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

code4sp
coding for social promotion

Εκπαιδευτικό Υλικό JavaScript
Υποκεφάλαιο 1 – Βασικές Γνώσεις JavaScript

WP3: Εκπαιδευτικό Υλικό του Έργου Code4SP

Εκπονητής:

Η υποστήριξη της Ευρωπαϊκής Επιτροπής για την έκδοση της παρούσας δημοσίευσης δε συνιστά αποδοχή του περιεχομένου της, το οποίο αντικατοπτρίζει μόνο τις απόψεις των συντακτών της, και η Επιτροπή δε φέρει καμία ευθύνη για οποιαδήποτε τυχόν χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου : 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

Υποκεφάλαιο 1: Βασικές Γνώσεις JavaScript

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

Τι είναι JavaScript?

- Η JavaScript (JS) είναι η πιο διαδεδομένη γλώσσα προγραμματισμού σεναρίων (scripting language) *από την πλευρά του πελάτη* (η γλώσσα προγραμματισμού σεναρίων, από την πλευρά του πελάτη, σχετίζεται με τα σενάρια που εκτελούνται μέσα σε ένα πρόγραμμα περιήγησης ιστού). Η JS προσθέτει διαδραστικότητα και δυναμικά εφέ στις ιστοσελίδες, επεξεργάζοντας το περιεχόμενο που επιστρέφεται από έναν διακομιστή ιστού.

- Η JavaScript είναι μια αντικειμενοστραφής γλώσσα, και έχει επίσης κάποιες ομοιότητες στη σύνταξη με τη γλώσσα προγραμματισμού Java, παρόλο που δεν σχετίζεται καθόλου με την Java.

Η JavaScript μπορεί να χρησιμοποιηθεί για ποικίλους σκοπούς, όπως:

- Για την τροποποίηση του περιεχομένου μιας ιστοσελίδας προσθέτοντας ή αφαιρώντας στοιχεία·
- Για την αλλαγή του στυλ και της θέση των στοιχείων σε μια ιστοσελίδα·
- Για την παρακολούθηση των ενεργειών, όπως το κλικ του ποντικιού, το κείμενο κατάδειξης (hover), κ.λπ. και την αντίδραση σε αυτά·
- Για την πραγματοποίηση και τον έλεγχο των μεταβάσεων και κινούμενων εικόνων·
- Για την δημιουργία αναδυόμενων παραθύρων ειδοποίησης (pop-up windows) για την εμφάνιση πληροφοριών ή προειδοποιητικών μηνυμάτων στον χρήστη.
- Για την ολοκλήρωση των λειτουργιών με βάση τα στοιχεία εισαγωγής από τον χρήστη και εμφάνιση των αποτελεσμάτων.
- Για την επικύρωση των στοιχείων εισαγωγής από τον χρήστη πριν τα υποβάλει στον διακομιστή.
- Και πολλούς άλλους ενδιαφέροντες λόγους που θα εξεταστούν αργότερα.

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

Ας ξεκινήσουμε λοιπόν με την JavaScript!

Ξεκινώντας από το σημείο αυτό, οι εκπαιδευόμενοι θα αντιληφθούν στην πορεία πόσο απλό μπορεί να είναι να προσθέσουν διαδραστικότητα σε μια ιστοσελίδα χρησιμοποιώντας την JavaScript.

Υπάρχουν 3 χαρακτηριστικοί τρόποι προσθήκης της JS σε μια ιστοσελίδα:

- **Ενσωμάτωση του κώδικα JavaScript μεταξύ μιας ετικέτας <script> και μιας ετικέτας </script> ;**
- **Η δημιουργία εξωτερικού αρχείου JavaScript με την επέκταση .js και στη συνέχεια τη φόρτωσή του μέσα στη σελίδα μέσω της ιδιότητας src της ετικέτας <script>.**
- **Η τοποθέτηση του κώδικα JavaScript απευθείας μέσα σε μια ετικέτα HTML χρησιμοποιώντας τα ειδικά χαρακτηριστικά ετικέτας όπως onclick , onmouseover, onkeypress, onload κ.λπ.**

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

Ας ξεκινήσουμε λοιπόν με την JavaScript

Ενσωμάτωση του κώδικα JavaScript μεταξύ μιας ετικέτας <script> και μιας ετικέτας </script> ;

Ο κώδικας της JavaScript μπορεί να ενσωματωθεί απευθείας μέσα σε μια ιστοσελίδα τοποθετώντας τον μεταξύ των ετικετών <script> και </script>. Η ετικέτα <script> υποδεικνύει στο πρόγραμμα περιήγησης ότι οι δηλώσεις που περιέχονται πρέπει να ερμηνεύονται ως εκτελέσιμο σενάριο και όχι ως HTML, όπως παρουσιάζεται στο παρακάτω παράδειγμα:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Embedding JavaScript</title>
</head>
<body>
    <script>
    var greet = "Hello World!";
    document.write(greet); // Prints: Hello World!
    </script>
</body>
</html>
```

Hello World!

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

Ας ξεκινήσουμε λοιπόν με την JavaScript!

Δημιουργία εξωτερικού αρχείου JavaScript με την επέκταση .js και στη συνέχεια η φόρτωσή του μέσα στη σελίδα μέσω της ιδιότητας src της ετικέτας <script>.

- Ένας κώδικας JavaScript μπορεί επίσης να τοποθετηθεί σε ένα ξεχωριστό αρχείο με επέκταση .js, που στη συνέχεια καλείται σε αυτό το αρχείο στο ίδιο έγγραφο μέσω του χαρακτηριστικού src της ετικέτας <script>, ως εξής:

<script src="js/hello.js"></script>

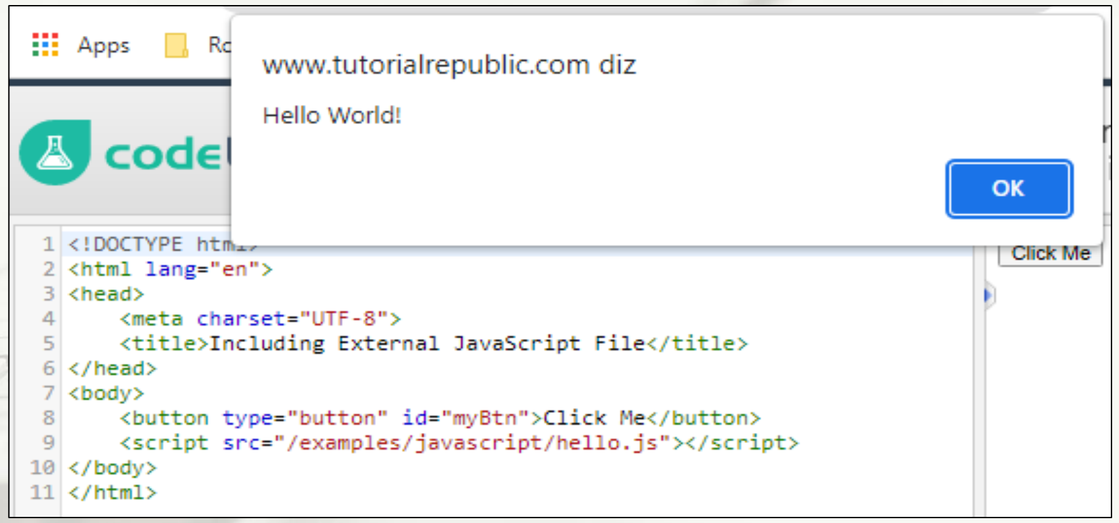
- Αυτός ο κώδικας μπορεί να φανεί ιδιαίτερα χρήσιμος εάν ο προγραμματιστής θέλει τα ίδια σενάρια να είναι διαθέσιμα σε πολλαπλά έγγραφα. Μετά από τη διαδικασία αυτή, θα αποφύγει να επαναλάβει την ίδια εργασία ξανά και ξανά, και αυτό καθιστά πολύ πιο απλή τη διαδικασία συντήρησης της ιστοσελίδας του/της.

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

Ας ξεκινήσουμε λοιπόν με την JavaScript!

Με την προώθηση της παραπάνω δήλωσης, θα δημιουργηθεί ένα αρχείο JavaScript με το όνομα "hello.js" και θα εισαχθεί σε αυτό ο ακόλουθος κώδικας:

