

JAVASCRIPT



Νίκος Παπακωνσταντόπουλος
Φιλοθέη Λιναρδάτου

Τι είναι η Javascript (JS)

- Γλώσσα σεναρίου και επέκτασης
- Αντικειμενοστρεφής γλώσσα προγραμματισμού, έχει και στοιχεία από συναρτησιακό προγραμματισμό και διαχείριση συμβάντων
- Διερμηνευόμενη
- Δημοφιλής:

<https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language>

- Χρησιμοποιείται στην ανάπτυξη ιστοσελίδων
- αναλαμβάνει το λειτουργικό κομμάτι της ιστοσελίδας συμβάλλοντας στην αλληλεπίδρασή της με τον χρήστη
- Ενσωματώνεται σε στατικές σελίδες HTML
- Είναι client-side γλώσσα, δηλαδή δεν τρέχει στον server, αλλά στον browser του client.

Data Types

1. Strings ('Hello world')
2. Boolean(true , false)
3. Number(1, 12.5 , NaN,Infinity, Number.MAX_VALUE,Number.MIN_VALUE)
4. Undefined (όταν δηλώνουμε μία μεταβλητή αλλά δεν την αρχικοποιούμε)
5. Null(όταν αρχικοποιούμε μία μεταβλητή αλλά δε θέλουμε να τις δώσουμε μία συγκεκριμένη τιμή)
6. Objects
7. Arrays

Variables

- Η JS είναι dynamically typed γλώσσα, δηλαδή μία μεταβλητή μπορεί να αναπαραστήσει οποιονδήποτε τύπο. Γι' αυτό δεν χρειάζεται να δηλώσουμε κάποιον τύπο όταν την ορίζουμε.
- `let x=8; x=true;`
- `var a;`
- `const z=' Hello world';`
- `typeof (x);`
- <https://www.freecodecamp.org/news/var-let-and-const-whats-the-difference/>

If statements

```
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 the condition1 is false and  
    condition2 is false  
}
```

Conditions

Κατά τα γνωστά... $a==b$, $a<b$, $a>b$

Επίσης,

Οποιοδήποτε (μη κενό) string έχει τιμή true

Οποιαδήποτε (μη μηδενική) αριθμητική έκφραση έχει τιμή true

For statement

- ```
for(let j=0; j<5; j++)
 {console.log('Hello world');}
```
- ```
const person={name:'John', lastname:'Smith'};  
var text="";  
for(let x in person)  
  text+=person[x]+" ";
```
- ```
let colors=' Green Yellow Blue Red'
var text='Some colors are :'
for(let x of colors)
 text+=colors[x]+' '
```

# String Methods

- `let text='Hello world';`
- `text.length;`
- `text.slice(start, end);`

Δημιουργεί ένα καινούριο string από τη θέση start έως τη θέση end-1

- `text.split(separator)` σπάει το string σε substrings ανάλογα με το separator και τα τοποθετεί σε πίνακα. Επιστρέφει τον πίνακα.



# String methods

- `text.trim()` αφαιρεί τα κενά από την αρχή και το τέλος του string
- `text.toUpperCase()`
- `text.toLowerCase()`
- `parseInt(text)`

Οι string methods αφήνουν ανεπηρέαστο το αρχικό string και επιστρέφουν ένα καινούριο με την ζητούμενη ιδιότητα!

# Array methods

- `let array=['HTML','CSS','Javascript']`
- `array.length`
- `array.sort()`
- `array.pop()`
- `array.push('Node.js')` επιστρέφει το νέο μήκος του πίνακα
- `array.splice(position,number of elements to be removed,elements to be added)`

Οι Array methods, δε δημιουργούν έναν καινούριο πίνακα, αλλά κάνουν τις αλλαγές στον αρχικό.

