



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

## 3.1. Πακέτο Εκπαιδευτικού Υλικού του έργου Code4SP

WP3:  
Εκπαιδευτικό Υλικό του  
έργου Code4SP

Εκπονητής:



## Πληροφορίες αναφορικά με το Έργο

Ακρώνυμο του Έργου: Code4SP

Τίτλος Έργου: Κωδικοποίηση για Κοινωνική Ένταξη/ Ενσωμάτωση

Αριθμός Έργου: 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN

Ιστοσελίδα του έργου: [www.code4sp.eu](http://www.code4sp.eu)

Συνεργάτης εξουσιοδότησης: CEPROF

Έκδοση Εγγράφου: 1.0

Ημερομηνία Προετοιμασίας: 11/03/2022

Ιστορικό Εγγράφου			
Ημερομηνία	Έκδοση	Συντάκτης	Περιγραφή
11/03/2022	1	SHA	Προσχέδιο
29/03/2022	2	SHA	Επανάληψη

## Πίνακας Περιεχομένων

Πληροφορίες αναφορικά με το έργο	2
JavaScript Window – Το Μοντέλο Αντικειμένου του Προγράμματος Περιήγησης	6
Το Μοντέλο Αντικειμένου Προγράμματος Περιήγησης (BOM)	6
Το Αντικείμενο Window	6
Μέγεθος Window	7
Άλλες μέθοδοι Window	7
Οθόνη Window JavaScript (Screen)	7
Οθόνη Window Screen	8
Πλάτος Οθόνης Window Screen (Width)	8
Ύψος Οθόνης Window Screen (Height)	9
Διαθέσιμο Πλάτος Οθόνης Window (Screen Available Width)	9
Διαθέσιμο Ύψος Οθόνης Window (Screen Available Height)	10
Βάθος Χρώματος Οθόνης Window (Screen Color Depth)	10
Βάθος Εικονοστοιχείων Window (Screen Pixel Depth)	11
Θέση JavaScript Window (Location)	11
Window Location Href	12
Window Location Hostname	13
Window Location Pathname	13
Πρωτόκολλο Θέσης Window (Location Protocol)	14
Θέση Θύρας Window (Location Port)	15
Ανάθεση Θέσης Window (Location Assign)	16
JavaScript Window History	16
Ιστορικό Window (History)	16
Window History Back	17
Window History Forward	17
Πλοηγητής JavaScript Window (Navigator)	18
Πλοηγητής Window	18
Cookies Προγράμματος Περιήγησης	19

Όνομα Εφαρμογής Προγράμματος Περιήγησης (Browser Application Name)	19
Όνομα Κώδικα Εφαρμογής Προγράμματος Περιήγησης (Browser Application Code Name)	19
Η Μηχανή Προγράμματος Περιήγησης (Browser Engine)	19
Η Έκδοση Προγράμματος Περιήγησης (The Browser Version)	19
The Browser Agent	19
Προσοχή !!!	20
Η Πλατφόρμα Προγράμματος Περιήγησης (The Browser Platform)	20
Η Γλώσσα Προγραμματισμού Προγράμματος Περιήγησης (The Browser Language)	20
Είναι συνδεδεμένο το πρόγραμμα περιήγησης στο διαδίκτυο;	20
Είναι ενεργοποιημένη η Java;	20
Αναδυόμενα Πλαίσια JavaScript (Popur Boxes)	20
Πλαίσιο Ειδοποίησης (Alert Box)	21
Σύνταξη (Syntax)	21
Πλαίσιο Επιβεβαίωσης (Confirm Box)	21
Σύνταξη	21
Πλαίσιο Προτροπής (Prompt Box)	22
Σύνταξη	22
Αλλαγή Γραμμής (Line Breaks)	23
Συμβάντα Συγχρονισμού JavaScript (Timing Events)	23
Συμβάντα Συγχρονισμού (Timing Events)	23
Η μέθοδος setTimeout()	24
Πώς να σταματήσω την εκτέλεση;	25
Η μέθοδος setInterval()	25
Πώς να σταματήσω την εκτέλεση?	26
JavaScript Cookies	27
Τί είναι τα Cookies?	27
Δημιουργία ενός Cookie με την JavaScript	28
Ανάγνωση ενός Cookie με την JavaScript	29
Αλλαγή ενός Cookie με την JavaScript	29
Διαγραφή ενός Cookie με την JavaScript	29
Η συμβολοσειρά Cookie	30
Παράδειγμα JavaScript Cookie	30

Μια Συνάρτηση για τον ορισμό της τιμής ενός Cookie	31
Μια Συνάρτηση για τη λήψη ενός Cookie	31
Μια Συνάρτηση για τον έλεγχο ενός Cookie	33

## Window JavaScript - Το Μοντέλο Αντικειμένου του Προγράμματος Περιήγησης

Το Browser Object Model (BOM) επιτρέπει στην JavaScript να «μιλήσει» στο πρόγραμμα περιήγησης.

### Το Μοντέλο Αντικειμένου του Προγράμματος Περιήγησης (BOM)

Δεν υπάρχουν επίσημα πρότυπα για το Browser Object Model (BOM).