```
1  // A function to display a message
2  function sayHello() {
3      alert("Hello World!");
4  }
5
6  // Call function on click of the button
7  document.getElementById("myBtn").onclick = sayHello;
```



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

Ας ξεκινήσουμε λοιπόν με την JavaScript!

Η τοποθέτηση του κώδικα JavaScript απευθείας μέσα σε μια ετικέτα HTML χρησιμοποιώντας τα ειδικά χαρακτηριστικά ετικέτας όπως onclick, onmouseover, onkeypress, onload κ.λπ.

Ο κώδικας JavaScript μπορεί να εισαχθεί ενσωματωμένος εισάγοντάς τον απευθείας μέσα στην ετικέτα HTML μέσω των ειδικών χαρακτηριστικών της ετικέτας όπως onclick, onmouseover, onkeypress, onload, κλπ.

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

Ας ξεκινήσουμε λοιπόν με την JavaScript

Παρ 'όλα αυτά, δεν συστήνεται η πρόσθηκη μεγάλου κώδικα JavaScript μέσα στον html κώδικα inline, καθώς μπορεί να προκαλέσει δυσλειτουργίες μεταξύ HTML με JavaScript, καθιστώντας δύσκολη τη διατήρηση του κώδικα JS. *Το Σχήμα 4* παρουσιάζει ένα παράδειγμα (στην περίπτωση αυτή, εμφανίζεται ένα μήνυμα ειδοποίησης όταν κάνετε κλικ στο στοιχείο κουμπιού):

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

Ας ξεκινήσουμε λοιπόν με την JavaScript!

- Το στοιχείο <script> μπορεί να τοποθετηθεί στο < head> ή στο τμήμα <body> ενός εγγράφου HTML. Ωστόσο, τα σενάρια θα πρέπει να τοποθετούνται κατά προτίμηση στο τέλος του τμήματος του σώματος, πριν από την </body> ετικέτα κλεισίματος.
- Αυτή η διαδικασία θα επιτρέψει στις ιστοσελίδες να φορτώθούν γρηγορότερα, δεδομένου ότι θα αποφευχθεί η παρεμπόδιση της απόκρισης της αρχικής σελίδας. Κάθε ετικέτα <script> αποκλείει τη διαδικασία απόκρισης μιας σελίδας μέχρι να κατεβάσει και να εκτελέσει πλήρως τον κώδικα JavaScript, οπότε η τοποθέτησή τους στο τμήμα της κεφαλίδας (δηλαδή στο < head> στοιχείο) του εγγράφου χωρίς κανένα έγκυρο λόγο θα επηρεάσει σημαντικά την απόκριση μιας ιστοσελίδας.

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

Ας ξεκινήσουμε λοιπόν με την JavaScript!

- **Υπάρχουν διαφορές μεταξύ της γλώσσας προγραμματισμού σεναρίων από την πλευρά του πελάτη και από την πλευρά του διακομιστή.**
- Οι γλώσσες σεναρίου από την πλευρά του πελάτη (π.χ. JavaScript ή VBScript) κατανοούνται και εκτελούνται από το πρόγραμμα περιήγησης ιστού, σε αντίθεση με τις γλώσσες σεναρίου από την πλευρά του διακομιστή (π.χ. PHP, ASP, Java, Python, Ruby κ.λπ.), οι οποίες εκτελούνται στον διακομιστή ιστού και η εξαγωγή τους επιστρέφεται πίσω στο πρόγραμμα περιήγησης ιστού σε μορφή HTML.

- Η γλώσσα σεναρίου από την πλευρά του πελάτη έχει πολλά πλεονεκτήματα σε σύγκριση με την κλασική γλώσσα σεναρίου από την πλευρά του διακομιστή. Για παράδειγμα, η JavaScript μπορεί να χρησιμοποιηθεί για να ελέγξει αν ο χρήστης έχει εισάγει μη έγκυρα δεδομένα σε πεδία φόρμας και να εμφανίσει συνεπώς ειδοποιήσεις για σφάλματα εισόδου σε πραγματικό χρόνο πριν από την υποβολή της φόρμας στο web-server, για την τελική επικύρωση και περαιτέρω επεξεργασία δεδομένων, προκειμένου να αποφευχθούν περιττές χρήσεις εύρους ζώνης δικτύου και η κακή χρήση των πόρων του συστήματος διακομιστή.

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

Σύνταξη κανόνων JavaScript

Η σύνταξη της JS είναι το **σύνολο των κανόνων** που περιλαμβάνουν ένα καλά δομημένο πρόγραμμα JavaScript. Το JS περιλαμβάνει δηλώσεις που τοποθετούνται εντός των ετικετών <script> και </script> HTML σε μια ιστοσελίδα ή εντός του εξωτερικού αρχείου JavaScript με επέκταση .js.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Example of JavaScript Statements</title>
</head>
<body>
    <script>
    var x = 5;
    var y = 10;
    var sum = x + y;
    document.write(sum); // Prints variable value
    </script>
</body>
</html>
```

```
15
```

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

Σύνταξη κανόνων JavaScript

- Οι εκπαιδευόμενοι θα πρέπει να δηλώσουν ότι η JavaScript έχει **διάκριση πεζών/κεφαλαίων**. Επομένως, οι μεταβλητές, οι λέξεις-κλειδιά, τα ονόματα των λειτουργιών και άλλα αναγνωριστικά πρέπει να δακτυλογραφούνται με συνέπεια όσον αφορά την κεφαλαιοποίηση των γραμμάτων. Για παράδειγμα, η μεταβλητή myVar πρέπει να πληκτρολογείται με αυτόν τον τρόπο (όχι "MYVAR", "myvar" κ.λπ.). Αυτό ισχύει για **όλες τις περιπτώσεις**.

- Επίσης, η JavaScript παρέχει επίσης τη δυνατότητα να γράψετε σχόλια σε όλες τις γραμμές κωδικοποίησης. Τα σχόλια εισάγονται κυρίως επειδή παρέχουν πρόσθετες πληροφορίες για τον πηγαίο κώδικα, αλλά και επειδή μπορούν να βοηθήσουν τους προγραμματιστές να κατανοήσουν τους κώδικες τους μετά από κάποιο χρονικό διάστημα, ομαδική εργασία κ.λπ.

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

Σύνταξη κανόνων JavaScript

Είναι δυνατή η προσθήκη σχολίων σε μία και πολλαπλές γραμμές στο JavaScript. Τα σχόλια μιας γραμμής ξεκινούν με μια διπλή κάθετο προς τα εμπρός (//), ακολουθούμενη από το κείμενο του σχολίου:

```
1  // This is my first JavaScript program
2  document.write("Hello World!");
```

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

JavaScript Syntax

Για ένα σχόλιο πολλαπλών γραμμών, μια κάθετος και ένας αστερίσκος (/*) είναι το σημείο εκκίνησης, που τελειώνει με έναν αστερίσκο και μια κάθετο (*/):

```
1  /* This is my first program
2  in JavaScript */
3  document.write("Hello World!");
```

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

Μεταβλητές JavaScript

- Για την αποθήκευση δεδομένων σε JavaScript, οι προγραμματιστές δημιουργούν μεταβλητές.
- Είναι το κλειδί για όλες τις γλώσσες προγραμματισμού και χρησιμοποιούνται για την αποθήκευση δεδομένων, για παράδειγμα με συμβολοσειρά κειμένου, αριθμούς ή άλλο/α στοιχείο.
- Όταν χρειαστεί, ο προγραμματιστής, μπορεί να ρυθμίσει, να ενημερώσει και να ανακτήσει δεδομένα ή τιμές που είναι αποθηκευμένες στις μεταβλητές. Οι μεταβλητές μπορούν να γίνουν κατανοητές ως συμβολικά ονόματα για τις τιμές.

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

Μεταβλητές JavaScript

- Μια μεταβλητή μπορεί να δημιουργηθεί χρησιμοποιώντας τη λέξη-κλειδί var, στην οποία ο τελεστής εκχώρησης (“=”) χρησιμοποιείται για την κατανομή τιμής σε μια μεταβλητή, ως εξής: `var varName = value`

```
1  var name = "Peter Parker";
2  var age = 21;
3  var isMarried = false;
```

3 μεταβλητές έχουν δημιουργηθεί

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

Μεταβλητές JavaScript

- Η τελευταία αναθεωρημένη έκδοση της JavaScript (ECMAScript 2015 ή ES6) εισάγει δύο νέες λέξεις-κλειδιά για τη δήλωση των μεταβλητών: let και const.
- Η λέξη κλειδί const λειτουργεί με τον ίδιο τρόπο όπως και το let. Ωστόσο, οι μεταβλητές που δηλώνονται με τη χρήση της const δεν μπορούν να ανατεθούν εκ νέου αργότερα στον κωδικό, ως εξής:

```
1  // Declaring variables
2  let name = "Harry Potter";
3  let age = 11;
4  let isStudent = true;
5
6  // Declaring constant
7  const PI = 3.14;
8  console.log(PI); // 3.14
9
10 // Trying to reassign
11 PI = 10; // error
```

Σε αντίθεση με τη λέξη-κλειδί var, η οποία δηλώνει μεταβλητές με εμβέλεια λειτουργίας στο σώμα μιας συνάρτησης (function-scoped variables), τόσο η λέξη-κλειδί let όσο και η λέξη-κλειδί const δηλώνουν μεταβλητές με εμβέλεια σε επίπεδο μπλοκ ({}). Η οριοθέτηση του πεδίου εφαρμογής κατά μπλοκ σημαίνει ότι δημιουργείται ένα νέο πεδίο εφαρμογής μεταξύ ενός ζεύγους αγκύλων.

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

Μεταβλητές JavaScript

Οι μεταβλητές JavaScript έχουν **συγκεκριμένους κανόνες για την ονομασία τους**:

- Ένα όνομα μεταβλητής πρέπει να ξεκινά με ένα γράμμα, με κάτω παύλα (_), ή το σύμβολο δολαρίου ($).
- Ένα όνομα μεταβλητής δεν μπορεί να ξεκινήσει με έναν αριθμό.
- Ένα μεταβλητό όνομα μπορεί να περιλαμβάνει μόνο αλφαριθμητικούς χαρακτήρες (A-Z, 0-9) και κάτω παύλες.
- Ένα όνομα μεταβλητής δεν μπορεί να κατανοήσει κενά.
- Ένα όνομα μεταβλητής δεν μπορεί να είναι μια λέξη-κλειδί JavaScript ή μια δεσμευμένη λέξη JavaScript (reserved words).