# Objects

- Τα αντικείμενα είναι μία συλλογή από ιδιότητες (properties)

```
let person={
 first name: 'John',
 last name:'Smith',
 age:25,
 fullname: function(){
 return this.firstname+" "+this.lastname;
 };
};
```

# Objects

- `person.firstname` ή `person['firstname']`
- `Object.values(person)` επιστρέφει πίνακα με στοιχεία τα values του person  
  
`John, Smith, 25`
- `JSON.stringify(person)` επιστρέφει string με τα properties και τα values του αντικειμένου  
  
`{"firstname": "John", "lastname": "Smith", "age": "25"}`
- `delete person.age`
- `person.nationality = 'English'`

# Functions

- `function hello (s)`  
`{return s;}`
- `let hello=function(s){return s;}`
- `let hello=(s)=>{return s;}`  
(arrow function)

Ψάξτε επίσης τα closures...

# Classes

- Syntax:

```
Class name{
 constructor(){...}
 method1(){...}
 method2(){...}
 variable1;
 variable2;
}
```

Αν κάποια μέθοδος είναι getter, δηλώνεται με το πρόθεμα get

# Classes

- Ένα αντικείμενο, δηλώνεται ως εξής:

`let myObject=new Classname( values for parameters of constructor)`

- `myObject.method1()`
- `myObject.variable1`

# Dates

- `let z=new Date();`

δημιουργεί ένα αντικείμενο με την τρέχουσα ημερομηνία

- `let z=new Date(year,month,day,hours,minutes,seconds,ms)`

η ημερομηνία καθορίζεται από τον χρήστη

- `let z=new Date(ms);`

η τρέχουσα ημερομηνία+ms(milleseconds)



# Dates

## Date Get Methods

| Method            | Description                                          |
|-------------------|------------------------------------------------------|
| getFullYear()     | Get <b>year</b> as a four digit number (yyyy)        |
| getMonth()        | Get <b>month</b> as a number (0-11)                  |
| getDate()         | Get <b>day</b> as a number (1-31)                    |
| getDay()          | Get <b>weekday</b> as a number (0-6)                 |
| getHours()        | Get <b>hour</b> (0-23)                               |
| getMinutes()      | Get <b>minute</b> (0-59)                             |
| getSeconds()      | Get <b>second</b> (0-59)                             |
| getMilliseconds() | Get <b>millisecond</b> (0-999)                       |
| getTime()         | Get <b>time</b> (milliseconds since January 1, 1970) |

# Προσθήκη JavaScript σε σελίδα HTML

internal  
JavaScript

```
<script>
 // κώδικας JavaScript
</script>
```

Ως κώδικας τυπικά τοποθετείται εντός της περιοχής head ενός εγγράφου, προσβάσιμος μόνο από αυτό.

external  
JavaScript

```
<script src="scriptfile.js" async></script>
```

**async** γνώρισμα για ασύγχρονη εκτέλεση πριν την ολοκλήρωση ανάλυσης HTML

**defer** μετά τη φόρτωση HTML ή άλλου κώδικα JS

inline  
JavaScript

```
<button onclick="createParagraph()">Click me!</button>
```

# async

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

Στην περίπτωση αυτή ο κώδικας JavaScript εκτελείται στην πρώτη ευκαιρία **μετά την ολοκλήρωση φόρτωσής του**, και πριν το γεγονός **load** του window. Δεν τηρείται σειρά και είναι πιθανόν να διακοπεί η δημιουργία του DOM.

HTML

δημιουργία DOM...

blocked

δημιουργία DOM...

JS

Φόρτωμα JavaScript

Εκτέλεση JavaScript

# defer

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

Στην περίπτωση αυτή ο κώδικας JavaScript εκτελείται μετά την ολοκλήρωση φορτώματος της HTML ή άλλου script αλλά πριν το γεγονός **DOMContentLoaded**.



## διαδοχικό φόρτωμα κώδικα JS

Αν κάποιος κώδικας JS πρέπει να περιμένει ή για άλλο τμήμα κώδικα ή για το φόρτωμα της HTML τότε μπορούμε να χρησιμοποιήσουμε το γνώρισμα **defer** στην ετικέτα `<script>` με τη σειρά με την οποία επιθυμούμε την εκτέλεσή τους.