Δεδομένου ότι τα σύγχρονα προγράμματα περιήγησης έχουν εφαρμόσει (σχεδόν) τις ίδιες μεθόδους και ιδιότητες για τη διαδραστικότητα της JavaScript, συχνά αυτά αναφέρονται, ως μέθοδοι και ιδιότητες του BOM.

### Το αντικείμενο window

Το αντικείμενο window υποστηρίζεται από όλα τα προγράμματα περιήγησης. Αντιπροσωπεύει το παράθυρο του προγράμματος περιήγησης.

Όλα τα παγκόσμια αντικείμενα JavaScript, οι λειτουργίες και οι μεταβλητές γίνονται αυτόματα μέλη του αντικειμένου window.

Οι καθολικές μεταβλητές είναι ιδιότητες του αντικειμένου window.



Οι καθολικές λειτουργίες είναι μέθοδοι του αντικειμένου window.

Ακόμα και το αντικείμενο εγγράφου (του HTML DOM) είναι μια ιδιότητα του αντικειμένου window:

```
window.document.getElementById("header");
```

είναι το ίδιο με:

```
document.getElementById("header");
```

## Μέγεθος Window

Δύο ιδιότητες μπορούν να χρησιμοποιηθούν για τον προσδιορισμό του μεγέθους του window του προγράμματος περιήγησης.

Και οι δύο ιδιότητες επιστρέφουν τα μεγέθη σε εικονοστοιχεία:

- window.innerHeight - το εσωτερικό ύψος του window του προγράμματος περιήγησης (σε εικονοστοιχεία)
- window.innerWidth - το εσωτερικό πλάτος του παραθύρου του προγράμματος περιήγησης (σε εικονοστοιχεία)

```
let w = window.innerWidth;  
let h = window.innerHeight;
```

## Άλλες μέθοδοι παραθύρων

Άλλες μέθοδοι

- window.open() - άνοιγμα νέου window

- `window.close()` - κλείσιμο του τρέχοντος window
- `window.moveTo()` - μετακίνηση του τρέχοντος window
- `window.resizeTo()` - αλλαγή μεγέθους του τρέχοντος window

## Οθόνη window JavaScript

Το αντικείμενο `window.screen` περιέχει πληροφορίες σχετικά με την οθόνη του χρήστη.

## Οθόνη window

Το αντικείμενο `window.screen` μπορεί να γραφτεί χωρίς το πρόθεμα του `window`.

Ιδιότητες:

- `screen.width`
- `screen.height`
- `screen.availWidth`
- `screen.availHeight`
- `screen.colorDepth`
- `screen.pixelDepth`

## Πλάτος οθόνης window

Η ιδιότητα `screen.width` επιστρέφει το πλάτος της οθόνης του επισκέπτη σε εικονοστοιχεία.



## Example

Display the width of the screen in pixels:

```
document.getElementById("demo").innerHTML =  
"Screen Width: " + screen.width;
```

Result will be:

```
Screen Width: 1280
```

Εικόνα 1 – Πλάτος οθόνης (Πηγή: [https://www.w3schools.com/js/js\\_window\\_screen.asp](https://www.w3schools.com/js/js_window_screen.asp))

## Ύψος οθόνης window

Η ιδιότητα `screen.height` επιστρέφει το ύψος της οθόνης του επισκέπτη σε εικονοστοιχεία.

### Example

Display the height of the screen in pixels:

```
document.getElementById("demo").innerHTML =  
"Screen Height: " + screen.height;
```

Result will be:

```
Screen Height: 720
```

Εικόνα 1 – Πλάτος οθόνης (Πηγή: [https://www.w3schools.com/js/js\\_window\\_screen.asp](https://www.w3schools.com/js/js_window_screen.asp))

## Πλάτος οθόνης window

Η ιδιότητα `screen.availWidth` επιστρέφει το πλάτος της οθόνης του επισκέπτη, σε εικονοστοιχεία, πλην των χαρακτηριστικών διεπαφής όπως η γραμμή εργασιών Window (taskbar).

## Example

Display the available width of the screen in pixels:

```
document.getElementById("demo").innerHTML =  
"Available Screen Width: " + screen.availWidth;
```

Result will be:

```
Available Screen Width: 1280
```

Εικόνα 1 – Πλάτος οθόνης (Πηγή: [https://www.w3schools.com/js/js\\_window\\_screen.asp](https://www.w3schools.com/js/js_window_screen.asp))

## Ύψος οθόνης window

Η ιδιότητα `screen.availWidth` επιστρέφει το πλάτος της οθόνης του επισκέπτη, σε εικονοστοιχεία, πλην τα χαρακτηριστικά διεπαφής όπως η γραμμή εργασιών Window (taskbar).

## Example

Display the available height of the screen in pixels:

```
document.getElementById("demo").innerHTML =  
"Available Screen Height: " + screen.availHeight;
```

Result will be:

```
Available Screen Height: 680
```

Εικόνα 1 – Πλάτος οθόνης (Πηγή: [https://www.w3schools.com/js/js\\_window\\_screen.asp](https://www.w3schools.com/js/js_window_screen.asp))

## Βάθος χρώματος οθόνης window

Η ιδιότητα `screen.colorDepth` επιστρέφει τον αριθμό των bit που χρησιμοποιήθηκαν για την εμφάνιση ενός χρώματος.