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

Παραγωγή Εξόδου JavaScript (Output)

- Υπάρχουν ορισμένες περιπτώσεις στις οποίες οι προγραμματιστές μπορεί να χρειαστεί να δημιουργήσουν εξόδους από τον κώδικα JS, π.χ., δείτε την τιμή της μεταβλητής, γράψτε ένα μήνυμα στην κονσόλα του προγράμματος περιήγησης, κλπ. Στην JavaScript, υπάρχουν μερικοί διαφορετικοί τρόποι παραγωγής εξόδου, συμπεριλαμβανομένης της εγγραφής εξόδου στο παράθυρο του προγράμματος περιήγησης ή στην κονσόλα του προγράμματος περιήγησης, της εμφάνισης εξόδου σε πλαίσια διαλόγου, της εγγραφής εξόδου σε στοιχείο HTML κ.λπ.

- Δεν είναι δύσκολη η εξαγωγή ενός μηνύματος ή η εγγραφή δεδομένων στην κονσόλα του προγράμματος περιήγησης (μπορείτε να έχετε πρόσβαση σε αυτό κάνοντας κλικ στο F12). Για τον σκοπό αυτό, θα πρέπει να εφαρμόζεται η console.log (), μια πολύ δυνατή αλλά και απλή μέθοδος.

- Για να γράψετε το περιεχόμενο σε ένα τρέχον έγγραφο, μόνο κατά την αποδόμησή του μπορεί να χρησιμοποιηθεί η μέθοδος document.write().

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

Παραγωγή Εξόδου JavaScript

- Εάν η μέθοδος document.write() χρησιμοποιηθεί μετά τη φόρτωση της σελίδας, θα αντικαταστήσει όλο το υφιστάμενο περιεχόμενο στο έγγραφο, όπως ακολουθεί στο σύνδεσμο αυτό.

- **Τα πλαίσια διαλόγου ειδοποίησης (alert dialog boxes)** μπορούν επίσης να προστεθούν για την εμφάνιση του μηνύματος ή των δεδομένων εξόδου στο χρήστη. Για τη δημιουργία ενός διαλόγου ειδοποιήσεων, χρησιμοποιείται η μέθοδος alert(), που έχει ως εξής:

```
1  // Displaying a simple text message
2  alert("Hello World!"); // Outputs: Hello World!
3
4  // Displaying a variable value
5  var x = 10;
6  var y = 20;
7  var sum = x + y;
8  alert(sum); // Outputs: 30
```

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

Παραγωγή Εξόδου JavaScript

- Οι έξοδοι μπορούν να εισαχθούν ή να γραφτούν μέσα σε ένα στοιχείο HTML χρησιμοποιώντας την ιδιότητα innerHTML. Παρ 'όλα αυτά, ο προγραμματιστής θα πρέπει να επιλέξει το στοιχείο πριν από την εγγραφή της εξόδου, χρησιμοποιώντας τη μέθοδο getElementById ().

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Writing into an HTML Element with JavaScript</title>
</head>
<body>
    <p id="greet"></p>
    <p id="result"></p>

    <script>
    // Writing text string inside an element
    document.getElementById("greet").innerHTML = "Hello World!";

    // Writing a variable value inside an element
    var x = 10;
    var y = 20;
    var sum = x + y;
    document.getElementById("result").innerHTML = sum;
    </script>
</body>
</html>
```

Hello World!

30

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

Τύποι Δεδομένων JavaScript

Οι τύποι δεδομένων ουσιαστικά ορίζουν τι είδους δεδομένα μπορούν να αποθηκευτούν και να επεξεργασθούν μέσα σε ένα πρόγραμμα. Υπάρχουν έξι βασικοί τύποι δεδομένων στην JS, οι οποίοι μπορούν να διακριθούν σε τρεις κύριες κατηγορίες:

- **Πρωτόγονοι τύποι δεδομένων (ή πρωτογενείς)** – Οι συμβολοσειρές, οι αριθμοί και τα δυαδικά είναι παραδείγματα πρωτόγονων τύπων δεδομένων, τα οποία μπορούν να περιέχουν μόνο μία τιμή κάθε φορά.
- **Σύνθετοι τύποι δεδομένων (ή τύποι δεδομένων αναφοράς/ reference)** – Το αντικείμενο, οι συστοιχίες και οι συναρτήσεις (που είναι όλοι οι τύποι αντικειμένων) είναι σύνθετοι τύποι δεδομένων. Αυτά μπορούν να περιέχουν συλλογές τιμών και πιο σύνθετων οντοτήτων · και
- **Ειδικοί τύποι δεδομένων** – οι ειδικοί τύποι δεδομένων είναι απροσδιόριστοι και μηδενικοί.

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

Τύποι Δεδομένων JavaScript

Ο τύπος δεδομένων συμβολοσειράς

Χρησιμοποιείται για την ενσωμάτωση κειμενικών δεδομένων (για παράδειγμα, ακολουθίες χαρακτήρων). Οι συμβολοσειρές δημιουργούνται χρησιμοποιώντας μονά ή διπλά εισαγωγικά που περιβάλλουν έναν ή περισσότερους χαρακτήρες:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript String Data Type</title>
</head>
<body>
    <script>
    // Creating variables
    var a = 'Hi there!';  // using single quotes
    var b = "Hi there!";  // using double quotes

    // Printing variable values
    document.write(a + "<br>");
    document.write(b);
    </script>
</body>
</html>
```

```
Hi there!
Hi there!
```

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

Τύποι Δεδομένων JavaScript

Ο τύπος αριθμητικών δεδομένων

Ο τύπος δεδομένων αριθμών είναι χρήσιμος για την εμφάνιση θετικών ή αρνητικών αριθμών με ή χωρίς δεκαδικό ψηφίο, ή αριθμών που γράφονται με εκθετική σημειογραφία, για παράδειγμα: 1.5e-4 (που ισοδυναμεί με $1.5x10^{-4}$)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Number Data Type</title>
</head>
<body>
    <script>
    // Creating variables
    var a = 25;
    var b = 80.5;
    var c = 4.25e+6;
    var d = 4.25e-6;

    // Printing variable values
    document.write(a + "<br>");
    document.write(b + "<br>");
    document.write(c + "<br>");
    document.write(d);
    </script>
</body>
</html>
```

```
25
80.5
4250000
0.00000425
```

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

Τύπος Δεδομένων JavaScript

Ο τύπος αριθμητικών δεδομένων

Ο τύπος αριθμητικών δεδομένων περιλαμβάνει επίσης ορισμένες ειδικές τιμές που είναι: Infinity, -Infinity και NaN. Το άπειρο αντιπροσωπεύει το μαθηματικό άπειρο (∞), το οποίο είναι μεγαλύτερο από οποιονδήποτε αριθμό. Το άπειρο είναι το αποτέλεσμα της διαίρεσης ενός μη μηδενικού αριθμού με το 0, όπως θα μπορούσε να ελεγχθεί στον πίνακα πιο κάτω:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Infinity</title>
</head>
<body>
    <script>
    document.write(16 / 0);
    document.write("<br>");
    document.write(-16 / 0);
    document.write("<br>");
    document.write(16 / -0);
    </script>
</body>
</html>
```

```
Infinity
-Infinity
-Infinity
```

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

Τύποι Δεδομένων JavaScript

Ο Τύπος Δεδομένων Αλήθειας ή Boolean

Δύο τιμές μπορούν να διατηρηθούν σε αυτόν τον τύπο δεδομένων: σωστό (true) ή λάθος (false). Χρησιμοποιείται κλασικά σε τιμές αποθέματος όπως "yes" (σωστό) ή "no" (λάθος), on (σωστό) ή off (λάθος) κ.λπ. ως εξής:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Boolean Data Type</title>
</head>
<body>
    <script>
    // Creating variables
    var isReading = true;   // yes, I'm reading
    var isSleeping = false; // no, I'm not sleeping

    // Printing variable values
    document.write(isReading + "<br>");
    document.write(isSleeping);
    </script>
</body>
</html>
```

```
true
false
```

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

Τύποι Δεδομένων JavaScript

Ο Αφηρημένος Τύπος Δεδομένων

Ο Τύπος Απροσδιόριστων Δεδομένων μπορεί να λάβει μόνο μία τιμή – η ειδική τιμή είναι απροσδιόριστη. Εάν μια μεταβλητή έχει δηλωθεί, αλλά δεν έχει οριστεί τιμή, η τιμή δηλώνεται ως απροσδιόριστη

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Undefined Data Type</title>
</head>
<body>
    <script>
    // Creating variables
    var a;
    var b = "Hello World!"

    // Printing variable values
    document.write(a + "<br>");
    document.write(b);
    </script>
</body>
</html>
```

```
undefined
Hello World!
```

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

Τύποι Δεδομένων JavaScript

Ο Τύπος Μηδενικών Δεδομένων

Αυτός είναι ένας ακόμα ειδικός τύπος δεδομένων που μπορεί να έχει μόνο μία τιμή – την μηδενική τιμή. Μια μηδενική τιμή σημαίνει ότι απλά δεν υπάρχει τιμή. Δεν είναι ισοδύναμη με μια κενή συμβολοσειρά ("") ή με το μηδέν, αλλά αποτελεί αμιγώς τίποτα.