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

```
<script src="js/script3.js" defer ></script>
```

Ο κώδικας `script2.js` θα εκτελεστεί πριν τον `script3.js`



# Μια αιτία σύγχυσης...

Σύγχρονη και ασύγχρονη εκτέλεση διεργασιών



σύγχρονη = με το ίδιο ρολόι χρονισμού



ασύγχρονη = με  
διαφορετικό ρολόι

# Επιλογές φόρτωσης κώδικα JS

- Γενικά τοποθετούμε το `<script>` μέσα στο `<head>` και χρησιμοποιούμε τα γνωρίσματα **async** και **defer** ώστε η JS να μην μπλοκάρει το κατέβασμα της σελίδας
- Χρησιμοποιούμε **`<script async ...>`** αν το περιεχόμενο της σελίδας είναι ανεξάρτητο από τον κώδικα JavaScript.
- Χρησιμοποιούμε **`<script defer ... >`** αν επιθυμούμε ο κώδικας να τρέξει μετά την ολοκλήρωση φορτώματος της HTML.
- Εναλλακτικά μπορούμε να χρησιμοποιήσουμε απλά το `<script>` αφού στο τέλος του `</body>` ώστε να μην μπλοκάρουμε το κατέβασμα της σελίδας παρά στο τέλος.



# Το web API της JS



Ο **navigator** αντιπροσωπεύει την κατάσταση και την ταυτότητα του φυλλομετρητή (user agent).

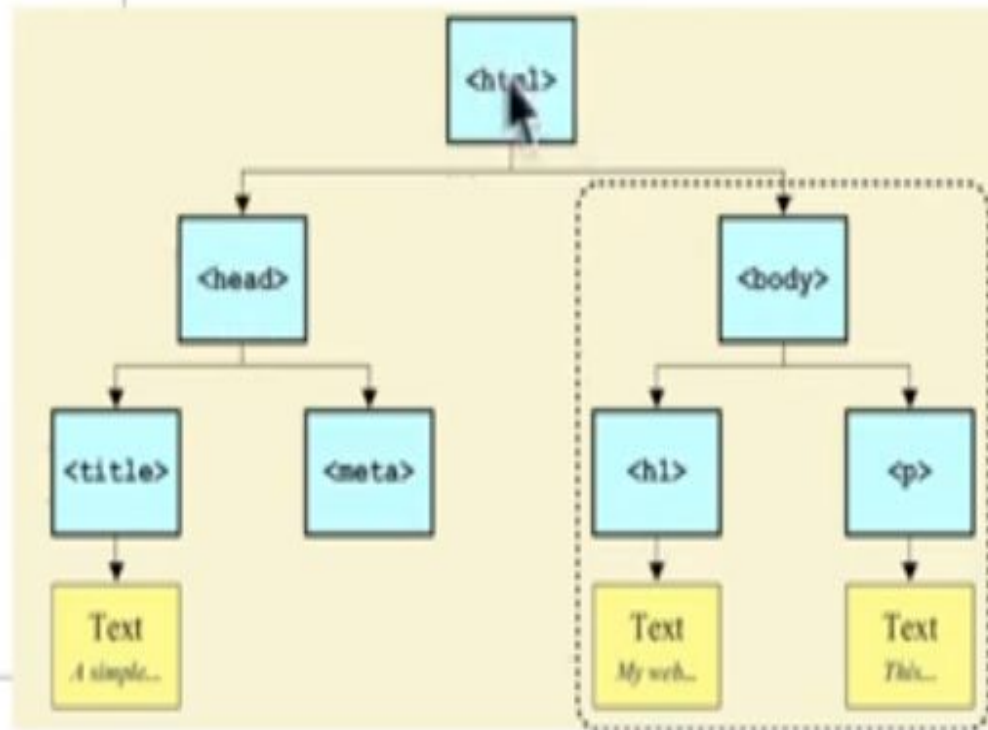
Το **window** είναι το παράθυρο που φορτώνεται η ιστοσελίδα. Από αυτό μπορούμε να πάρουμε το μέγεθος του παραθύρου (`Window.innerWidth`, `Window.innerHeight`), κλπ.

Το **document** (που αντιπροσωπεύεται από το DOM στον φυλλομετρητή) είναι η ιστοσελίδα που έχει φορτωθεί στο παράθυρο, περιέχει ιεραρχία στοιχείων, `<html>`, ...



