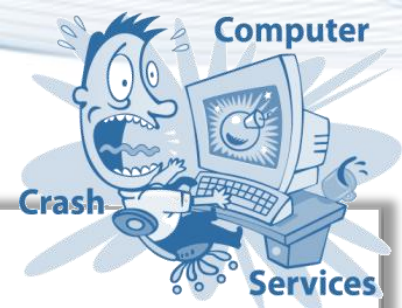
γ) **\$NaN**: το «four» δεν μετατρέπεται σε αριθμό.

Ο υπολογιστής κάνει ό,τι του γράψουμε: πρέπει να ελέγχουμε την είσοδο.

Μεταβλητές & Τύποι δεδομένων

- Σχεδόν όλες οι γλώσσες προγραμματισμού υποστηρίζουν **μεταβλητές (variables)** κάποιου είδους.
 - Είναι ιδιαίτερα **χρήσιμες**.
 - Μας επιτρέπουν να **αποθηκεύουμε δεδομένα** σε κάποιο πρόγραμμα και να τα ανακτούμε μετέπειτα σε κάποιο άλλο χρονικό σημείο, όταν τα χρειαστούμε.
 - Μπορείτε να τις σκέφτεστε σαν τις **μεταβλητές στα μαθηματικά**, αλλά αν δε σας αρέσουν τα μαθηματικά μην ανησυχείτε, εδώ δε θα τα χρησιμοποιήσουμε. Απλά θα δούμε πώς λειτουργούν οι μεταβλητές.
- Επίσης, θα δούμε τους **τύπους δεδομένων (Data Types)** στη JavaScript.

Οι μεταβλητές πιο αναλυτικά



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



```
C:\examples>node undeclared.js
C:\examples\undeclared.js:1
console.log(x);
      ^
```

```
ReferenceError: x is not defined
    at Object.<anonymous> (C:\examples\undeclared.js:1:13)
    at Module._compile (node:internal/modules/cjs/loader:1929:14)
    at Object..js (node:internal/modules/cjs/loader:2060:10)
    at Module.load (node:internal/modules/cjs/loader:1651:32)
    at Module._load (node:internal/modules/cjs/loader:1443:12)
    at wrapModuleLoad (node:internal/modules/cjs/loader:261:19)
    at Module.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:154:5)
    at node:internal/main/run_main_module:33:47
```

```
Node.js v24.22.0
```

```
1 let x;
2
3 console.log(x);
```



```
C:\examples>node uninitialised.js
undefined
```

- Οι μεταβλητές πρέπει να έχουν μία τιμή. Θα πρέπει να τις **αρχικοποιήσετε** (θυμηθείτε το Scratch, Διάλεξη 2).
- Αποθηκεύουν τα δεδομένα στη **μνήμη** μέχρι το πρόγραμμα να τερματιστεί.

```
1 let x;
2
3 x = 3;
4 console.log(x);
```

Δήλωση Μεταβλητών – Συμβάσεις

```
1  let x;  
2  
3  x = 3;  
4  console.log(x);  
5  console.log(x);  
6  
7  x = 5;  
8  console.log(x);  
9  
10 let y = 2;  
11  
12 console.log(y);
```



```
1  let x;  
2  let y = 2;  
3  
4  x = 3;  
5  console.log(x);  
6  console.log(x);  
7  
8  x = 5;  
9  console.log(x);  
10  
11 console.log(y);
```



```
1  let x, y = 2;  
2  
3  x = 3;  
4  console.log(x);  
5  console.log(x);  
6  
7  x = 5;  
8  console.log(x);  
9  
10 console.log(y);
```

- Το αποτέλεσμα είναι το ίδιο αλλά είναι πολύ σημαντικό να ακολουθούμε **συμβάσεις** για να είναι πιο **ευανάγνωστος (readable)** ο κώδικάς μας.



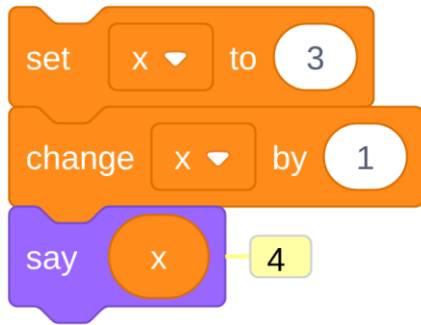
```
let x = 3,  
    y = 2;
```



```
let x = 3,    y = 2;
```

Μεταβλητές σε τρεις γλώσσες

Scratch



Δημιουργία με το κουμπί **Make a Variable**: όνομα και επιλογή «για όλα τα sprites» ή «για ένα».