Μια μεταβλητή μπορεί σαφώς να αδειάσει από το τρέχον περιεχόμενό της μέσω της εκχώρησης της **μηδενικής τιμής** σ' αυτήν.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Null Data Type</title>
</head>
<body>
    <script>
    var a = null;
    document.write(a + "<br>"); // Print: null

    var b = "Hello World!"
    document.write(b + "<br>"); // Print: Hello World!

    b = null;
    document.write(b) // Print: null
    </script>
</body>
</html>
```

```
null
Hello World!
null
```

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

Τύποι Δεδομένων JavaScript

Ο Τύπος Δεδομένων Αντικειμένου

Το αντικείμενο είναι ένας πολύπλευρος τύπος δεδομένων που επιτρέπει την αποθήκευση συλλογών δεδομένων.

Ένα αντικείμενο περιέχει ιδιότητες, που ορίζονται ως ζεύγος τιμής-κλειδιού. Μια διότητα key (όνομα) είναι πάντα μια συμβολοσειρά, αλλά η τιμή μπορεί να είναι οποιοσδήποτε τύπος δεδομένων (συμβολοσειρές, αριθμοί, δυαδικές τιμές, ή σύνθετοι τύποι δεδομένων όπως πίνακες, συναρτήσεις και άλλα αντικείμενα). Πιο κάτω παρατίθεται ο απλούστερος τρόπος για να δημιουργήσετε ένα αντικείμενο στο JavaScript:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Object Data Type</title>
</head>
<body>
    <script>
    var emptyObject = {};
    var person = {"name": "Clark", "surname": "Kent", "age": "36"};

    // For better reading
    var car = {
        "modal": "BMW X3",
        "color": "white",
        "doors": 5
    }

    // Print variables values in browser's console
    console.log(person);
    console.log(car);
    </script>
    <p><strong>Note:</strong> Check out the browser console by pressing the f12 key on the keyboard.</p>
</body>
</html>
```

Note: Check out the browser console by pressing the f12 key on the keyboard.

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

Τύποι Δεδομένων JavaScript

Ο τύπος δεδομένων συνάρτησης

Μια συνάρτηση είναι ένα αντικείμενο που μπορεί να κληθεί για τη δημιουργία ενός κώδικα μπλοκ. Οι συναρτήσεις είναι αντικείμενα, επομένως είναι δυνατό να προσθέσετε σ' αυτά μεταβλητές, όπως φαίνεται στο πιο κάτω παράδειγμα:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Function Data Type</title>
</head>
<body>
    <script>
    var greeting = function(){
        return "Hello World!";
    }

    // Check the type of greeting variable
    document.write(typeof greeting) // Output: function
    document.write("<br>");
    document.write(greeting());     // Output: Hello World!
    </script>
</body>
</html>
```

```
function
Hello World!
```

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

Τύποι Δεδομένων JavaScript

Ο τύπος του Τελεστή

Ο τύπος του Τελεστή είναι ως επί το πλείστων επωφελής για την επεξεργασία των τιμών διαφόρων τύπων με διαφορετικό τρόπο. Ωστόσο, ο προγραμματιστής θα πρέπει να είναι προσεκτικός, καθώς μπορεί να παράγει απρόβλεπτα αποτελέσματα σε ορισμένες περιπτώσεις:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript typeof Operator</title>
</head>
<body>
    <script>
    // Numbers
    document.write(typeof 15 + "<br>");  // Prints: "number"
    document.write(typeof 42.7 + "<br>");  // Prints: "number"
    document.write(typeof 2.5e-4 + "<br>");  // Prints: "number"
    document.write(typeof Infinity + "<br>");  // Prints: "number"
    document.write(typeof NaN + "<br>");  // Prints: "number". Despite being "Not-A-Number"

    // Strings
    document.write(typeof '' + "<br>");  // Prints: "string"
    document.write(typeof 'hello' + "<br>");  // Prints: "string"
    document.write(typeof '12' + "<br>");  // Prints: "string". Number within quotes is document.write(typeof string

    // Booleans
    document.write(typeof true + "<br>");  // Prints: "boolean"
    document.write(typeof false + "<br>");  // Prints: "boolean"

    // Undefined
    document.write(typeof undefined + "<br>");  // Prints: "undefined"
    document.write(typeof undeclaredVariable + "<br>"); // Prints: "undefined"

    // Null
    document.write(typeof Null + "<br>");  // Prints: "object"

    // Objects
    document.write(typeof {name: "John", age: 18} + "<br>");  // Prints: "object"

    // Arrays
    document.write(typeof [1, 2, 4] + "<br>");  // Prints: "object"

    // Functions
    document.write(typeof function(){});  // Prints: "function"
    </script>
</body>
</html>
```

```
number
number
number
number
number
string
string
string
boolean
boolean
undefined
undefined
undefined
object
object
function
```

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

Τελεστές JavaScript

- Οι Τελεστές JavaScript είναι σύμβολα ή λέξεις-κλειδιά που ενημερώνουν την μηχανή της JavaScript για να εκτελέσει μια δεδομένη ενέργεια. Για παράδειγμα, το σύμβολο add (+) είναι ένας τελεστής που λέει στον κινητήρα JavaScript να προσθέσει δύο μεταβλητές ή τιμές, ενώ τα σύμβολα equal-to (=), greater-than (>) ή less-than (<) είναι οι τελεστές που λένε στην μηχανή της JavaScript να συγκρίνει δύο μεταβλητές ή τιμές κ.λπ.
-
- Μεταξύ των διαφόρων τελεστών που χρησιμοποιούνται στην JavaScript, οι πρώτοι που θα περιγραφούν είναι οι **αριθμητικοί τελεστές** της JavaScript. Αυτοί οι αριθμητικοί τελεστές τίθενται σε λειτουργία προκειμένου να εκτελεστούν κοινές αριθμητικές πράξεις (πρόσθεση, αφαίρεση, πολλαπλασιασμός και ούτω καθεξής).

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

Τελεστές JavaScript

Αριθμητικοί Τελεστές JavaScript

Operator	Description	Example	Result
+	Πρόσθεση	x + y	Άθροισμα του x και y
-	Αφαίρεση	x - y	Διαφορά του x και y.
*	Πολλαπλασιασμός	x * y	Γινόμενο του x και y.
/	Διαίρεση	x / y	Πηλίκο του x και y
%	Όρισμα	x % y	Αποτέλεσμα του x διαιρούμενου με το y

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

Τελεστές JavaScript

Αριθμητικοί Τελεστές JS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Arithmetic Operators</title>
</head>
<body>
    <script>
    var x = 10;
    var y = 4;
    document.write(x + y); // Prints: 14
    document.write("<br>");

    document.write(x - y); // Prints: 6
    document.write("<br>");

    document.write(x * y); // Prints: 40
    document.write("<br>");

    document.write(x / y); // Prints: 2.5
    document.write("<br>");

    document.write(x % y); // Prints: 2
    </script>
</body>
</html>
```

```
14
6
40
2.5
2
```

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

Τελεστές JavaScript

Τελεστές Στοιχειοσειράς JS

Τελεστής	Περιγραφή	Παράδειγμα	Αποτέλεσμα
+	Συνένωση	str1 + str2	Συνένωση του str1 και του str2
+=	Τελεστής εκχώρησης συνένωσης	str1 += str2	Προσθέτει το str2 και το str1

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

Τελεστές JavaScript

Τελεστές Στοιχειοσειρών JS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript String Operators</title>
</head>
<body>
    <script>
    var str1 = "Hello";
    var str2 = " World!";

    document.write(str1 + str2 + "<br>"); // Outputs: Hello World!

    str1 += str2;
    document.write(str1); // Outputs: Hello World!
    </script>
</body>
</html>
```

Hello World!
Hello World!

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

Τελεστές JavaScript

Τελεστές Αύξησης και Μείωσης JS

Τελεστής	Όνομα	Αποτέλεσμα
++x	Πριν την αύξηση	Προσαυξήσεις του x κατά ένα, έπειτα επιστροφή του x
x++	Μετά την αύξηση	Επιστροφή του x, έπειτα επιστροφή του x κατά ένα
--x	Πριν την μείωση	Μείωση του x κατά ένα, έπειτα επιστροφή του x
x--	Μετά την μείωση	Επιστροφή του x, έπειτα μείωση του x κατά ένα

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

Τελεστές JavaScript

Τελεστές Αύξησης και Μείωσης JS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Incrementing and Decrementing Operators</title>
</head>
<body>
    <script>
    var x; // Declaring Variable

    x = 10;
    document.write(++x); // Prints: 11
    document.write("<p>" + x + "</p>");    // Prints: 11

    x = 10;
    document.write(x++); // Prints: 10
    document.write("<p>" + x + "</p>");    // Prints: 11

    x = 10;
    document.write(--x); // Prints: 9
    document.write("<p>" + x + "</p>");    // Prints: 9

    x = 10;
    document.write(x--); // Prints: 10
    document.write("<p>" + x + "</p>");    // Prints: 9
    </script>
</body>
</html>
```

```
11
11
10
11
9
9
10
9
```

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

Τελεστές JavaScript

Λογικοί Τελεστές JS

Τελεστής	Όνομα	Παράδειγμα	Αποτέλεσμα
&&	Και	x && y	Αληθές αν και το x και το y είναι αληθή
||	Ή	x || y	Αληθές εάν είτε το x ή το y είναι αληθή
!	Όχι	!x	Αληθές εάν το x είναι ψευδές

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

Τελεστές JavaScript

Λογικοί Τελεστές JS Logical

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Logical Operators</title>
</head>
<body>
    <script>
    var year = 2018;

    // Leap years are divisible by 400 or by 4 but not 100
    if((year % 400 == 0) || ((year % 100 != 0) && (year % 4 == 0))){
        document.write(year + " is a leap year.");
    } else{
        document.write(year + " is not a leap year.");
    }
    </script>
</body>
</html>
```

```
2018 is not a leap year.
```