# Παράδειγμα DOM

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <title>a simple web page</title>
</head>
<body>
 <h1>My Web Page</h1>
 <p>This is a welcome page</p>
</body>
</html>
```



# Κόμβοι του DOM

**Element node:** Ένα στοιχείο του DOM

**Root node:** Ο κορυφαίος κόμβος στο δέντρο, ο οποίος στην περίπτωση της HTML είναι πάντα ο κόμβος <html>

**Child node:** Ένας κόμβος απευθείας μέσα σε άλλο κόμβο.

**Descendant node:** Ένας κόμβος οπουδήποτε μέσα σε άλλο κόμβο.

**Parent node:** Ο κόμβος μέσα στον οποίο βρίσκεται ο κόμβος

**Sibling nodes:** Κόμβοι που είναι στο ίδιο επίπεδο του δένδρου DOM.

**Text node:** Ένας κόμβος που περιέχει κείμενο.

# Πρόσβαση της JS σε HTML elements

- `document.getElementById()`
- `document.querySelector()`

(δέχεται ως παράμετρο επιλογείς CSS)

`element.setAttribute(attribute, value)`

`element.getAttribute(attribute)`

`element.removeAttribute(attribute)`

`element.innerHTML`   `element.textContent`


`element.style` e.g. `element.style.fontSize = '20px';`

