



Co-funded by the  
Erasmus+ Programme  
of the European Union



# Εκπαιδευτικό Υλικό JavaScript Υποενότητα – Advanced JavaScript

WP3: Εκπαιδευτικό Υλικό του Έργου Code4SP

Εκπονήθηκε από:



CITIZENS  
IN POWER



Center for Social  
Innovation



Co-funded by the  
Erasmus+ Programme  
of the European Union





# Προηγμένες Λειτουργίες της JavaScript



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



## Ημερομηνία και Ώρα JavaScript (Date και Time)

Με την JavaScript μπορείτε να δημιουργήσουμε ένα νέο τύπο αντικειμένου Date προσθέτοντας την αναπαράσταση της συμβολοσειράς μιας ημερομηνίας ορίζοντας τη μέθοδο κατασκευαστή Date() (constructor), ή προσθέτοντας τα επιμέρους στοιχεία μιας ημερομηνίας ως παραμέτρους στη μέθοδο κατασκευαστή Date() (χρόνος, μήνας, μέρα, ώρα, λεπτό, δευτερόλεπτο, χιλιοστό του δευτερολέπτου).

```
// Create a date object for today's date:  
const now = new Date();  
// Outputs today's date and time in standard format  
console.log(now);
```

Εξερευνήστε τις διάφορες δυνατότητες που παρέχει η μέθοδος κατασκευαστή:

[https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global\\_Objects/Date](https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/Date)



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



# Εκτέλεση Μαθηματικών Λειτουργιών JavaScript

Με την JavaScript μπορείτε να εκτελέσετε μαθηματικές λειτουργίες χρησιμοποιώντας αριθμητικούς τελεστές. Αυτό μπορεί να γίνει με την βοήθεια των τελεστών + (πρόσθεση), - (αφαίρεση), \* (πολλαπλασιασμός), / (διαίρεση) και % (όρισμα) operators.

Στο παράδειγμα πιο κάτω, χρησιμοποιούμε τον τελεστή της πρόσθεσης για να προσθέσουμε δύο αριθμούς:

```
var x = 10; var y = 5; var z = x + y;
```

Αυτή η πρόσθεση θα μας δώσει την τιμή z ως 15.

Εξερευνήστε τις διάφορες δυνατότητες που παρέχουν οι μαθηματικοί τελεστές JavaScript:

[https://www.w3schools.com/js/js\\_arithmetic.asp](https://www.w3schools.com/js/js_arithmetic.asp)



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



# Μετατροπές Τύπου JavaScript (Type Conversions)

Μπορείτε να μετατρέψετε μια τιμή σε ένα συγκεκριμένο τύπο ημερομηνίας χρησιμοποιώντας μεθόδους ενσωμάτωσης (built-in methods), όπως την μέθοδο `parseInt()` για την μετατροπή μιας τιμής σε ακέραιο αριθμό, ή την μέθοδο `parseFloat()` για την μετατροπή μιας τιμής σε έναν αριθμό κινητής υποδιαστολής (float).

Μπορείτε επίσης να χρησιμοποιήσετε τη μέθοδο `Number()` για να μετατρέψετε μια τιμή σε έναν αριθμό, ή την μέθοδο `Boolean()` για να μετατρέψετε μια τιμή σε έναν δυαδικό αριθμό (boolean).

Στο παράδειγμα που ακολουθεί, θα μετατρέψουμε μια συμβολοσειρά σε έναν αριθμό χρησιμοποιώντας τη μέθοδο `Number()`:

```
var x = "100"; var y = Number(x); console.log(y); // 100.
```

Εξερευνήστε τις διάφορες δυνατότητες που προσφέρουν οι Μετατροπές Τύπου:

[https://www.w3schools.com/js/js\\_type\\_conversion.asp](https://www.w3schools.com/js/js_type_conversion.asp)



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



# Συμβάντα Listeners JavaScript

Το συμβάντα Listeners της JavaScript επιτρέπουν τον ορισμό λειτουργιών που θα εκτελεστούν όταν εμφανίζεται ένα συγκεκριμένο συμβάν σε ένα αντικείμενο στην DOM.