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

Τελεστές JavaScript

JS Comparison Operators

Τελεστής	Όνομα	Παράδειγμα	Αποτέλεσμα
==	Ίσο	x == y	Αληθές αν το x είναι ίσο με το y
===	Πανομοιότυπο	x === y	Αληθές αν το είναι ίσο με το y, και είναι του ίδιου τύπου
!=	Άνισο	x != y	Αληθές αν το x δεν είναι ίσο με το y
!==	Μη πανομοιότυπο	x !== y	Αληθές αν το x δεν είναι ίσο με το y, ή δεν είναι του ίδιου τύπου
<	Μικρότερο από	x < y	Αληθές αν το x είναι μικρότερο του y
>	Μεγαλύτερο από	x > y	Αληθές αν το x είναι μεγαλύτερο του y
>=	Μεγαλύτερο ή ίσο από	x >= y	Αληθές αν το x είναι μεγαλύτερο ή ίσο με το y
<=	Μικρότερο από ή ίσο από	x <= y	Αληθές αν το x είναι μικρότερο ή ίσο από το y

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

Τελεστές JavaScript

Τελεστές Σύγκρισης JS

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>JavaScript Comparison Operators</title>
6  </head>
7  <body>
8      <script>
9      var x = 25;
10     var y = 35;
11     var z = "25";
12
13     document.write(x == z);  // Prints: true
14     document.write("<br>");
15
16     document.write(x === z); // Prints: false
17     document.write("<br>");
18
19     document.write(x != y);  // Prints: true
20     document.write("<br>");
21
22     document.write(x !== z); // Prints: true
23     document.write("<br>");
24
25     document.write(x < y);   // Prints: true
26     document.write("<br>");
27
28     document.write(x > y);   // Prints: false
29     document.write("<br>");
30
31     document.write(x <= y);  // Prints: true
32     document.write("<br>");
33
34     document.write(x >= y);  // Prints: false
35     </script>
36 </body>
37 </html>
```