Όλοι οι σύγχρονοι υπολογιστές χρησιμοποιούν 24 bit ή 32 bit υλικό για την ανάλυση του χρώματος:

24 bits = 16.777.216 διαφορετικά "Αληθινά χρώματα" (True Colours)

32 bit = 4.294.967.296 διαφορετικά "Βαθιά χρώματα" (Deep Colors)

Οι παλαιότεροι υπολογιστές χρησιμοποιούσαν 16 bits: 65,536 διαφορετικά "Υψηλά Χρώματα" ("High Colors") για ανάλυση.

Οι πολύ παλιοί υπολογιστές, και τα παλιά κινητά τηλέφωνα χρησιμοποιούσαν 8 bits: 256 διαφορετικά "χρώματα VGA".



Example

Display the color depth of the screen in bits:

```
document.getElementById("demo").innerHTML = "Screen Color Depth: " + screen.colorDepth;
```

Result will be:

Screen Color Depth: 24

Εικόνα 5 – Πλάτος οθόνης (Πηγή: [https://www.w3schools.com/js/js\\_window\\_screen.asp](https://www.w3schools.com/js/js_window_screen.asp))

## Βάθος εικονοστοιχείου οθόνης window

Η ιδιότητα `screen.pixelDepth` επιστρέφει το βάθος εικονοστοιχείου της οθόνης.

## Example

Display the pixel depth of the screen in bits:

```
document.getElementById("demo").innerHTML =  
"Screen Pixel Depth: " + screen.pixelDepth;
```

Result will be:

```
Screen Pixel Depth: 24
```

Εικόνα 6 – Βάθος εικονοστοιχείου οθόνης (Πηγή: [https://www.w3schools.com/js/js\\_window\\_screen.asp](https://www.w3schools.com/js/js_window_screen.asp))

## Τοποθεσία window στην JavaScript (Location)

Το αντικείμενο `window.location` μπορεί να χρησιμοποιηθεί για τη λήψη της τρέχουσας διεύθυνσης σελίδας (URL) και για την ανακατεύθυνση του προγράμματος περιήγησης σε μια νέα σελίδα.

Τοποθεσία window

Το αντικείμενο `window.screen` μπορεί να γραφτεί χωρίς το πρόθεμα του window.

Για παράδειγμα:

- Η ιδιότητα `window.location.href` επιστρέφει το href (URL) της τρέχουσας σελίδας
- Η ιδιότητα `window.location.hostname` επιστρέφει το όνομα τομέα (domain name) της φιλοξενίας ιστοσελίδων (web host)
- Η ιδιότητα `window.location.pathname` επιστρέφει τη διαδρομή (path) και το όνομα αρχείου (filename) της τρέχουσας σελίδας
- Η ιδιότητα `window.location.protocol` επιστρέφει το πρωτόκολλο ιστού που χρησιμοποιήθηκε (http: ή https:)
- Η ιδιότητα `window.location.assign()` φορτώνει ένα νέο έγγραφο

## Window Location

Η ιδιότητα `window.location.href` επιστρέφει τη διεύθυνση URL της τρέχουσας σελίδας.

### Example

Display the href (URL) of the current page:

```
document.getElementById("demo").innerHTML =  
"Page location is " + window.location.href;
```

Result is:

```
Page location is https://www.w3schools.com/js/js_window_location.asp
```

Εικόνα 7 – Τοποθεσία href (Πηγή: [https://www.w3schools.com/js/js\\_window\\_location.asp](https://www.w3schools.com/js/js_window_location.asp))

## Window Location Hostname

Η ιδιότητα `window.location.hostname` επιστρέφει το όνομα του internet host (της τρέχουσας σελίδας).

### Example

Display the name of the host:

```
document.getElementById("demo").innerHTML =  
"Page hostname is " + window.location.hostname;
```

Result is:

```
Page hostname is www.w3schools.com
```



Εικόνα 7 – Τοποθεσία href (Πηγή: [https://www.w3schools.com/js/js\\_window\\_location.asp](https://www.w3schools.com/js/js_window_location.asp))

## Window Location Pathname

Η ιδιότητα `window.location.pathname` επιστρέφει το `pathname` της τρέχουσας σελίδας.

Example

Display the path name of the current URL:

```
document.getElementById("demo").innerHTML = "Page path is " + window.location.pathname;
```

Result is:

```
Page path is /js/js_window_location.asp
```

Εικόνα 7 – Τοποθεσία href (Πηγή: [https://www.w3schools.com/js/js\\_window\\_location.asp](https://www.w3schools.com/js/js_window_location.asp))

## Window Location Protocol

Η ιδιότητα `window.location.protocol` επιστρέφει το πρωτόκολλο ιστού της σελίδας.



## Example

Display the web protocol:

```
document.getElementById("demo").innerHTML =  
"Page protocol is " + window.location.protocol;
```

Result is:

```
Page protocol is https:
```

Εικόνα 7 – Τοποθεσία href (Πηγή: [https://www.w3schools.com/js/js\\_window\\_location.asp](https://www.w3schools.com/js/js_window_location.asp))

## Τοποθεσία window

Η ιδιότητα `window.location.port` επιστρέφει τον αριθμό της θύρας φιλοξενίας διαδικτύου (της τρέχουσας σελίδας).