Υπάρχει ένας αριθμός διαφορετικών συμβάντων που εμφανίζονται σε ένα αντικείμενο DOM, όπως ένα αντικείμενο το οποίο είναι πατημένο για υποβολή, όταν το ποντίκι είναι από πάνω του, ή ακόμη όταν τα περιεχόμενα ενός αντικειμένου αλλάζουν.

Για να προσθέσετε ένα συμβάν listener σε ένα στοιχείο, πρώτα πρέπει να επιλέξετε ένα στοιχείο χρησιμοποιώντας μια από τις μεθόδους επιλογής DOM, όπως `document.querySelector()` ή `document.getElementById()`.

Όταν επιλέγετε ένα στοιχείο, μπορείτε να χρησιμοποιήσετε τη μέθοδο `addEventListener()` για να προσαρτήσετε ένα συμβάν listener στο στοιχείο.

Η μέθοδος `addEventListener()` λαμβάνει δύο παραμέτρους: το όνομα του συμβάντος listener, και μια λειτουργία που θα εκτελεστεί όταν το συμβάν εμφανιστεί.

Για παράδειγμα, για να προσθέσετε ένα συμβάν listener σε ένα στοιχείο κουμπιού υποβολής (button element), θα πρέπει να χρησιμοποιήσετε τον ακόλουθο κώδικα:

```
button.addEventListener ( "click" , function ( ) { console.log ( "The button was clicked!" ) ; } );
```

Εξερευνήστε τις διάφορες δυνατότητες με την λειτουργία Συμβάντος Listener:

[https://www.w3schools.com/js/js\\_htmlDOM\\_eventlistener.asp](https://www.w3schools.com/js/js_htmlDOM_eventlistener.asp)



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



# Διάδοση Συμβάντων ενός Στοιχείου στην JavaScript (Event Propagation)

Όταν ένα συμβάν εμφανίζεται σε ένα στοιχείο, αυτό αναγνωρίζεται από την JavaScript και γίνεται επενέργεια σ' αυτό.

Για παράδειγμα, εάν θέλετε να δημιουργήσετε ένα αντικείμενο στην ιστοσελίδα σας στο οποίο οι χρήστες θα μπορούν να έχουν πρόσβαση κάνοντας κλικ στο εικονίδιό του, μπορείτε να προγραμματίσετε την JavaScript με τέτοιο τρόπο ούτως ώστε να αναγνωρίζει και να ενεργοποιεί το συμβάν click.

Όταν κάνετε κλικ σε ένα στοιχείο ενεργοποιείται στο συμβάν click.

Το συμβάν click αναπαράγει το δέντρο του DOM.

Αυτό σημαίνει ότι εάν υπάρχει ένας διαχειριστής συμβάντων (event handler) σε ένα γονικό στοιχείο (parent element), τότε αυτός θα ενεργοποιηθεί.

Με τον ίδιο τρόπο, εάν υπάρχει ένας διαχειριστής συμβάντων σε ένα στοιχείο grandparent, τότε αυτός πάλι θα ενεργοποιηθεί.

Αυτό αναπαράγει το δέντρο DOM μέχρι το συμβάν να φτάσει στο στοιχείο ρίζας (root element) ή μέχρι το συμβάν να τελειώσει.

Μερικά παραδείγματα της Διάδοσης Συμβάντων σε ένα στοιχείο:

<https://developer.mozilla.org/en-US/docs/Web/API/Event/stopPropagation>



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



# Μέθοδοι Δανεισμού στην JavaScript (Borrowing Methods)

Θα χρησιμοποιήσουμε ένα παράδειγμα ενός αντικειμένου το οποίο έχει μια ιδιότητα την οποία θέλουμε να δανειστούμε.

```
var myObject = {  
  someProperty: "foo",  
};
```

Μπορούμε να δανειστούμε την ιδιότητα `someProperty` από το `myObject`, όπως παρουσιάζεται πιο κάτω:

```
var myProperty = myObject.someProperty;
```

Η ιδιότητα `myProperty` θα λάβει τώρα την τιμή "foo".