JavaScript

```
let x = 3;  
x = x + 1;    // ή: x += 1  
console.log(x); // 4
```

Δήλωση με **let** (ή **const**). Το «=» σημαίνει **ανάθεση**, όχι ισότητα.

Python

```
x = 3  
x = x + 1    # ή: x += 1  
print(x)     # 4
```

Η μεταβλητή δημιουργείται με την **πρώτη ανάθεση** – δεν υπάρχει λέξη-κλειδί.

Ίδια ιδέα παντού:

ένα **όνομα** που κρατάει μια **τιμή** στη μνήμη και η τιμή μπορεί να **αλλάξει**.

let ή const; (και το παλιό var)

- **let**: για τιμές που θα **αλλάξουν** (π.χ. ένας μετρητής).
- **const**: για τιμές που **δεν αλλάζουν** αφού τις ορίσουμε (π.χ. `totalCost = 14.99`). Αν προσπαθήσουμε να τις αλλάξουμε, παίρνουμε **σφάλμα**.
- **var**: ο παλιός τρόπος (πριν το 2015). Θα τον βρείτε σε παλιό κώδικα και παραδείγματα στο Διαδίκτυο. Εμείς δεν τον χρησιμοποιούμε.
- **Πρακτική**: ξεκινάμε με **const** και αλλάζουμε σε **let** μόνο αν χρειαστεί.

```
1  const totalCost = 14.99;  
2  let  sizeOfParty = 4;  
3  
4  sizeOfParty = 5;           // OK: let  
5  console.log(sizeOfParty);  
6  
7  totalCost = 20;           // Λάθος: const
```

```
C:\examples>node const.js  
5  
C:\examples\const.js:7  
totalCost = 20;           // Λάθος: const  
      ^  
  
TypeError: Assignment to constant variable.  
    at Object.<anonymous> (C:\examples\const.js:7:11)  
    at Module._compile (node:internal/modules/cjs/loader:1929:14)  
    at Object..js (node:internal/modules/cjs/loader:2060:10)  
    at Module.load (node:internal/modules/cjs/loader:1651:32)  
    at Module._load (node:internal/modules/cjs/loader:1443:12)  
    at wrapModuleLoad (node:internal/modules/cjs/loader:261:19)  
    at Module.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:154:5)  
    at node:internal/main/run_main_module:33:47  
  
Node.js v24.22.0
```

Τύποι Δεδομένων – Data Types

- Οι **τιμές** που αναθέτουμε σε κάποια μεταβλητή έχουν ένα **τύπο, τύπο δεδομένων**.
- Ένας τύπος δεδομένων είναι στην πραγματικότητα η **ταξινόμηση** μίας **τιμής**.
- Στις **διάφορες γλώσσες** προγραμματισμού υπάρχουν **διάφοροι τύποι** δεδομένων.

```
1  let x = 3,  
2      y = 2.5;  
3  let z = '3';  
4  let result = x > y;  
5  
6  console.log(typeof x);  
7  console.log(typeof y);  
8  console.log(typeof z);  
9  console.log(result);  
10 console.log(typeof result);  
11 console.log(typeof console);
```

```
C:\examples>node types.js  
number  
number  
string  
true  
boolean  
object
```

Προσοχή όμως στο ακόλουθο:

```
1  let x = 3,  
2      y = 2.5;  
3  let z = '3';  
4  let result = x > y;  
5  
6  console.log(typeof x);  
7  console.log(typeof y);  
8  console.log(typeof z);  
9  
10 y = "Είμαι αριθμός;";  
11 console.log(typeof y);  
12  
13 console.log(result);  
14 console.log(typeof result);  
15 console.log(typeof console);
```

- Προσοχή!

Οι τύποι δεδομένων είναι τόσο σημαντικοί γιατί ο τύπος από κάτι συχνά συμβάλλει στην απόφαση του **τι** μπορείτε να κάνετε με αυτό και πώς θα **συμπεριφερθεί**!

```
C:\examples>node types.js  
number  
number  
string  
string  
true  
boolean  
object
```

Στην πράξη η JavaScript βοηθάει... αλλά...

```
1 let x = 3,  
2   y = 2.5;  
3 let z = '3';  
4 let result = x > y;  
5  
6 let answer = x + y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

C:\examples>node types.js
5.5
number

```
1 let x = 3,  
2   y = 2.5;  
3 let z = '3';  
4 let result = x > y;  
5  
6 let answer = x + z;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

C:\examples>node types.js
33
string

- Η JavaScript προσπαθεί να σας βοηθήσει κάνοντας υποθέσεις, αλλά...