```
true
false
true
true
true
false
true
false
```

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

Συμβάντα JavaScript

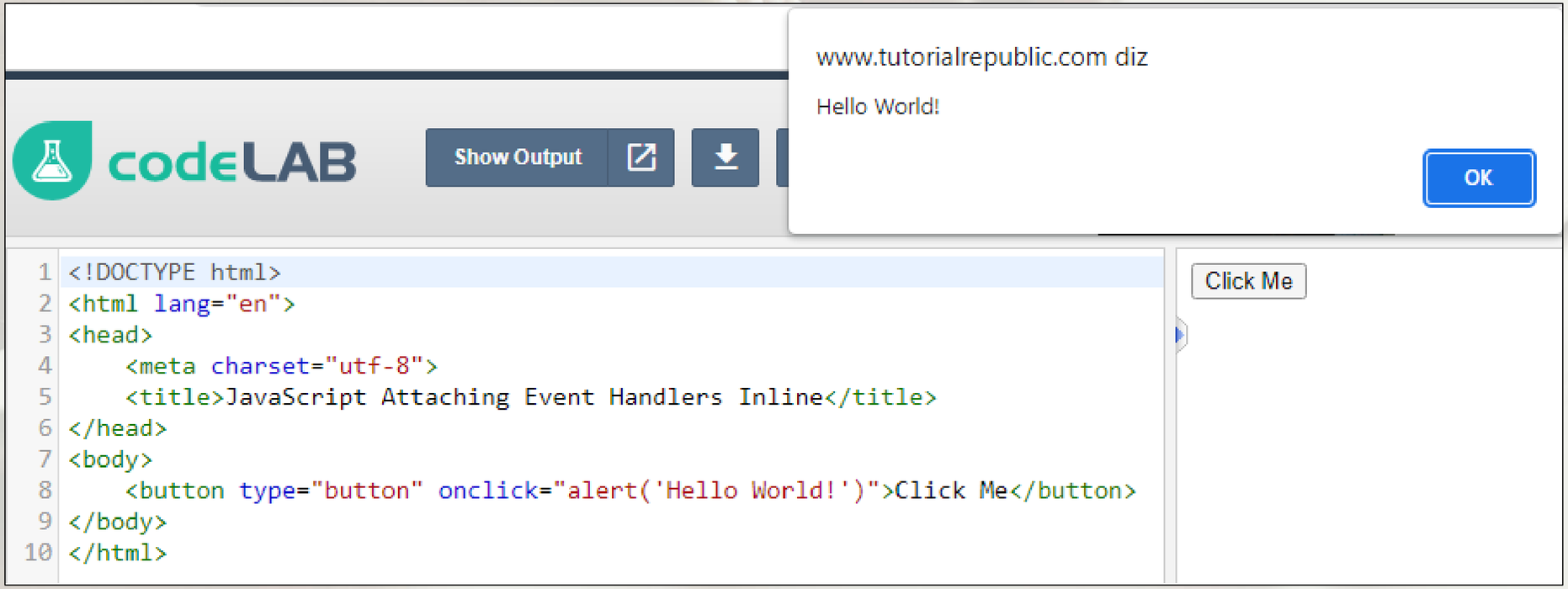
- Πριν μπούμε στα βαθιά του σημείου αυτού, είναι σημαντικό να αναγνωρίσουμε τι είναι ένα γεγονός στο πλαίσιο αυτό. Ένα συμβάν είναι κάτι που συμβαίνει κάθε φορά που οι χρήστες αλληλεπιδρούν με την ιστοσελίδα, όπως όταν ένας σύνδεσμος ή ένα κουμπί πατιέται, το κείμενο εισάγεται σε ένα πλαίσιο εισόδου ή περιοχή κειμένου, μια επιλογή γίνεται σε ένα πλαίσιο επιλογής, το πλήκτρο πατιέται στο πληκτρολόγιο, ο δείκτης του ποντικιού μετακινείται, μια φόρμα υποβάλλεται, και ούτω καθεξής. Κάθε τόσο, το πρόγραμμα περιήγησης είναι σε θέση να ενεργοποιήσει τα ίδια τα συμβάντα, για παράδειγμα κατά τη φόρτωση μιας σελίδας.

- Πριν μπούμε στα βαθιά του σημείου αυτού, είναι σημαντικό να αναγνωρίσουμε τι είναι ένα γεγονός στο πλαίσιο αυτό. Ένα συμβάν είναι κάτι που συμβαίνει κάθε φορά που οι χρήστες αλληλεπιδρούν με την ιστοσελίδα, όπως όταν ένας σύνδεσμος ή ένα κουμπί πατιέται, το κείμενο εισάγεται σε ένα πλαίσιο εισόδου ή περιοχή κειμένου, μια επιλογή γίνεται σε ένα πλαίσιο επιλογής, το πλήκτρο πατιέται στο πληκτρολόγιο, ο δείκτης του ποντικιού μετακινείται, μια φόρμα υποβάλλεται, και ούτω καθεξής. Κάθε τόσο, το πρόγραμμα περιήγησης είναι σε θέση να ενεργοποιήσει τα ίδια τα συμβάντα, για παράδειγμα κατά τη φόρτωση μιας σελίδας.

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

Συμβάντα JavaScript

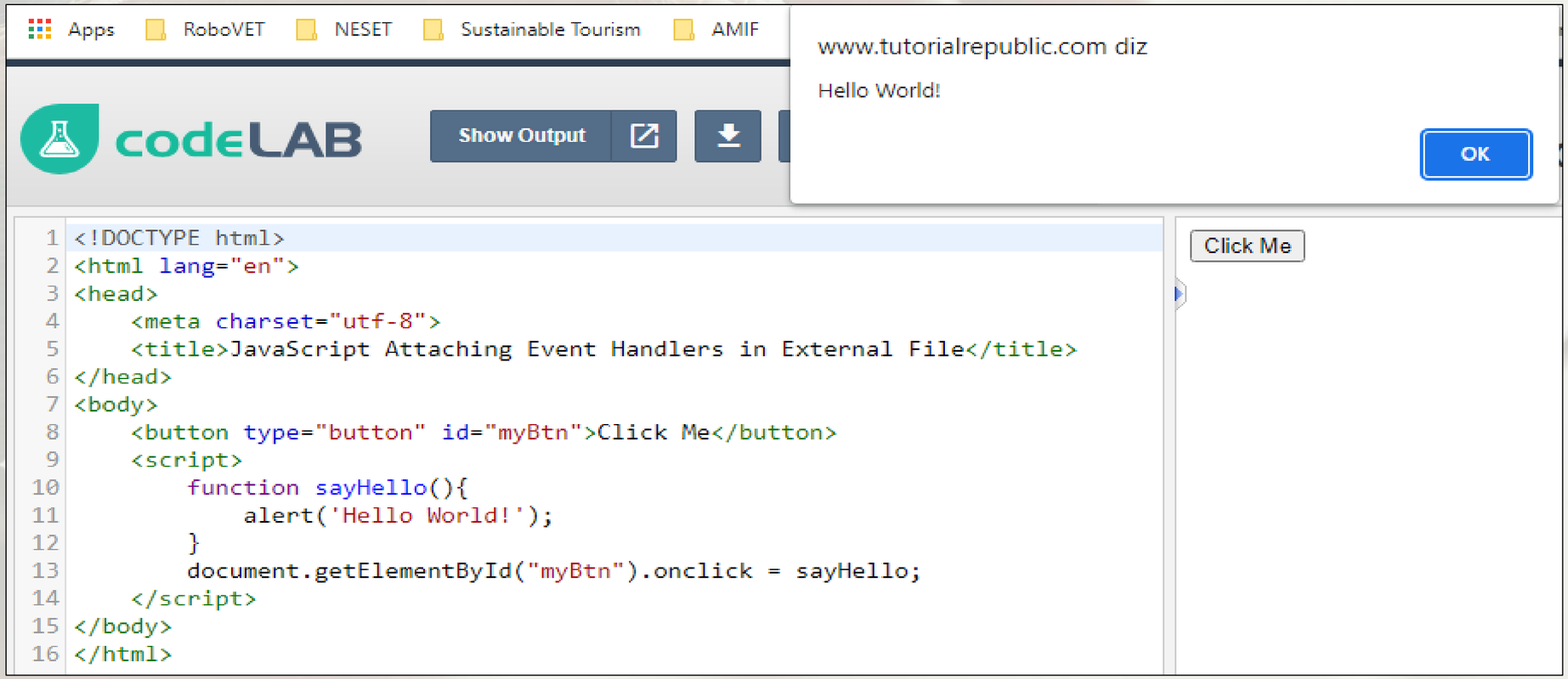
- Υπάρχουν πολλοί τρόποι ανάθεσης ενός χειριστή συμβάντων. Ο απλούστερος τρόπος είναι να τα προσθέσετε απευθείας στην ετικέτα έναρξης των στοιχείων HTML, μέσω των ειδικών χαρακτηριστικών χειριστή συμβάντων. Π.χ., για να αναθέσετε ένα χειριστή κλικαρίσματος για ένα στοιχείο κουμπιού, μπορεί να χρησιμοποιηθεί το χαρακτηριστικό onclick, ως εξής:



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

Συμβάντα JavaScript

- Παρ 'όλα αυτά, για να κρατήσουν την JavaScript αποκομμένη από την HTML, οι προγραμματιστές μπορούν να ρυθμίσουν τον χειριστή συμβάντων σε ένα εξωτερικό αρχείο JavaScript ή εντός των ετικετών < script > και</script >, ως εξής:



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

JavaScript Events

Γενικά, τα συμβάντα μπορούν να κατηγοριοποιηθούν σε τέσσερις κύριες ομάδες — **συμβάντα ποντικιού, συμβάντα πληκτρολογίου, συμβάντα φόρμας και συμβάντα εγγράφου/παραθύρου**.

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

Συμβάντα JavaScript

- **Συμβάντα ποντικού**

Ένα συμβάν ποντικιού ενεργοποιείται όταν ο χρήστης κάνει κλικ σε κάποιο στοιχείο, μετακινεί το δείκτη του ποντικιού πάνω σε ένα στοιχείο και ούτω καθεξής. Μερικά σημαντικά γεγονότα του ποντικιού και των χειριστών των συμβάντων τους είναι οι εξής:

- **Το συμβάν κλικ (onclick):** Το συμβάν κλικ συμβαίνει όταν ένας χρήστης κάνει κλικ σε ένα στοιχείο σε μια ιστοσελίδα. Συνήθως, πρόκειται για στοιχεία φόρμας και συνδέσμων. Μπορείτε να χειριστείτε ένα συμβάν κλικ με έναν χειριστή συμβάντων onclick.

  Το ακόλουθο παράδειγμα παρουσιάζει την περίπτωση ενός μηνύματος ειδοποίησης που εμφανίζεται όταν ο χρήστης κάνει κλικ στα στοιχεία.
- **Το Contextmenu Event (oncontextmenu):** συμβαίνει όταν οι χρήστες κάνουν κλικ στο δεξί κουμπί του ποντικιού σε ένα στοιχείο, ανοίγοντας ένα μενού περιεχομένων. Ο χειριστής συμβάντων Oncontextmenu χειρίζεται ένα συμβάν μενού συμφραζομένων. Αυτό το παράδειγμα εμφανίζει ένα μήνυμα ειδοποίησης όταν οι χρήστες κάνουν δεξί κλικ στα στοιχεία.
- **Το συμβάν του ποντικιού (onmouseover):** συμβαίνει όταν οι χρήστες μετακινούν το δείκτη του ποντικιού πάνω από ένα στοιχείο. Μπορεί να αντιμετωπιστεί με τον χειριστή συμβάντων onmouseover. Το ακόλουθο παράδειγμα εμφανίζει ένα μήνυμα ειδοποίησης όταν το ποντίκι τοποθετείται πάνω από τα στοιχεία.
- **Το συμβάν Mouseout (onmouseout):** λαμβάνει χώρα όταν οι χρήστες μετακινούν τον δείκτη του ποντικιού έξω από ένα στοιχείο. Μπορείτε να το χειριστείτε χρησιμοποιώντας τον χειριστή συμβάντων onmouseout. Το ακόλουθο παράδειγμα θα σας εμφανίσει ένα μήνυμα ειδοποίησης όταν πραγματοποιηθεί το συμβάν του mouseout.

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

Συμβάντα JavaScript

- **Συμβάντα πληκτρολογίου**

Ένα συμβάν πληκτρολογίου λαμβάνει χώρα όταν ο χρήστης πατήσει ή απελευθερώσει ένα πλήκτρο στο πληκτρολόγιο. Μερικά από τα πιο σημαντικά συμβάντα πληκτρολογίου και οι χειριστές των συμβάντων τους είναι τα εξής:

- **Το Συμβάν Keydown (onkeydown):** συμβαίνει όταν οι χρήστες πατούν ένα πλήκτρο στο πληκτρολόγιο. Μπορείτε να το χειριστείτε χρησιμοποιώντας τον χειριστή συμβάντων onmouseout. Αυτό το παράδειγμα εμφανίζει ένα μήνυμα ειδοποίησης κάθε φορά που πραγματοποιείται ένα συμβάν κλειδιού.
- **The Συμβάν Keyup (onkeyup):** συμβαίνει όταν οι χρήστες απελευθερώνουν ένα πλήκτρο στο πληκτρολόγιο. ,το οποίο χειρίζεται ένας χειριστής συμβάντων onkeyup. Αυτό το παράδειγμα εμφανίζει ένα μήνυμα ειδοποίησης όταν συμβαίνει.
- **The Συμβάν Keypress (onkeypress):** συμβαίνει όταν ένας χρήστης πατήσει ένα πλήκτρο στο πληκτρολόγιο που έχει μια τιμή χαρακτήρα που συνδέεται με αυτό. Π.χ., Ctrl, Shift, Alt, Esc, Arrow keys κ.λπ. δεν θα δημιουργήσει ένα συμβάν keypress, αλλά θα δημιουργήσουν ένα συμβάν keydown και keyup. Ο χειριστής συμβάντων onkeypress χειρίζεται ένα συμβάν keypress. Μπορεί να ελεγχθεί εδώ μέσω ενός μηνύματος συναγερμού.

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

Συμβάντα JavaScript

- **Συμβάντα Φόρμας (Form Events)**

Ένα συμβάν φόρμας ενεργοποιείται μόλις ένας έλεγχος φόρμας λάβει ή χάσει εστίαση ή όταν ο χρήστης τροποποιήσει μια τιμή ελέγχου φόρμας (π.χ. πληκτρολογώντας κείμενο σε μια είσοδο κειμένου), ή πατήσει μια επιλογή σε ένα πλαίσιο επιλογών κ.λπ.

- **Το Focus Event (onfocus):** συμβαίνει όταν οι χρήστες δίνουν έμφαση σε ένα στοιχείο σε μια ιστοσελίδα. και τον διαχειρίζεται ένας χειριστής συμβάντων εστίασης. Αυτό το παράδειγμα θα επισημάνει το φόντο της εισαγωγής κειμένου σε κίτρινο χρώμα όταν ένας χρήστης εστιάσει πάνω σ' αυτό.
- **Το συμβάν θόλωσης (onblur):** συμβαίνει όταν ο χρήστης απομακρύνει την εστίαση από ένα παράθυρο ή στοιχείο φόρμας. Αυτό το διεχειρίζεται ένας χειριστής συμβάντων onblur. Το ακόλουθο παράδειγμα θα εμφανίσει ένα μήνυμα ειδοποίησης μόλις το στοιχείο εισαγωγής κειμένου χάσει εστίαση.
- **Το συμβάν αλλαγής (onchange):** συμβαίνει μόλις ένας χρήστης τροποποιήσει την τιμή ενός στοιχείου φόρμας. Χειριστής συμβάντων: onchange. Αυτό το παράδειγμα εμφανίζει ένα μήνυμα ειδοποίησης όταν αλλάζει η επιλογή στο πλαίσιο επιλογής.

Το συμβάν υποβολής (onsubmit): συμβαίνει μόνο όταν ο χρήστης υποβάλλει μια φόρμα σε μια ιστοσελίδα. Χειριστής συμβάντων: onsubmit.

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

JavaScript Events

- **Document/Window Events**

Situations in which the page has loaded or resized the browser window can as well trigger events.

- **The Load Event (onload):** it happens when a webpage fully loads in a web browser. Event handler: onload.
- **The Unload Event (onunload):** when the user leaves the present webpage, this event takes place. Event handler: onunload.
- **The Resize Event (onresize):** it happens when the Internet user resizes, minimises or maximises the browser window. Event handler: onresize.

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

Συμβολοσειρές JavaScript

Στην JavaScript, οι συμβολοσειρές διαδραματίζουν βασικό ρόλο στη συνολική δομή μιας ιστοσελίδας, καθώς είναι μια ακολουθία γραμμάτων, αριθμών, ειδικών χαρακτήρων και αριθμητικών τιμών ή ακόμα και ένας συνδυασμός όλων. Μπορούν να δημιουργηθούν με το περιτύλιγμα/ την περιτύλιξη της συμβολοσειράς/στοιχειοσειράς κειμένου (δηλαδή χαρακτήρες συμβολοσειράς) είτε μέσα σε μονά εισαγωγικά (') ή διπλά εισαγωγικά ("), ως εξής:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Creating Strings in JavaScript</title>
</head>
<body>
    <script>
    // Creating variables
    var myString = 'Hello World!'; // Single quoted string
    var myString = "Hello World!"; // Double quoted string

    // Printing variable values
    document.write(myString + "<br>");
    document.write(myString);
    </script>
</body>
</html>
```

Hello World!
Hello World!