Για περισσότερες πληροφορίες αναφορικά με τις Μεθόδους Δανεισμού :

<https://www.tutorialrepublic.com/javascript-tutorial/javascript-borrowing-methods.php>



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





# Ανύψωση JavaScript (Hoisting)

Η ανύψωση είναι μια συμπεριφορά πολύ ιδιαίτερη της JavaScript. Αποτελεί έναν όρο ο οποίος συνήθως συγχέεται ακόμα και από τους σχεδιαστές ιστοσελίδων. Για να κατανοήσουμε την συμπεριφορά αυτή, θα πρέπει να κατανοήσουμε τον όρο «ανύψωση».

Η ανύψωση είναι μια προεπιλεγμένη συμπεριφορά της JavaScript για την μετακίνηση όλων των δηλώσεων στην κορυφή του τρέχοντος πεδίου εφαρμογής (στην κορυφή της τρέχουσας δέσμης ενεργειών ή στην τρέχουσα συνάρτηση).

Παρατηρήστε τον ακόλουθο κώδικα:

```
var foo = 1; function bar() { if (!foo) { var foo = 10; } console.log(foo); } bar();
```

Τί νομίζετε ότι θα είναι το αποτέλεσμα αυτού του κώδικα;

Αν υπολογίσατε ότι το αποτέλεσμα θα είναι 10, τότε λανθάνετε. Το αποτέλεσμα αυτού του κώδικα είναι στην πραγματικότητα 1.

Αυτό γιατί η JavaScript ανυψώνει μόνο δηλώσεις (declaration) και όχι αρχικοποιήσεις (initialization). Όταν λοιπόν εκτελείται μια δήλωση if, η τιμή foo είναι απροσδιόριστη (undefined) και επομένως γίνεται 10.

Αυτό μπορεί να προκαλεί συνήθως σύγχυση, αλλά είναι σημαντικό να κατανοούμε πώς λειτουργεί η συμπεριφορά ανύψωσης στην JavaScript. Αυτό θα σας βοηθήσει να αποφεύγετε κοινά λάθη.

Για περισσότερες πληροφορίες αναφορικά με την Ανύψωση:

[https://www.w3schools.com/js/js\\_hoisting.asp](https://www.w3schools.com/js/js_hoisting.asp)



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



# Κλείσιμο της JavaScript (Function Closures)

Η λειτουργία κλεισίματος της JavaScript είναι μια ενσωματωμένη λειτουργία η οποία έχει πρόσβαση σε μεταβλητές που ορίζονται στο εσωτερικό ή εκτός μιας συνάρτησης. Η λειτουργία κλεισίματος μπορεί να έχει συνολικά πρόσβαση 1. σε μεταβλητές που ορίζονται στο εσωτερικό μιας συνάρτησης (δηλαδή μεταξύ των αγκύλων), 2. σε μεταβλητές που ορίζονται εκτός μιας συνάρτησης και 3. σε καθολικές μεταβλητές.

Ένα κλείσιμο της JavaScript μπορεί να προκύψει όταν μια συνάρτηση ορίζεται μέσα σε μια άλλη συνάρτηση και η εσωτερική αναφέρεται σε μεταβλητές της εξωτερικής. Τα κλεισίματα χρησιμοποιούνται εκτενώς στις βιβλιοθήκες της JavaScript όπως το jQuery.