### Example

Display the name of the host:

```
document.getElementById("demo").innerHTML =  
"Port number is " + window.location.port;
```

Result is:

```
Port number is
```

Εικόνα 7 – Τοποθεσία href (Πηγή: [https://www.w3schools.com/js/js\\_window\\_location.asp](https://www.w3schools.com/js/js_window_location.asp))

## Ανάθεση τοποθεσίας παραθύρου (Window Location Assign)

Η ιδιότητα `window.location.assign()` φορτώνει ένα νέο έγγραφο

Example

Load a new document:

```

<html>
<head>
<script>
function newDoc() {
  window.location.assign("https://www.w3schools.com")
}
</script>
</head>
<body>

<input type="button" value="Load new document" onclick="newDoc()">

</body>
</html>

```

Εικόνα 7 – Τοποθεσία href (Πηγή: [https://www.w3schools.com/js/js\\_window\\_location.asp](https://www.w3schools.com/js/js_window_location.asp))

## Ιστορικό window στην JavaScript

Το αντικείμενο `window.history` περιέχει το ιστορικό του προγράμματος περιήγησης.

## Ιστορικό window

Το αντικείμενο `window.screen` μπορεί να γραφτεί χωρίς το πρόθεμα του `window`.

Για την προστασία του απορρήτου των χρηστών, υπάρχουν περιορισμοί στον τρόπο με τον οποίο η JavaScript μπορεί να έχει πρόσβαση σε αυτό το αντικείμενο.

Ορισμένες μέθοδοι:

- `history.back()` - το ίδιο με το να κάνετε κλικ πίσω στο πρόγραμμα περιήγησης
- `history.forward()` - το ίδιο με το να κάνετε κλικ προς τα εμπρός στο πρόγραμμα περιήγησης

## Ιστορικό window

Η μέθοδος `history.back()` φορτώνει την προηγούμενη διεύθυνση URL στη λίστα ιστορικού.

Αυτό είναι το ίδιο με το να κάνετε κλικ στο κουμπί Πίσω στο πρόγραμμα περιήγησης.

Example


Create a back button on a page:

```
<html>
<head>
<script>
function goBack() {
  window.history.back()
}
</script>
</head>
<body>

<input type="button" value="Back" onclick="goBack()">

</body>
</html>
```

The output of the code above will be:



Εικόνα 13 – Ιστορικό (Πηγή: [https://www.w3schools.com/js/js\\_window\\_history.asp](https://www.w3schools.com/js/js_window_history.asp))

## Πρώθηση ιστορικού window

Η μέθοδος `history.forward()` φορτώνει την επόμενη διεύθυνση URL στη λίστα ιστορικού.

Αυτό είναι το ίδιο με το να κάνετε κλικ στο κουμπί Forward στο πρόγραμμα περιήγησης.

**Example**

Create a forward button on a page:

```
<html>
<head>
<script>
function goForward() {
  window.history.forward()
}
</script>
</head>
<body>

<input type="button" value="Forward" onclick="goForward()">

</body>
</html>
```

The output of the code above will be:



Εικόνα 13 – Ιστορικό (Πηγή: [https://www.w3schools.com/js/js\\_window\\_history.asp](https://www.w3schools.com/js/js_window_history.asp))

## Πλοηγός window στην JavaScript

Το αντικείμενο `window.navigator` περιέχει πληροφορίες σχετικά με το πρόγραμμα περιήγησης του επισκέπτη.

## Περιηγητής window

Το αντικείμενο `window.screen` μπορεί να γραφτεί χωρίς το πρόθεμα του `window`.

Για παράδειγμα:

- `navigator.appName`
- `navigator.appCodeName`
- `navigator.platform`

## Cookies προγράμματος περιήγησης

Η ιδιότητα `CookieEnabled` επιστρέφει την τιμή `true` αν τα cookies είναι ενεργοποιημένα, διαφορετικά επιστρέφει την τιμή `false`:

```
document.getElementById("demo").innerHTML =  
"cookiesEnabled is " + navigator.cookieEnabled;
```

Εικόνα 15 – Cookies προγράμματος περιήγησης (Πηγή: [https://www.w3schools.com/js/js\\_window\\_navigator.asp](https://www.w3schools.com/js/js_window_navigator.asp))

## Εφαρμογή προγράμματος περιήγησης

Η ιδιότητα `appName` επιστρέφει το όνομα της εφαρμογής του προγράμματος περιήγησης.

## Όνομα Κώδικα Εφαρμογής Προγράμματος Περιήγησης (Browser Application Code Name)

Η ιδιότητα `appCodeName` επιστρέφει το όνομα κώδικα της εφαρμογής του προγράμματος περιήγησης.

## Η μηχανή του προγράμματος περιήγησης

Η ιδιότητα `product` (product) επιστρέφει το όνομα προϊόντος της μηχανής του προγράμματος περιήγησης.

## Έκδοση Προγράμματος Περιήγησης (The Browser Version)

Η ιδιότητα `appVersion` επιστρέφει πληροφορίες έκδοσης σχετικά με το πρόγραμμα περιήγησης.



## Browser Agent

Η ιδιότητα userAgent επιστρέφει την κεφαλίδα user-agent που αποστέλλεται από το πρόγραμμα περιήγησης στο διακομιστή.