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

Συμβολοσειρές JavaScript

Τα εισαγωγικά μπορούν να χρησιμοποιηθούν μέσα σε μια συμβολοσειρά, αλλά δεν πρέπει να ταιριάζουν με τα εισαγωγικά που περιβάλλουν τη συμβολοσειρά:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Using Quotes inside JavaScript Strings</title>
</head>
<body>
    <script>
    // Creating variables
    var str1 = "it's okay";
    var str2 = 'He said "Goodbye"';
    var str3 = "She replied 'Calm down, please'";

    // Printing variable values
    document.write(str1 + "<br>");
    document.write(str2 + "<br>");
    document.write(str3);
    </script>
</body>
</html>
```

```
it's okay
He said "Goodbye"
She replied 'Calm down, please'
```

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

Συμβολοσειρές JavaScript

Παρ 'όλα αυτά, τα μονά εισαγωγικά μπορούν ακόμα να τοποθετηθούν μέσα σε στοιχειοσειρές με μονά εισαγωγικά ενώ τα διπλά εισαγωγικά σε συμβολοσειρές με διπλά εισαγωγικά, διαχωρίζοντας τα εισαγωγικά με ένα χαρακτήρα backlash (\), όπως φαίνεται στο *Σχήμα 36*. Το backlash ορίζεται ως ένας **χαρακτηρας διαφυγής** και οι ακολουθίες\'and\' είναι **ακολουθίες διαφυγής**.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Escaping Quotes inside JavaScript Strings</title>
</head>
<body>
    <script>
    // Creating variables
    var str1 = 'it\'s okay';
    var str2 = "He said \"Goodbye\"";
    var str3 = 'She replied \'Calm down, please\'';

    // Printing variable values
    document.write(str1 + "<br>");
    document.write(str2 + "<br>");
    document.write(str3);
    </script>
</body>
</html>
```

```
it's okay
He said "Goodbye"
She replied 'Calm down, please'
```

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

Συμβολοσειρές JavaScript

Οι ακολουθίες διαφυγής είναι επίσης χρήσιμες για την προσθήκη χαρακτήρων που δεν μπορούν να εισαχθούν μέσω ενός πληκτρολογίου. Μερικές άλλες ακολουθίες διαφυγής που χρησιμοποιούνται συχνότερα είναι:

- Το\n αντικαθίσταται από το χαρακτήρα νέας γραμμής (newline charater)
- \t αντικαθίσταται από τον χαρακτήρα καρτέλας (tab character)
- το\r αντικαθίσταται από το χαρακτήρα επιστροφής φορέα (carriage-return character)
- Το\b αντικαθίσταται από τον χαρακτήρα backspace
- Το\\ αντικαθίσταται από ένα χαρακτήρα backlash (\)

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

Συμβολοσειρές JavaScript

Εκτέλεση λειτουργιών σε συμβολοσειρές

- Η JavaScript καθιστά διαθέσιμες διάφορες ιδιότητες και μεθόδους για να εκτελεί λειτουργίες στις τιμές των στοιχειοσειρών.
- Πιο συγκεκριμένα, μόνο τα αντικείμενα μπορούν να έχουν ιδιότητες και μεθόδους. Ωστόσο, στην JavaScript, οι πρωτόγονοι τύποι δεδομένων μπορούν να εκτελέσουν λειτουργίες όπως και τα αντικείμενα, όταν ο προγραμματιστής αναφέρεται σε αυτά με την ιδιότητα σημειογραφίας πρόσβασης (property access notation).

Η JavaScript προσφέρει αυτή τη δυνατότητα μέσω της δημιουργίας ενός provisional wrapper object για πρωτόγονους τύπους δεδομένων. Αυτή η διαδικασία γίνεται αυτόματα από τον διερμηνέα JS στο παρασκήνιο.

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

Συμβολοσειρές JavaScript

Λήψη του μήκους μιας συμβολοσειράς

Η ιδιότητα μήκους επιστρέφει το μήκος μιας συμβολοσειράς, το οποίο είναι ο αριθμός των χαρακτήρων που οριοθετούνται σε μια συμβολοσειρά, συμπεριλαμβανομένου του αριθμού των ειδικών χαρακτήρων επίσης, όπως\t ή\n. Οι προγραμματιστές πρέπει να είναι προσεκτικοί στη χρήση παρενθέσεων μετά το μήκος (π.χ. str.length ()), καθώς ο σωστός τρόπος είναι str.length (διαφορετικά, θα δημιουργήσει ένα σφάλμα).

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Get String Length in JavaScript</title>
</head>
<body>
    <script>
    var str1 = "This is a paragraph of text.";
    document.write(str1.length + "<br>"); // Prints 28

    var str2 = "This is a \n paragraph of text.";
    document.write(str2.length); // Prints 30, because \n is only one
character
    </script>
</body>
</html>
```

28
30

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

Στοιχειοσειρές JavaScript

Η μέθοδος indexOf () μπορεί να χρησιμοποιηθεί για την εύρεση μιας υποσυμβολοσειράς ή συμβολοσειράς μέσα σε μια άλλη συμβολοσειρά. Αυτή η τεχνική επιστρέφει τον δείκτη ή τη θέση της πρώτης εμφάνισης μιας καθορισμένης συμβολοσειράς μέσα σε μια συμβολοσειρά.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Find the Position of Substring within a String</title>
</head>
<body>
    <script>
    var str = "If the facts don't fit the theory, change the facts.";
    var pos = str.indexOf("facts");
    document.write(pos); // 0utputs: 7
    </script>
</body>
</html>
```

```
7
```

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

Στοιχειοσειρές JavaScript

Εύρεση συμβολοσειράς μέσα σε άλλη συμβολοσειρά

Παρομοίως, η τεχνικήlastIndexOf () μπορεί να χρησιμοποιηθεί για να λάβει το δείκτη ή τη θέση της τελευταίας εμφάνισης της καθορισμένης συμβολοσειράς μέσα σε μια συμβολοσειρά, ως εξής:

Τόσο οι προσεγγίσεις του indexOf(), όσο και του lastIndexOf () μπορούν να επιστραφούν -1 εάν η υποσυμβολοσειρά δεν βρεθεί. Και οι δύο μέθοδοι επίσης δέχονται μια προαιρετική ακέραια παράμετρο η οποία ορίζει τη θέση εντός της συμβολοσειράς στην οποία θα ξεκινήσει η αναζήτηση.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Find the Position of Substring within a String</title>
</head>
<body>
    <script>
    var str = "If the facts don't fit the theory, change the facts.";
    var pos = str.lastIndexOf("facts");
    document.write(pos); // 0utputs: 46
    </script>
</body>
</html>
```

```
46
```

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

Συμβολοσειρές JavaScript

Αναζήτηση ενός μοτίβου μέσα σε μια συμβολοσειρά

Η μέθοδος αναζήτησης() μπορεί να χρησιμοποιηθεί για την αναζήτηση ενός συγκεκριμένου κομματιού κειμένου ή μοτίβου μέσα σε μια συμβολοσειρά. Καθώς η προσέγγισηindexOf (), η search() επιστρέφει επίσης το ευρετήριο της πρώτης αντιστοίχισης, και επιστρέφει το -1 no matches were found, αλλά σε αντίθεση με το indexOf (), το search () μπορεί επίσης να λάβει μια *κανονική έκφραση* ως όρισμα/παράμετρο για την παροχή προηγμένων δεξιοτήτων αναζήτησης. Θα πρέπει να αναφερθεί ότι η προσέγγιση αναζήτησης() δεν υποστηρίζει παγκόσμιες αναζητήσεις, καθώς αγνοεί τη σημαία g (flag) ή τον τροποποιητή του ορίσματος της κανονικής έκφρασης.

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

Συμβολοσειρές JavaScript

Αναζήτηση ενός μοτίβου μέσα σε μια συμβολοσειρά

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Search Text or Pattern inside a String</title>
</head>
<body>
    <script>
    var str = "Color red looks brighter than color blue.";

    // Case sensitive search
    var pos1 = str.search("color");
    document.write(pos1 + "<br>"); // 0utputs: 30

    // Case insensitive search using regexp
    var pos2 = str.search(/color/i);
    document.write(pos2); // 0utputs: 0
    </script>
</body>
</html>
```

```
30
0
```