`document.getElementById(id)`

`document.getElementsByClassName(class)`

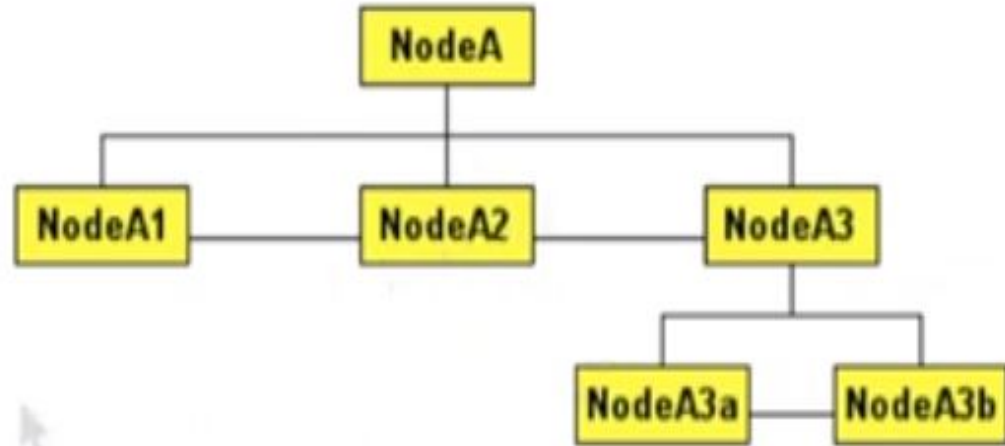
`document.getElementsByTagName(tagname)`

προσοχή στα ονόματα  
των παραμέτρων του  
CSS : από **font-size** σε  
**fontSize**



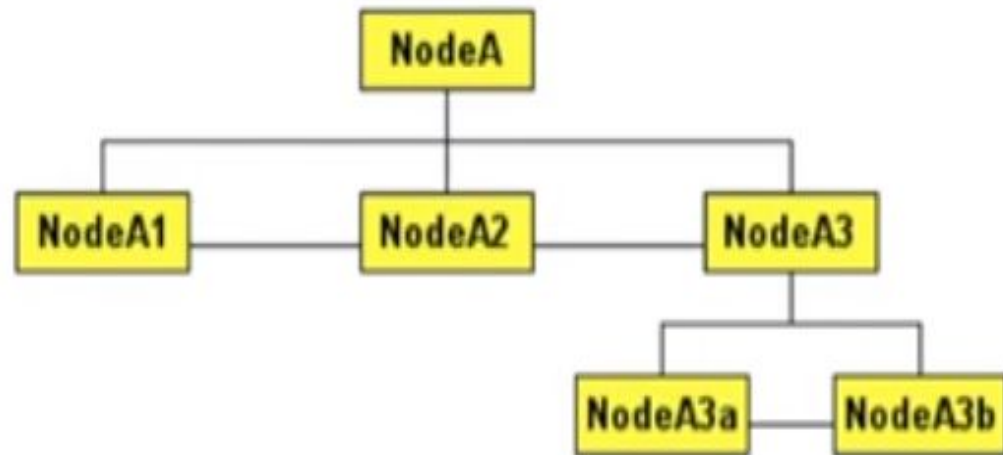
# Παράδειγμα αναφοράς σε στοιχεία του DOM

NodeA.firstChild  
NodeA.lastChild  
NodeA.childNodes.length  
NodeA.childNodes[0]  
NodeA.childNodes[1]  
NodeA.childNodes[2]  
NodeA1.parentNode  
NodeA1.nextSibling  
NodeA3.prevSibling  
NodeA3.nextSibling  
NodeA.lastChild.firstChild  
NodeA3b.parentNode.parentNode



# Παράδειγμα αναφοράς σε στοιχεία του DOM

NodeA.firstChild = NodeA1  
NodeA.lastChild = NodeA3  
NodeA.childNodes.length = 3  
NodeA.childNodes[0] = NodeA1  
NodeA.childNodes[1] = NodeA2  
NodeA.childNodes[2] = NodeA3  
NodeA1.parentNode = NodeA  
NodeA1.nextSibling = NodeA2  
NodeA3.prevSibling = NodeA2  
NodeA3.nextSibling = null  
NodeA.lastChild.firstChild = NodeA3a  
NodeA3b.parentNode.parentNode = NodeA





# Πώς λέγεται ο προπάππους?