## Προσοχή!!

Οι πληροφορίες από το αντικείμενο του πλοηγού μπορεί συχνά να είναι παραπλανητικές και δεν πρέπει να χρησιμοποιούνται για την ανίχνευση εκδόσεων του προγράμματος περιήγησης επειδή:

- Διαφορετικά προγράμματα περιήγησης μπορούν να χρησιμοποιήσουν το ίδιο όνομα
- Τα δεδομένα του πλοηγού μπορούν να αλλάξουν από τον κάτοχο του προγράμματος περιήγησης
- Ορισμένα προγράμματα περιήγησης αυτοπροσδιορίζονται εσφαλμένα για να παρακάμψουν τις δοκιμές ιστότοπου
- Τα προγράμματα περιήγησης δεν μπορούν να αναφέρουν νέα λειτουργικά συστήματα, τα οποία θα κυκλοφορήσουν αργότερα από το πρόγραμμα περιήγησης

## Η πλατφόρμα του προγράμματος περιήγησης (browser platform)

Η ιδιότητα της πλατφόρμας επιστρέφει την πλατφόρμα του προγράμματος περιήγησης (λειτουργικό σύστημα).

## Γλώσσα Προγράμματος Περιήγησης (Browser Language)

Η ιδιότητα γλώσσας επιστρέφει τη γλώσσα του προγράμματος περιήγησης.

### Είναι το πρόγραμμα περιήγησης σε σύνδεση;

Η ιδιότητα onLine επιστρέφει την τιμή "αληθές" εάν το πρόγραμμα περιήγησης είναι συνδεδεμένο.

### Είναι ενεργοποιημένη η Java;

Η μέθοδος javaEnabled () επιστρέφει την τιμή "αληθές" αν είναι ενεργοποιημένη η Java.

## Αναδυόμενα πλαίσια JavaScript (Popup Boxes)

Το JavaScript έχει τρία είδη αναδυόμενων πλαισίων: πλαίσιο ειδοποίησης (alert box), πλαίσιο επιβεβαίωσης (confirm box) και πλαίσιο προτροπής (prompt box).

## Πλαίσιο Ειδοποίησης

Ένα πλαίσιο ειδοποίησης χρησιμοποιείται συχνά εάν θέλετε να βεβαιωθείτε ότι οι πληροφορίες φτάνουν στο χρήστη.

Όταν εμφανιστεί ένα πλαίσιο ειδοποίησης, ο χρήστης θα πρέπει να κάνει κλικ στο "OK" για να συνεχίσει.

## Σύνταξη

```
window.alert("sometext");
```

Η μέθοδος `window.alert ()` μπορεί να γραφτεί χωρίς το πρόθεμα `window`.

```
alert("I am an alert box!");
```

## Επιβεβαίωση πλαισίου

Ένα πλαίσιο επιβεβαίωσης χρησιμοποιείται συχνά εάν θέλετε ο χρήστης να επαληθεύσει ή να αποδεχτεί κάτι.

Όταν εμφανιστεί ένα πλαίσιο επιβεβαίωσης, ο χρήστης θα πρέπει να κάνει κλικ είτε στο "OK" είτε στο "Cancel" για να συνεχίσει.

Εάν ο χρήστης κάνει κλικ στο "OK", το πλαίσιο επιστρέφει την τιμή "αληθές". Εάν ο χρήστης κάνει κλικ στην επιλογή "Ακύρωση", το πλαίσιο επιστρέφει την τιμή "ψευδές".

## Σύνταξη

```
window.confirm("sometext");
```

Η μέθοδος `window.alert ()` μπορεί να γραφτεί χωρίς το πρόθεμα `window`.

```
if (confirm("Press a button!")) {  
    txt = "You pressed OK!";  
} else {  
    txt = "You pressed Cancel!";  
}
```

Εικόνα 16 – Επιβεβαίωση πλαισίου (Πηγή: [https://www.w3schools.com/js/js\\_popup.asp](https://www.w3schools.com/js/js_popup.asp))

## Πλαίσιο Προτροπής

Ένα πλαίσιο προτροπής χρησιμοποιείται συχνά εάν θέλετε ο χρήστης να εισάγει μια τιμή πριν από την είσοδο σε μια σελίδα.

Όταν εμφανιστεί ένα πλαίσιο προτροπής, ο χρήστης θα πρέπει να κάνει κλικ είτε στο "OK" είτε στο "Cancel" για να προχωρήσει μετά την εισαγωγή μιας τιμής εισόδου.

Εάν ο χρήστης κάνει κλικ στο "OK", το πλαίσιο επιστρέφει την τιμή εισόδου. Εάν ο χρήστης κάνει κλικ στην επιλογή "Ακύρωση", το πλαίσιο επιστρέφει την τιμή "ψευδές".