Πιο κάτω παρατίθεται ένα παράδειγμα κλεισίματος της JavaScript:

```
function outerFunction(x) {  
  var innerFunction = function(y) {  
    return x + y;  
  }  
  return innerFunction;  
}
```

```
var add5 = outerFunction(5);  
var add10 = outerFunction(10);
```

```
console.log(add5(2)); // 7  
console.log(add10(2)); // 12
```

Για περισσότερες πληροφορίες αναφορικά με τα κλεισίματα της JavaScript:

[https://www.w3schools.com/js/js\\_function\\_closures.asp](https://www.w3schools.com/js/js_function_closures.asp)



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



# Διαχείριση Σφαλμάτων JavaScript (Error Handling)

Η JavaScript είναι μια πολύ ευέλικτη γλώσσα προγραμματισμού, που σημαίνει ότι υπάρχουν πολλοί τρόποι εγγραφής ενός κώδικα. Αυτή όμως η ευελιξία που παρέχει η JavaScript ως προς την εγγραφή κωδίκων μπορεί να οδηγήσει σε σφάλματα εάν δεν προσέχετε πως να συντάσσεται την δομή τους.

Ένας κοινός τρόπος με τον οποίο μπορείτε να αντιμετωπίσετε τυχόν σφάλματα στην εγγραφή κώδικα είναι η χρήση των μπλοκ try/catch. Αυτά τα μπλοκ σας επιτρέπουν να «δοκιμάσετε» έναν κώδικα και να «αρπάξετε» οποιοδήποτε σφάλμα εμφανιστεί κατά την ώρα της εκτέλεσης του. Η σύνταξη για τα μπλοκ try/catch πρέπει να έχει ως εξής:

```
try { // Code to try goes here } catch (error) { // Code to handle errors goes here }
```

Ο κώδικας μέσα στον μπλοκ try θα εκτελεστεί πρώτος. Αν δεν εμφανιστούν σφάλματα, τότε ο κώδικας μέσα στο μπλοκ catch δεν θα εκτελεστεί ποτέ. Ωστόσο, εάν εμφανιστεί κάποιο σφάλμα, τότε η εκτέλεση θα μεταβεί απευθείας στο μπλοκ catch και οποιοσδήποτε κώδικας προκύψει από το μπλοκ block θα παραλειφθεί.

Εξερευνήστε τις διάφορες δυνατότητες του Error Handling:

[https://www.w3schools.com/js/js\\_errors.asp](https://www.w3schools.com/js/js_errors.asp)



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



# Κανονικές Εκφράσεις JavaScript (Common Expressions)

Οι κανονικές εκφράσεις είναι ένα πολυδύναμο εργαλείο της JavaScript που χρησιμοποιείται συχνά με δύο μεθόδους συμβολοσειράς: `search ()` και `replace ()`, δηλαδή για την εκτέλεση των τύπων λειτουργιών αναζήτησης ή αντικατάστασης κειμένου.

Μια κανονική έκφραση είναι μια ακολουθία χαρακτήρων που σχηματίζει ένα πρότυπο αναζήτησης. Για παράδειγμα, αν χρησιμοποιηθεί η κανονική έκφραση `/A/` θα ταιριάξει με κάθε συμβολοσειρά που περιέχει ένα κεφαλαίο Α.

Μια κανονική έκφραση είναι απλά μια συμβολοσειρά η οποία περιέχει ένα μοτίβο το οποίο θέλετε να ταιριάξετε, το οποίο πρέπει να περικλείεται εντός δύο καθέτων ( `/` ). Για παράδειγμα:

```
const regex = /abc/;  
console.log(regex); // => /abc/
```

Εξερευνήστε τις δυνατότητες των Κανονικών Εκφράσεων JavaScript:

[https://www.w3schools.com/jsref/jsref\\_obj\\_regexp.asp](https://www.w3schools.com/jsref/jsref_obj_regexp.asp)



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



## Επικύρωση Φόρμας JavaScript (Form Validation)

Η JavaScript είναι μια γλώσσα προγραμματισμού από την πλευρά του πελάτη, που σημαίνει ότι η παραγωγή του τελικού περιεχομένου HTML πραγματοποιείται στο πρόγραμμα περιήγησης των επισκεπτών. Αυτό σας επιτρέπει να εκτελείτε λειτουργίες όπως π.χ. στην περίπτωση που θέλετε να ελέγξετε αν μια ηλεκτρονική διεύθυνση καταχωρήθηκε ορθά πριν αποσταλεί στον διακομιστή. Επίσης, αυτό μειώνει την πίεση στον διακομιστή επειδή τα δεδομένα φόρμας δεν είναι αναγκαίο να υποβληθούν μέχρι να γίνει ο έλεγχος σφαλμάτων από την JavaScript.