```
minor = ['Am', 'A#m', 'Bm', 'Cm', 'C#m', 'Dm',
 'D#m', 'Em', 'Fm', 'F#m', 'Gm', 'G#m'];
buttons = document.querySelectorAll('button');
inputTransporto =
document.querySelectorAll('input');

for (let btn of buttons){
 btn.addEventListener('click', showHide);
}

for (let inp of inputTransporto){
 inp.addEventListener('keyup', (e) => {
 if(e.code==='Enter') {
 let u = transporto(inp,
inp.parentNode.parentNode.parentNode);
 if (u)
 inp.parentNode.parentNode.parentNode.querySelector("#lyrics").innerHTML = u;
 }
 })
}
```

## Επεξεργασία στοιχείων ενός element

```
let link = document.querySelector('a');
link.textContent = 'University of Patras';
link.href = 'https://www.upatras.gr';
```



## Δημιουργία, Διαγραφή στοιχείων

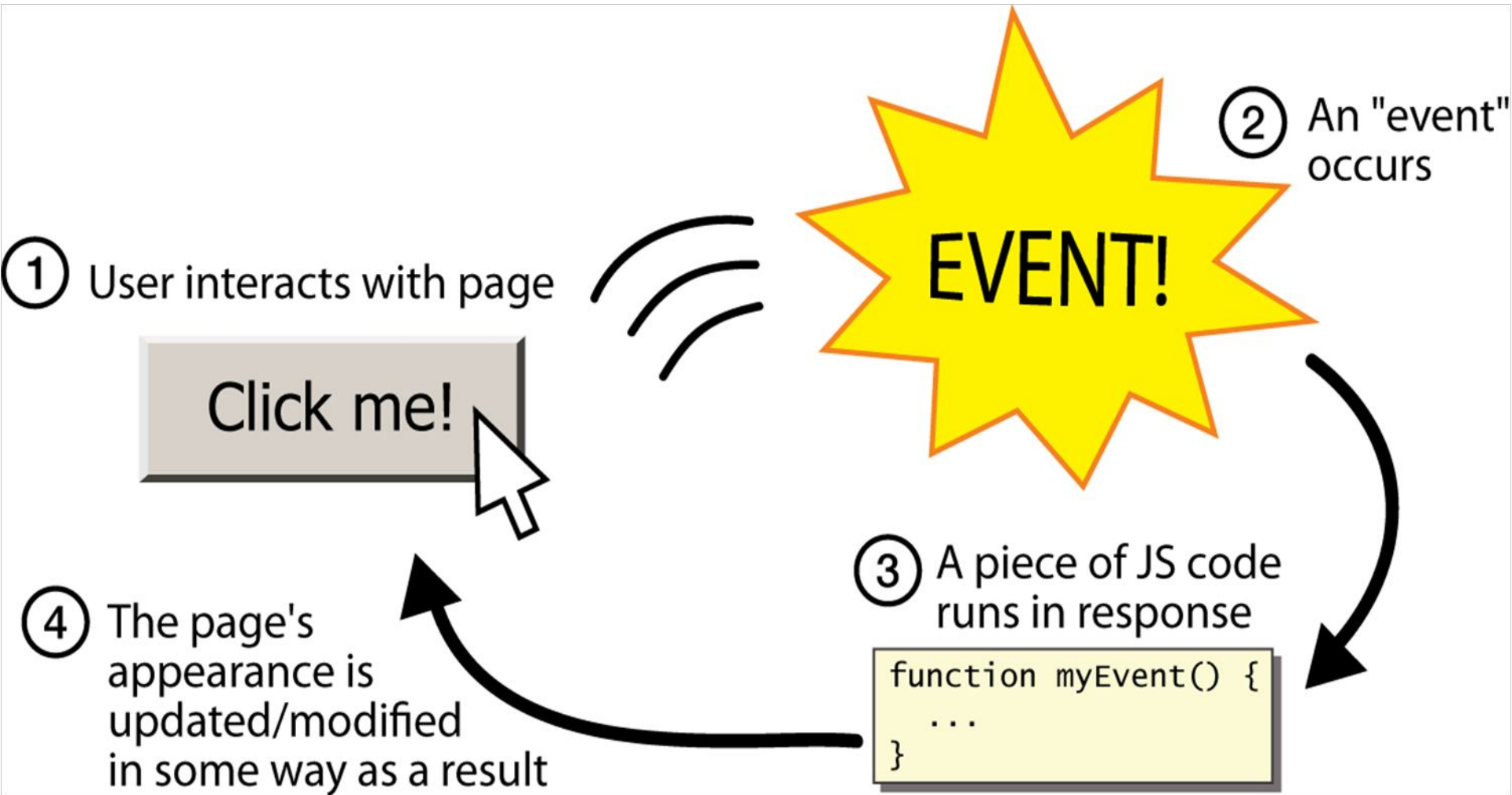
Έστω ότι θέλουμε να δημιουργήσουμε ένα κόμβο <p> κάτω από τον κόμβο <section>

```
let sect = document.querySelector('section');
let para = document.createElement('p');
para.textContent = 'Thank you for visiting!';
sect.appendChild(para);
```

Έστω ότι θέλουμε να διαγράψουμε ένα κόμβο