## Σύνταξη

```
window.prompt("sometext", "defaultText");
```

Η μέθοδος `window.alert()` μπορεί να γραφτεί χωρίς το πρόθεμα `window`.

```
let person = prompt("Please enter your name", "Harry Potter");
let text;
if (person == null || person == "") {
  text = "User cancelled the prompt.";
} else {
  text = "Hello " + person + "! How are you today?";
}
```

Εικόνα 17 – Προτροπή (Πηγή: [https://www.w3schools.com/js/js\\_popup.asp](https://www.w3schools.com/js/js_popup.asp))

## Αλλαγή γραμμής (Line Break):

Για να εμφανιστεί η λειτουργία αλλαγής γραμμής μέσα σε ένα αναδυόμενο πλαίσιο, χρησιμοποιήστε μια κάθετο backlash ακολουθούμενη από το χαρακτήρα `n`.

```
alert("Hello\nHow are you?");
```

## Συμβάντα Συγχρονισμού JavaScript (Timing Events)

### Συμβάντα Συγχρονισμού

Το αντικείμενο window επιτρέπει την εκτέλεση του κώδικα σε καθορισμένα χρονικά διαστήματα.

Αυτά τα χρονικά διαστήματα ονομάζονται συμβάντα συγχρονισμού.

Οι δύο βασικές μέθοδοι που χρησιμοποιούνται με το JavaScript είναι:

- `setTimeout(function,milliseconds)`:  
Εκτελεί μια συνάρτηση, μετά από αναμονή ενός καθορισμένου αριθμού χιλιοστών του δευτερολέπτου.
- `setInterval(function,milliseconds)`:  
Εκτελεί το ίδιο με το `setTimeout()`, αλλά επαναλαμβάνει την εκτέλεση της συνάρτησης συνεχώς.

Το `setTimeout()` και το `setInterval()` είναι και οι δύο μέθοδοι του αντικειμένου HTML DOM Window.

### Η μέθοδος `setTimeout()`

```
window.setTimeout(function, milliseconds);
```

Η μέθοδος `window.alert()` μπορεί να γραφτεί χωρίς το πρόθεμα `window`.

Η πρώτη παράμετρος είναι μια συνάρτηση που θα εκτελεστεί.



Η δεύτερη παράμετρος δείχνει τον αριθμό των χιλιοστών του δευτερολέπτου πριν από την εκτέλεση.

## Example

Click a button. Wait 3 seconds, and the page will alert "Hello":

```
<button onclick="setTimeout(myFunction, 3000)">Try it</button>

<script>
function myFunction() {
  alert('Hello');
}
</script>
```

Εικόνα 18 – Ορισμός χρονικού ορίου (Πηγή: [https://www.w3schools.com/js/js\\_timing.asp](https://www.w3schools.com/js/js_timing.asp))

## Πώς να σταματήσετε την εκτέλεση;

Η μέθοδος `clearTimeout()` σταματά την εκτέλεση της λειτουργίας που καθορίζεται στο `setTimeout()`.

```
window.clearTimeout(timeoutVariable)
```

Η μέθοδος `window.alert()` μπορεί να γραφτεί χωρίς το πρόθεμα `window`.

Η μέθοδος `clearTimeout()` χρησιμοποιεί τη μεταβλητή που επέστρεψε από το `setTimeout()`:

```
myVar = setTimeout(function, milliseconds);
clearTimeout(myVar);
```



Εάν η λειτουργία δεν έχει ήδη εκτελεστεί, μπορείτε να σταματήσετε την εκτέλεση καλώντας τη μέθοδο `clearTimeout()`:

#### Example

Same example as above, but with an added "Stop" button:

```
<button onclick="myVar = setTimeout(myFunction, 3000)">Try it</button>  
<button onclick="clearTimeout(myVar)">Stop it</button>
```

Εικόνα 18 – Ορισμός χρονικού ορίου (Πηγή: [https://www.w3schools.com/js/js\\_timing.asp](https://www.w3schools.com/js/js_timing.asp))

## Η μέθοδος `setInterval()`

Η μέθοδος `setInterval()` επαναλαμβάνει μια δοσμένη συνάρτηση σε κάθε δεδομένο χρονικό διάστημα.

```
window.setInterval(function, milliseconds);
```

Η μέθοδος `window.alert()` μπορεί να γραφτεί χωρίς το πρόθεμα `window`.

Η πρώτη παράμετρος είναι μια συνάρτηση που θα εκτελεστεί.

Η δεύτερη παράμετρος υποδεικνύει τη διάρκεια του χρονικού διαστήματος μεταξύ κάθε εκτέλεσης.

Αυτό το παράδειγμα εκτελεί μια λειτουργία που ονομάζεται "myTimer" μία φορά κάθε δευτερόλεπτο (όπως ένα ψηφιακό ρολόι).

## Example

Display the current time:

```
setInterval(myTimer, 1000);  
  
function myTimer() {  
  const d = new Date();  
  document.getElementById("demo").innerHTML = d.toLocaleTimeString();  
}
```

Εικόνα 20 – `setInterval` (Πηγή: [https://www.w3schools.com/js/js\\_timing.asp](https://www.w3schools.com/js/js_timing.asp))

## Πώς να σταματήσετε την εκτέλεση;

Η μέθοδος `clearInterval()` σταματά τις εκτελέσεις της συνάρτησης που καθορίζεται στη μέθοδο `setInterval()`.

```
window.clearInterval(timerVariable)
```

Η μέθοδος `window.alert()` μπορεί να γραφτεί χωρίς το πρόθεμα `window`.