Στην πράξη η JavaScript βοηθάει... αλλά...

```
1 let x = 3,  
2   y = 2.5;  
3 let z = '3';  
4 let result = x > y;  
5  
6 let answer = x * y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

C:\examples>node types.js
7.5
number

```
1 let x = 3,  
2   y = 2.5;  
3 let z = '3';  
4 let result = x > y;  
5  
6 let answer = z * y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

C:\examples>node types.js
7.5
number

- Η JavaScript προσπαθεί να σας βοηθήσει κάνοντας υποθέσεις, αλλά...

Κάνετε μερικές δοκιμές...

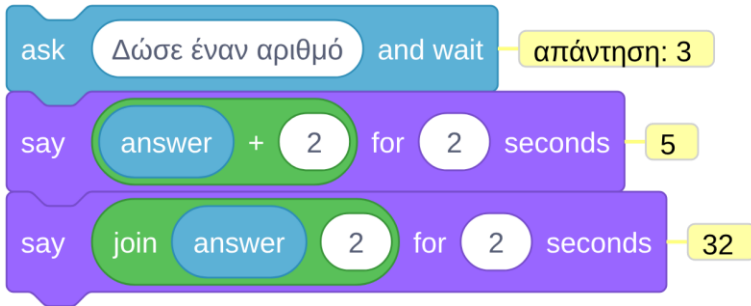
- Προσθέστε το 'Hello' με το 'Hello'
 - Πολλαπλασιάστε το '2.5' με το '3.5'
 - Πολλαπλασιάστε το 'Hello' με το 'Hello'
- Αν δείτε “NaN” σημαίνει “not a number”
(με άλλα λόγια δεν είναι εφικτός ο υπολογισμός).

```
> 'Hello' + 'Hello'  
'HelloHello'  
> '2.5' * '3.5'  
8.75  
> 'Hello' * 'Hello'  
NaN  
>
```



Το '3' + 2 σε τρεις γλώσσες

Scratch



Στο Scratch το + κάνει **πάντα** πρόσθεση (5).

Για κείμενα υπάρχει ξεχωριστό block: το **join** (32).

JavaScript

```
1 console.log('3' + 2); // σύνδεση κειμένων
2 console.log('3' * 2); // μετατροπή σε αριθμό
3 console.log(Number('3') + 2); // μετατροπή από εμάς
4 console.log('Hello' * 'Hello');
```

```
C:\examples>node coercion.js
32
6
5
NaN
```

Η JavaScript **μαντεύει**: με + ενώνει κείμενα, με * κάνει αριθμούς.

Python

```
1 print(int('3') + 2) # μετατροπή από εμάς
2 print('3' * 2)     # επανάληψη κειμένου!
3 print('3' + 2)     # σφάλμα τύπου
```

```
C:\examples>python coercion.py
5
33
Traceback (most recent call last):
  File "C:\examples\coercion.py", line 3, in <module>
    print('3' + 2)           # σφάλμα τύπου
    ~~~~^~~~
TypeError: can only concatenate str (not "int") to str
```

Η Python **σταματά** με σφάλμα: πρέπει να μετατρέψουμε εμείς (int).

[Demos_2026-2027/coercion.js](#), [coercion.py](#)

JavaScript: «χαλαρή» (weak) στους τύπους – Python: «αυστηρή» (strong).

Θυμηθείτε τον άξονα **Strong – Weak** στη διαφάνεια 7!

Όταν δεν είμαστε σίγουροι, μετατρέπουμε **εμείς**: `Number(...)` / `int(...)`.

Αν θέλαμε όμως να πάρουμε τα πράγματα στα χέρια μας...

```
1 let x = 3,  
2   y = 2.5;  
3 let z = '3';  
4 let result = x > y;  
5  
6 let answer = z + y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

```
C:\examples>node types.js  
32.5  
string
```

«3» + 2.5 → κείμενο!

Αν θέλουμε η απάντηση να είναι αριθμός....

```
1 let x = 3,  
2   y = 2.5;  
3 let z = '3';  
4 let result = x > y;  
5  
6 let answer = parseInt(z) + y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

(ή `Number(z)`, που μετατρέπει και δεκαδικούς)

Μπορούμε να ----->
το κάνουμε!

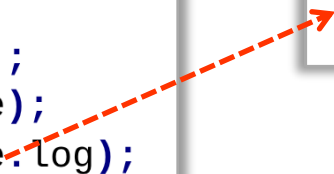
```
C:\examples>node types.js  
5.5  
number
```

Και μία μικρή εισαγωγή στις συναρτήσεις πριν τις δούμε αναλυτικά...