```
sect.removeChild(linkPara); // to remove a child of element sect
```

# Σύνδεση HTML-JS



# Events

- Είναι ενέργειες που γίνονται είτε από τον browser είτε από τον χρήστη και πυροδοτούν την κλήση μιας συνάρτησης

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

# Events

- Μπορούμε να δημιουργήσουμε ένα event για ένα HTML element θέτοντας έναν event listener και ορίζοντας τη συνάρτηση που θα κληθεί (event handler)

`element.addEventListener(event,function)`

- Διαφορετικά: απευθείας στο element

`<element event="function()">`

- Αφαίρεση eventListener

`element.removeEventListener(event,function)`

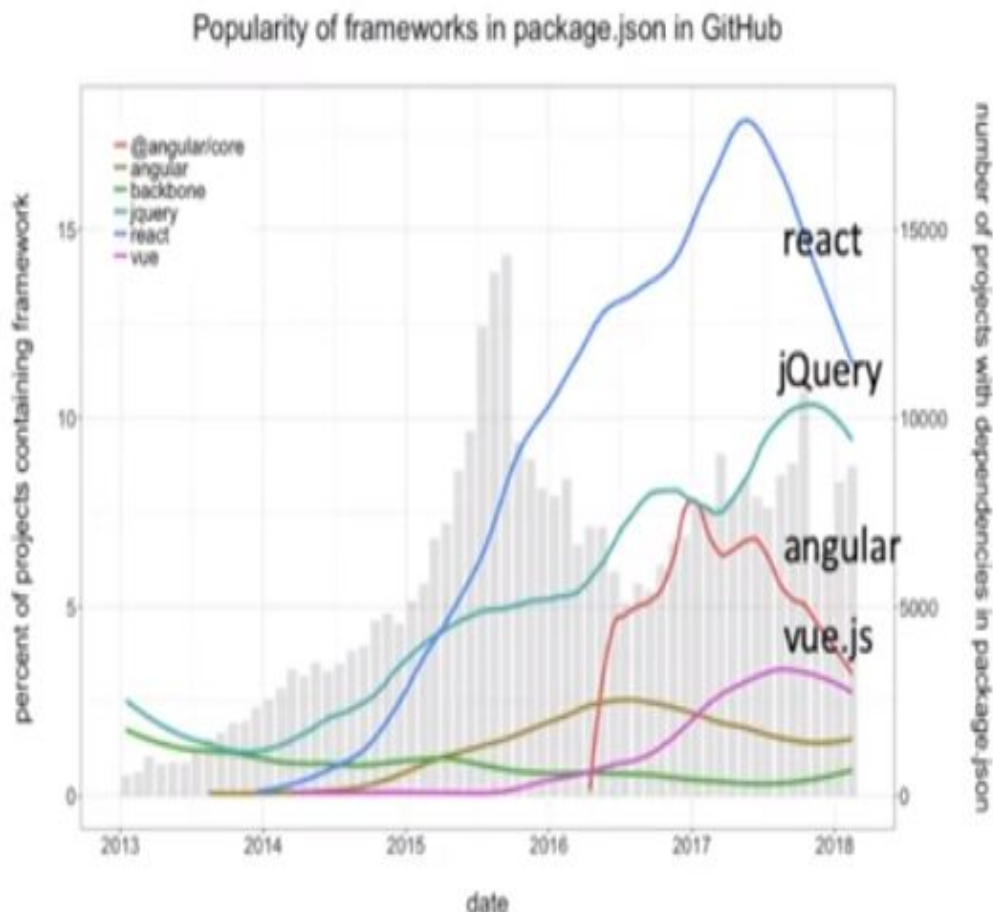
## Επικοινωνία της JS με τον χρήστη (δεξί κλικ Έλεγχος...Console)

- Από την κονσόλα του browser: `console.log('hello')`
- alert box: `window.alert('hello')` // μήνυμα σε αναδυόμενο παράθυρο
- prompt: `window.prompt('name:')` // εισαγωγή πληροφορίας
- Με τροποποίηση ενός element: `innerHTML('hello')`
- Ή με αλλαγή του `textContent`: `element.textContent = 'hello'`
- εισαγωγή στοιχείων σε φόρμα:  