Η μέθοδος `clearInterval()` χρησιμοποιεί τη μεταβλητή που επέστρεψε από το `setInterval()`:

```
let myVar = setInterval(function, milliseconds);  
clearInterval(myVar);
```

## Example

Same example as above, but we have added a "Stop time" button:

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

<button onclick="clearInterval(myVar)">Stop time</button>

<script>
let myVar = setInterval(myTimer, 1000);
function myTimer() {
  const d = new Date();
  document.getElementById("demo").innerHTML = d.toLocaleTimeString();
}
</script>
```

Εικόνα 21 – Καθαρό διάστημα/clear interval (Πηγή: [https://www.w3schools.com/js/js\\_timing.asp](https://www.w3schools.com/js/js_timing.asp))

## JavaScript Cookies

Τα cookies σάς επιτρέπουν να αποθηκεύετε πληροφορίες χρηστών σε ιστοσελίδες.

### Τι είναι τα cookies;

Τα cookies είναι δεδομένα, αποθηκευμένα σε μικρά αρχεία κειμένου, στον υπολογιστή σας.

Όταν ένας διακομιστής ιστού έχει στείλει μια ιστοσελίδα σε ένα πρόγραμμα περιήγησης, η σύνδεση τερματίζεται και ο διακομιστής ξεχνά τα πάντα για τον χρήστη.

Τα cookies εφευρέθηκαν για την επίλυση του προβλήματος «πώς να θυμάστε πληροφορίες σχετικά με τον χρήστη»:

- Όταν ένας χρήστης επισκέπτεται μια ιστοσελίδα, το όνομά του/της μπορεί να αποθηκευτεί σε ένα cookie.
- Την επόμενη φορά που ο χρήστης επισκέπτεται τη σελίδα, το cookie "θυμάται" το όνομά του/της.

Τα cookies αποθηκεύονται σε ζεύγη ονομάτων-τιμής όπως:

```
username = John Doe
```

Όταν ένα πρόγραμμα περιήγησης ζητά μια ιστοσελίδα από έναν διακομιστή, τα cookies που ανήκουν στη σελίδα προστίθενται στο αίτημα. Με αυτόν τον τρόπο ο διακομιστής παίρνει τα απαραίτητα δεδομένα για να "θυμηθεί" πληροφορίες σχετικά με τους χρήστες.

## Δημιουργία cookie με JavaScript

Η JavaScript μπορεί να δημιουργήσει, να διαβάσει και να διαγράψει cookies με την ιδιότητα `document.cookie`.

Με το JavaScript, ένα cookie μπορεί να δημιουργηθεί ως εξής:

```
document.cookie = "username=John Doe";
```

Μπορείτε επίσης να προσθέσετε μια ημερομηνία λήξης (σε ώρα UTC). Από προεπιλογή, το cookie διαγράφεται όταν το πρόγραμμα περιήγησης είναι κλειστό:

```
document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC";
```

Με μια παράμετρο `path`, μπορείτε να πείτε στο πρόγραμμα περιήγησης σε ποιο `path` ανήκει το cookie. Από προεπιλογή, το cookie ανήκει στην τρέχουσα σελίδα.

```
document.cookie = "username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC; path=/";
```

## Διαβάστε ένα Cookie με JavaScript

Με το JavaScript, τα cookies μπορούν να διαβαστούν ως εξής:

```
let x = document.cookie;
```

Η ιδιότητα `document.cookie` θα επιστρέψει όλα τα cookies σε μια συμβολοσειρά όπως: `cookie1=value; cookie2=value; cookie3=value;`

## Αλλαγή cookie με JavaScript

Με το JavaScript, μπορείτε να αλλάξετε ένα cookie με τον ίδιο τρόπο που το δημιουργείτε:

```
document.cookie = "username=John Smith; expires=Thu, 18 Dec 2013 12:00:00 UTC; path="/;
```

Το παλιό cookie έχει αντικατασταθεί.

## Διαγραφή cookie με JavaScript

Η διαγραφή ενός cookie είναι πολύ απλή.

Δεν χρειάζεται να καθορίσετε μια τιμή cookie όταν διαγράφετε ένα cookie.

Απλά ορίστε την παράμετρο λήξης σε μια προηγούμενη ημερομηνία:

```
document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 UTC; path="/;
```

Θα πρέπει να ορίσετε τη διαδρομή cookie για να βεβαιωθείτε ότι διαγράφετε το σωστό cookie. Ορισμένα προγράμματα περιήγησης δεν θα σας επιτρέψουν να διαγράψετε ένα cookie αν δεν καθορίσετε τη διαδρομή.



## Η συμβολοσειρά cookie

Η ιδιότητα `document.cookie` μοιάζει με κανονική συμβολοσειρά κειμένου. Αλλά δεν είναι.

Ακόμα κι αν γράψετε μια ολόκληρη συμβολοσειρά `cookie` στο `document.cookie`, όταν το διαβάσετε ξανά, μπορείτε να δείτε μόνο το ζεύγος του ονόματος-τιμής του.

Εάν ορίσετε ένα νέο `cookie`, τα παλαιότερα `cookies` δεν αντικαθίστανται. Το νέο `cookie` προστίθεται στο `document.cookie`, οπότε αν το ξαναδιαβάσετε θα λάβετε κάτι σαν:

```
cookie1 = value; cookie2 = value;
```

Αν θέλετε να βρείτε την τιμή ενός συγκεκριμένου `cookie`, πρέπει να γράψετε μια λειτουργία JavaScript που αναζητά την τιμή `cookie` στη συμβολοσειρά `cookie`.

## Παράδειγμα JavaScript Cookie

Στο παράδειγμα που θα ακολουθήσουμε, θα δημιουργήσουμε ένα `cookie` που αποθηκεύει το όνομα ενός επισκέπτη.

Την πρώτη φορά που ένας επισκέπτης φτάνει στην ιστοσελίδα, θα του ζητηθεί να συμπληρώσει το όνομά του. Το όνομα αποθηκεύεται στη συνέχεια σε ένα `cookie`.

Την επόμενη φορά που ο επισκέπτης θα φτάσει στην ίδια σελίδα, θα λάβει ένα μήνυμα καλωσορίσματος.

Για παράδειγμα θα δημιουργήσουμε 3 λειτουργίες JavaScript:

1. Μια συνάρτηση για να ορίσετε μια τιμή `cookie`
2. Μια συνάρτηση για τη λήψη μιας τιμής `cookie`
3. Μια συνάρτηση για τον έλεγχο μιας τιμής `cookie`



## Μια συνάρτηση για να ορίσετε μια τιμή cookie

Πρώτον, δημιουργούμε μια συνάρτηση που αποθηκεύει το όνομα του επισκέπτη σε μια μεταβλητή cookie:

```
function setCookie(cname, cvalue, exdays) {  
  const d = new Date();  
  d.setTime(d.getTime() + (exdays*24*60*60*1000));  
  let expires = "expires="+ d.toUTCString();  
  document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";  
}
```

Εικόνα 22 – Ορισμός συνάρτησης cookie (Πηγή: [https://www.w3schools.com/js/js\\_cookies.asp](https://www.w3schools.com/js/js_cookies.asp))

Επεξήγηση παραδείγματος:

Οι παράμετροι της συνάρτησης πιο πάνω είναι το όνομα του cookie (cname), η τιμή του cookie (cvalue) και ο αριθμός των ημερών μέχρι τη λήξη του cookie (exdays).

Η συνάρτηση ορίζει ένα cookie προσθέτοντας μαζί το όνομα cookiename, την τιμή cookie και τη συμβολοσειρά λήξης (expire string).

## Μια συνάρτηση για τη λήψη μιας τιμής cookie

Στη συνέχεια, δημιουργούμε μια συνάρτηση που επιστρέφει την τιμή ενός καθορισμένου cookie:

```
function getCookie(cname) {
  let name = cname + "=";
  let decodedCookie = decodeURIComponent(document.cookie);
  let ca = decodedCookie.split(';');
  for(let i = 0; i <ca.length; i++) {
    let c = ca[i];
    while (c.charAt(0) == ' ') {
      c = c.substring(1);
    }
    if (c.indexOf(name) == 0) {
      return c.substring(name.length, c.length);
    }
  }
  return "";
}
```

Εικόνα 22 – Ορισμός συνάρτησης cookie (Πηγή: [https://www.w3schools.com/js/js\\_cookies.asp](https://www.w3schools.com/js/js_cookies.asp))

Επεξήγηση της συνάρτησης:

Λάβετε το όνομα cookiename ως παράμετρο (cname).

Δημιουργήστε μια μεταβλητή (όνομα) με το κείμενο για αναζήτηση (cname + "=").

Αποκωδικοποιήστε τη συμβολοσειρά cookie, για να χειριστείτε τα cookies με ειδικούς χαρακτήρες, π.χ. '\$'

Διαχωρίστε το document.cookie σε ερωτηματικά σε έναν πίνακα που ονομάζεται ca (ca =decodedCookie.split(';')).

Διασχίστε τη συστοιχία ca (i = 0; i < ca.length; i++) και διαβάστε κάθε τιμή c = ca[i]).

Αν βρεθεί το cookie (c.indexOf (name) == 0), επιστρέψτε την τιμή του cookie (c.substring(name.length, c.length).

Εάν το cookie δεν βρεθεί, επιστρέψτε "".

## Μια συνάρτηση για τον έλεγχο μιας τιμής cookie

Τέλος, δημιουργούμε τη λειτουργία που ελέγχει αν έχει οριστεί ένα cookie.

Εάν έχει οριστεί το cookie, θα εμφανιστεί ένας χαιρετισμός (greeting).

Εάν το cookie δεν έχει οριστεί, θα εμφανιστεί ένα πλαίσιο προτροπής, ζητώντας το όνομα του χρήστη, και αποθηκεύει το cookie του ονόματος χρήστη για 365 ημέρες, καλώντας τη λειτουργία setCookie:

```
function checkCookie() {  
  let username = getCookie("username");  
  if (username != "") {  
    alert("Welcome again " + username);  
  } else {  
    username = prompt("Please enter your name:", "");  
    if (username != "" && username != null) {  
      setCookie("username", username, 365);  
    }  
  }  
}
```

Εικόνα 22 – Ορισμός συνάρτησης cookie (Πηγή: [https://www.w3schools.com/js/js\\_cookies.asp](https://www.w3schools.com/js/js_cookies.asp))