Εξερευνήστε τις διάφορες δυνατότητες της Επικύρωσης Φόρμας της JavaScript:

[https://www.w3schools.com/js/js\\_validation.asp](https://www.w3schools.com/js/js_validation.asp)



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



# JavaScript Cookies

Η δημιουργία ενός cookie στην JavaScript είναι πολύ απλή. Απλά πρέπει να χρησιμοποιήσετε ένα αντικείμενο εγγράφου και την μέθοδο `createCookie()`. Η σύνταξη αυτής της λειτουργίας θα πρέπει να έχει ως εξής:

```
document.cookie = "name=value; expires=date";
```

Η παράμετρος **όνομα** (name parameter) απευθύνεται στο όνομα του cookie που θέλετε να δημιουργήσετε, ενώ η παράμετρος **τιμή** αναφέρεται στο περιεχόμενο του (συμβολοσειρά). Εάν θέλετε να θέσετε μια ημερομηνία λήξης σ' αυτό, τότε πρέπει να προσθέσετε μια άλλη ιδιότητα, που ονομάζεται **expires** με μια έγκυρη συμβολοσειρά **Date()** ή απλά περάστε μια μηδενική τιμή null, εάν δεν θέλετε να έχει μια ημερομηνία λήξης.

Για περισσότερες πληροφορίες αναφορικά με τα Cookies:

[https://www.w3schools.com/js/js\\_cookies.asp](https://www.w3schools.com/js/js_cookies.asp)



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



# JavaScript Ajax : Αποστολή Αιτημάτων σε Server

Το Ajax είναι ένα σύνολο από Web development τεχνικές για τη δημιουργία διαδραστικών διαδικτυακών εφαρμογών. Στόχος του Ajax είναι η κατασκευή ιστοσελίδων που αποδίδουν πιο αποτελεσματικά το περιεχόμενό τους μέσω της ανταλλαγής μικρότερων όγκων δεδομένων με τους διακομιστές που λειτουργούν στο παρασκήνιο, ούτως ώστε ολόκληρη η σελίδα να μην χρειάζεται επαναφόρτωση κάθε φορά που ο χρήστης εκτελεί διαφορετικές ενέργειες πάνω σ' αυτή.

Αφού έχετε συμπεριλάβει στην ιστοσελίδα σας την βιβλιοθήκη Ajax, μπορείτε να ξεκινήσετε να στέλνετε αιτήματα στον διακομιστή. Για παράδειγμα, στην περίπτωση που θέλετε να φορτώσετε κάποια δεδομένα ενός αρχείου στον διακομιστή, θα πρέπει να χρησιμοποιήσετε τον ακόλουθο κώδικα:

```
$.ajax({ url: 'data.json', success: function(data) { // do something with the data } });
```

Αυτός ο κώδικας θα αποστείλει ένα αίτημα Ajax στο URL 'data.json'. **Εάν το αίτημα είναι επιτυχές, τότε θα εκτελεστεί η λειτουργία της απάντησης πίσω (callback) μαζί με τα δεδομένα από τον διακομιστή ως η παράμετρός του.** If the request is successful, the function in the 'success' callback will be executed with the data from the server as its argument.

Εξερευνήστε τις δυνατότητες της JavaScript Ajax:

[https://www.w3schools.com/xml/ajax\\_intro.asp](https://www.w3schools.com/xml/ajax_intro.asp)



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