όνομα: `<input id="in1" type="text">`  
`console.log(document.querySelector('#in1').value);`

# Το οικοσύστημα της JS

## Τα πιο σημαντικά frameworks της JS

- React
- jQuery
- Angular
- Vue.js

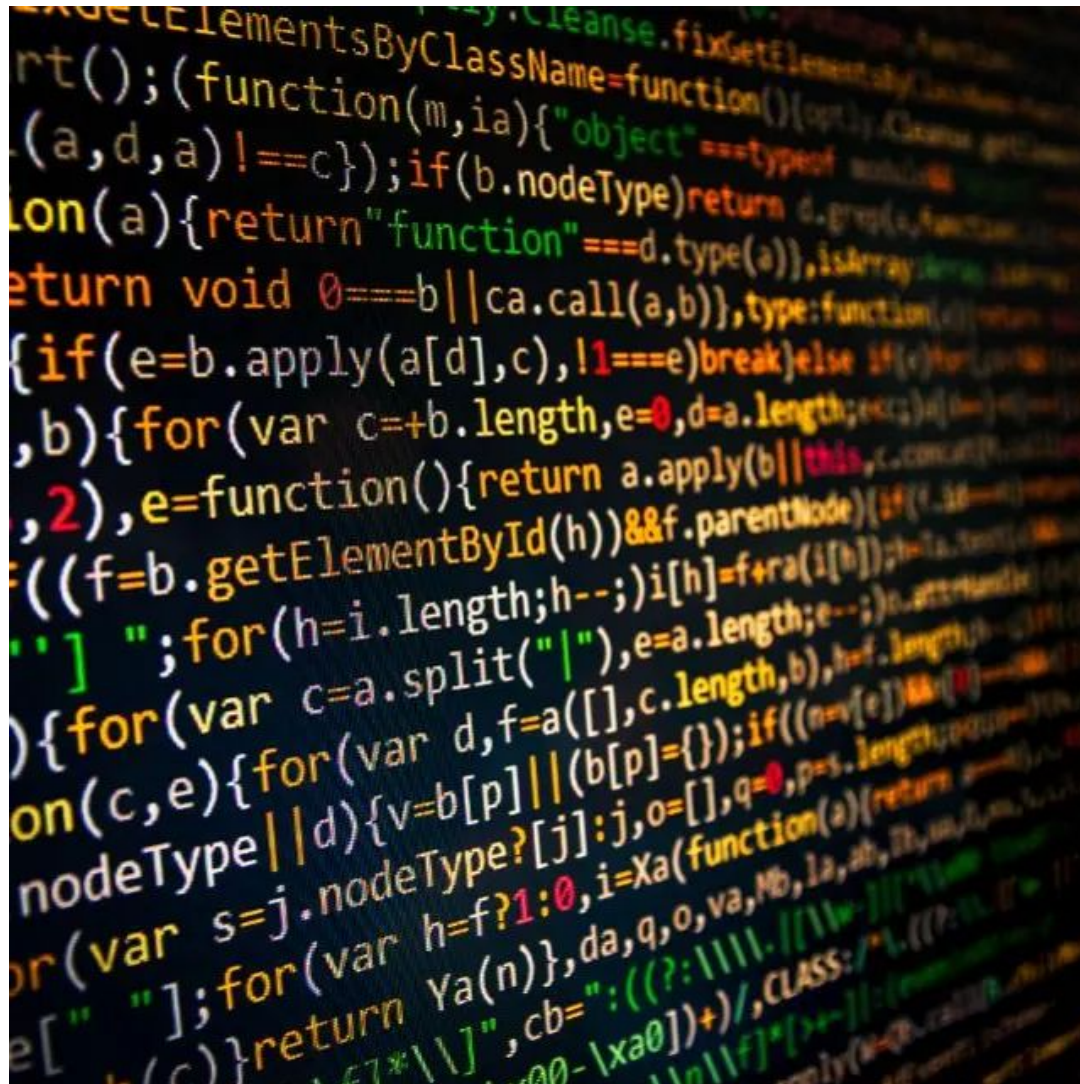


## References

Εισαγωγή στην ανάπτυξη ιστοσελίδων με  
HTML5, CSS3, Javascript - Νίκος Αβούρης  
Ιστότοπος W3schools.com



# Hands on practice now!



```
rt();(function(m,ia){"object"===typeof
(a,d,a)!==c});if(b.nodeType)return d.get(a,function
lon(a){return"function"===d.type(a)},isArray
return void 0===b||ca.call(a,b)},type:function
{if(e=b.apply(a[d],c),!1===e)break}else if
,b){for(var c+=b.length,e=0,d=a.length;e<d
,2),e=function(){return a.apply(b||this
((f=b.getElementById(h))&&f.parentNode){if
";for(h=i.length;h--;)i[h]=f+a(i[h]);
{for(var c=a.split("|"),e=a.length;e--;
on(c,e){for(var d,f=a([],c.length,b),h=f.length
nodeType||d){v=b[p]||(b[p]={});if((n=v[e])&&(p
or(var s=j.nodeType?[j]:j,o=[],q=0,p=s.length;e--)
e[""];for(var Ya(n),da,q,o,va,Mb,la,th,un,lm,va,lf
e[""];for(var Ya(n),da,q,o,va,Mb,la,th,un,lm,va,lf
```