- Ας δούμε έναν ακόμα ειδικό τύπο στη JavaScript.

```
1 let x = 3,  
2     y = 2.5;  
3 let z = '3';  
4 let result = x > y;  
5  
6 let answer = parseInt(z) + y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);  
10  
11 console.log(typeof y);  
12 console.log(typeof z);  
13 console.log(typeof result);  
14 console.log(typeof console);  
15 console.log(typeof console.log);
```

```
C:\examples>node types.js  
5.5  
number  
number  
string  
boolean  
object  
function
```



- Οι **συναρτήσεις (functions)** στη JavaScript είναι ιδιαίτερα **ισχυρές** γιατί περιέχουν κώδικα που μπορείτε να καλέσετε εντός του προγράμματός σας – αυτό κάναμε και με την **console.log**.

Συναρτήσεις – Functions

```
1 function writeGreeting() {  
2     console.log("Hello!");  
3     console.log("Hello!");  
4 }  
5  
6 writeGreeting();
```

Κώδικας

Κλήση

```
C:\examples>node greeting.js  
Hello!  
Hello!
```

```
1 function writeGreeting() {  
2     let message = "Hello!";  
3     console.log(message);  
4     console.log(message);  
5 }  
6  
7 writeGreeting();
```

Στις συναρτήσεις μπορούμε να ορίσουμε μεταβλητές

```
1 function writeGreeting(name) {  
2     let message = "Hello! " + name;  
3     console.log(message);  
4 }  
5  
6 writeGreeting("John");  
7 writeGreeting("Mary");
```

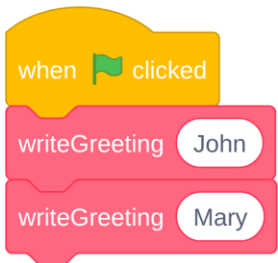
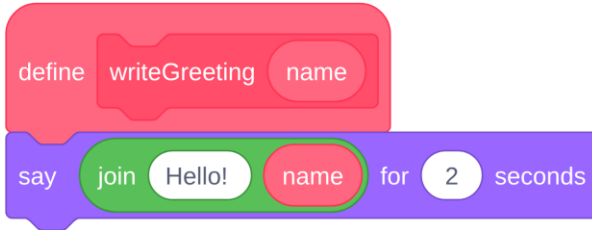
Στις συναρτήσεις μπορούμε να περάσουμε παραμέτρους

```
C:\examples>node greeting.js  
Hello! John  
Hello! Mary
```

- **Κάντε μια μικρή δοκιμή.** Τι συμβαίνει αν χρησιμοποιήσετε παρενθέσεις δίπλα σε μία μεταβλητή που δεν είναι συνάρτηση; π.χ. x();

Συναρτήσεις σε τρεις γλώσσες

Scratch



My Blocks → Make a Block

JavaScript

```
1 function writeGreeting(name) {  
2     let message = "Hello! " + name;  
3     console.log(message);  
4 }  
5  
6 writeGreeting("John");  
7 writeGreeting("Mary");
```

Python

```
1 def write_greeting(name):  
2     message = "Hello! " + name  
3     print(message)  
4  
5 write_greeting("John")  
6 write_greeting("Mary")
```

```
C:\examples>node greeting.js  
Hello! John  
Hello! Mary
```

```
C:\examples>python greeting.py  
Hello! John  
Hello! Mary
```

[Demos_2026-2027/greeting.js, greeting.py](#)

Ίδια ιδέα:

δίνουμε **όνομα** σε ένα κομμάτι κώδικα, το **καλούμε** όποτε θέλουμε και του περνάμε **παραμέτρους**.
Αναλυτικά στην επόμενη διάλεξη.

Τι είδαμε μέχρι τώρα;

- Κάναμε μία **εισαγωγή** στη JavaScript.
- Είδαμε **εργαλεία** (Node.js, REPL, VS Code) με τα οποία μπορούμε να **γράψουμε** κώδικα σε JavaScript και πώς να τον **τρέξουμε**.
- Ασχοληθήκαμε με κάποια εισαγωγικά σημεία σε σχέση με το **συντακτικό** της γλώσσας.
- Μιλήσαμε για τις **μεταβλητές** (let, const) και τους **τύπους δεδομένων** της JavaScript – και πώς τους «μετατρέπει» μόνη της – και τέλος...
- Είδαμε πολύ επιφανειακά τις **συναρτήσεις** με τις οποίες θα ασχοληθούμε αναλυτικά στο επόμενο μάθημα.
- Οδηγός αναφοράς για τη JavaScript: developer.mozilla.org (MDN).