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

Συμβολοσειρές JavaScript

Αναζήτηση ενός μοτίβου μέσα σε μια συμβολοσειρά

Για να αφαιρέσετε ένα μέρος ή υποσυμβολοσειρά από μια συμβολοσειρά, μπορείτε να χρησιμοποιήσετε την μέφοδο slice (). Αυτή λαμβάνει δύο παραμέτρους: *αρχικός δείκτης* (δείκτης όπου εκκινεί την εξαγωγή), και ένας προαιρετικός *τελικός δείκτης (δείκτης* πριν από τον οποίο τελειώνει η εξαγωγή), όπως το str.slice(startIndex, endIndex). Το παρακάτω παράδειγμα χωρίζει ένα τμήμα μιας συμβολοσειράς από τη θέση 4 στη θέση 15:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Slice Out a Portion of a String</title>
</head>
<body>
    <script>
    var str = "The quick brown fox jumps over the lazy dog.";
    var subStr = str.slice(4, 15);
    document.write(subStr); // Prints: quick brown
    </script>
</body>
</html>
```

quick brown

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

Συμβολοσειρές JavaScript

Εξαγωγή μιας υποσυμβολοσειράς από μια συμβολοσειρά

Μπορούν επίσης να καθοριστούν αρνητικές τιμές. Αυτές οι τιμές επεξεργάζονται ως strLength + startIndex, όπου το strLength είναι το μήκος της συμβολοσειράς (str.length), για παράδειγμα, αν το startIndex είναι -5 θα επεξεργασθεί ως strLength - 5. Αν το startIndex είναι μεγαλύτερο ή ίσο με το μήκος της συμβολοσειράς, η μέθοδος slice() επιστρέφει μια κενή συμβολοσειρά. Αντίστοιχα, εάν το προαιρετικό endIndex δεν έχει καθοριστεί ή παραλειφθεί, η μέθοδος slice() εξάγεται στο τέλος της συμβολοσειράς

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

Συμβολοσειρές JavaScript

Εξαγωγή μιας υποσυμβολοσειράς από μια συμβολοσειρά

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>JavaScript Slice Strings Using Negative Indexes</title>
6  </head>
7  <body>
8      <script>
9      var str = "The quick brown fox jumps over the lazy dog.";
10     document.write(str.slice(-28, -19) + "<br>"); // Prints: fox jumps
11     document.write(str.slice(31)); // Prints: the lazy dog.
12     </script>
13 </body>
14 </html>
```

```
fox jumps
the lazy dog.
```

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

Συμβολοσειρές JavaScript

Εξαγωγή μιας υποσυμβολοσειράς από μια συμβολοσειρά

Η μέθοδος subring () για την εξαγωγή ενός τμήματος της δοσμένης συμβολοσειράς με βάση τους δείκτες έναρξης και τέλους, ως str.subring(startIndex, endIndex). Η μέθοδος υποσυμβολοσειράς() είναι κατά πολύ συγκρίσιμη με την τομή() (slice), εκτός από ορισμένες διαφορές:

- Εάν οποιοδήποτε από τα δύο ορίσματα είναι μικρότερο από 0 ή είναι NaN, τότε αντιμετωπίζεται ως 0.
- Εάν οποιοδήποτε από τα δύο ορίσματα είναιμεγαλύτερο από το str.length, αντιμετωπίζεται σαν να ήταν str.length.
- Αν το startIndex είναι μεγαλύτερο από toendIndex, τότε το subring () θα αλλάξει αυτά τα δύο ορίσματα, δηλαδή το str.substring (5, 0) ==str.substring (0, 5).

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

Συμβολοσειρές JavaScript

Εξαγωγή συγκεκριμένου αριθμού χαρακτήρων από μια συμβολοσειρά

Η JavaScript παρέχει επίσης την τεχνικήsubr (), η οποία είναι παρόμοια με την slice() με μια μικρή διαφορά: η δεύτερη παράμετρος ορίζει τον αριθμό των χαρακτήρων που θα εξαχθούν αντί για τον τελικό δείκτη, ως str.substr (startIndex, length). Αν το μήκος είναι 0 ή αρνητικός αριθμός, επιστρέφεται μια κενή συμβολοσειρά.

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

Συμβολοσειρές JavaScript

Εξαγωγή συγκεκριμένου αριθμού χαρακτήρων από μια συμβολοσειρά

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>JavaScript Extract Fixed Number of Characters from a
   String</title>
6  </head>
7  <body>
8      <script>
9      var str = "The quick brown fox jumps over the lazy dog.";
10     document.write(str.substr(4, 15) + "<br>"); // Prints: quick brown fox
11     document.write(str.substr(-28, -19) + "<br>"); // Prints nothing
12     document.write(str.substr(-28, 9) + "<br>"); // Prints: fox jumps
13     document.write(str.substr(31)); // Prints: the lazy dog.
14     </script>
15 </body>
16 </html>
```

```
quick brown fox

fox jumps
the lazy dog.
```

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

Συμβολοσειρά JavaScript

Αντικατάσταση των περιεχομένων μιας συμβολοσειράς

Η μέθοδος replace() χρησιμοποιείται για την αντικατάσταση μέρους μιας συμβολοσειράς με άλλη συμβολοσειρά. Αυτή η προσέγγιση παίρνει μια κανονική έκφραση για να ταιριάξει η υποσυμβολοσειρά με δύο παραμέτρους και μια συμβολοσειρά αντικατάστασης, δηλαδή str.replace (regexp|subr, newSubstr). Η μέθοδος replace() επιστρέφει μια νέα συμβολοσειρά και δεν ταράζει την αρχική συμβολοσειρά, η οποία θα παραμείνει ανεπηρέαστη.

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

Συμβολοσειρά JavaScript

Αντικατάσταση των περιεχομένων μιας συμβολοσειράς

- Από προεπιλογή, η μέθοδος replace() υποκαθιστά μόνο την πρώτη αντιστοίχιση και ;έχει διάκριση πεζών/ κεφαλαίων. Για να αντικαταστήσετε την υποσυμβολοσειρά μέσα σε μια συμβολοσειρά με έναν τρόπο που δεν επηρεάζεται από πεζά/κεφαλαία γράμματα, μπορεί να χρησιμοποιηθεί μια *κανονική έκφραση (regexp)* με έναν τροποποιητή i,
- Επίσης, για να αντικατασταθούν όλες οι περιπτώσεις μιας υποσυμβολοσειράς μέσα σε μια συμβολοσειρά με έναν τρόπο που δεν επηρεάζεται από πεζά/κεφαλαία γράμματα, μπορεί να χρησιμοποιηθεί ο τροποποιητής g μαζί με τον τροποποιητή i.

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

Συμβολοσειρές JavaScript

Μετατροπή συμβολοσειράς σε κεφαλαία ή πεζά γράμματα

- Η μέθοδος toUpperCase() χρησιμοποιείται για τη μετατροπή μιας συμβολοσειράς σε κεφαλαία
- Παρομοίως, η μέθοδος toLowerCase() χρησιμοποιείται για τη μετατροπή μιας συμβολοσειράς σε πεζά.

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

Συμβολοσειρές JavaScript

Συνδυασμός δύο ή περισσότερων συμβολοσειρών

Δύο ή περισσότερες συμβολοσειρές μπορούν να συνενωθούν ή να συνδυαστούν χρησιμοποιώντας τους τελεστές ανάθεσης + και +=.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Join Two or More Strings</title>
</head>
<body>
    <script>
    var hello = "Hello";
    var world = "World";
    var greet = hello + " " + world;
    document.write(greet + "<br>"); // Prints: Hello World

    var wish  = "Happy";
        wish += " New Year";
    document.write(wish); // Prints: Happy New Year
    </script>
</body>
</html>
```

Hello World
Happy New Year

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

Συμβολοσειρές JavaScript

Πρόσβαση σε μεμονωμένους χαρακτήρες από μια συμβολοσειρά

- Η μέθοδοςCharAt () μπορεί να χρησιμοποιηθεί για την πρόσβαση μεμονωμένου χαρακτήρα από μια συμβολοσειρά, ως str.charAt(index). Ο καθορισμένος δείκτης θα πρέπει να είναι ένας αριθμός μεταξύ 0 και str.length - 1. Αν δεν δοθεί ευρετήριο, επιστρέφεται ο πρώτος χαρακτήρας στη συμβολοσειρά, αφού η προεπιλογή είναι 0.

- Ωστόσο, υπάρχει μια καλή εναλλακτική λύση σε αυτή τη διαδικασία. Δεδομένου ότι το ECMAScript 5, οι συμβολοσειρές μπορούν να αντιμετωπιστούν ως πίνακες μόνο για ανάγνωση και οι μεμονωμένοι χαρακτήρες μπορούν να εμφανιστούν από μια συμβολοσειρά χρησιμοποιώντας αγκύλες ([]) αντί της προσέγγισης CharAt ().

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

Συμβολοσειρές JavaScript

Διαχωρισμός μιας συμβολοσειράς σε μια συστοιχία

Η μέθοδος split() μπορεί να χρησιμοποιηθεί για να κατακερματίσει μια συμβολοσειρά σε έναν πίνακα συμβολοσειρών, χρησιμοποιώντας την σύνταξη str.split(διαχωριστικό, όριο). Το όρισμα διαχωριστή ορίζει τη συμβολοσειρά στην οποία θα πρέπει να συμβεί κάθε διαίρεση, ενώ τα ορίσματα ορίου καθορίζουν το μέγιστο μήκος του πίνακα. Εάν το διαχωριστικό όρισμα παραλειφθεί ή δεν βρεθεί στην καθορισμένη συμβολοσειρά, ολόκληρη η συμβολοσειρά κατανέμεται στο πρώτο στοιχείο του πίνακα.

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

Αριθμοί JavaScript

Υπάρχουν δύο τύποι αριθμών στο JavaScript:

- **Οι κανονικοί αριθμοί** σε JavaScript αποθηκεύονται σε μορφή 64-bit IEEE-754, αλλιώς ειναι γνωστοί ως «*αριθμοί κινητής υποδιαστολής διπλής ακρίβειας*». Αυτοί είναι οι πιο συνηθισμένοι τύποι αριθμών.

- **Αριθμοί BigInt**, που αντιπροσωπεύουν ακέραιους αριθμούς τυχαίου μήκους (random lenghtlength). Μερικές φορές χρειάζονται, επειδή ένας κανονικός αριθμός δεν μπορεί να υπερβαίνει με ασφάλεια τα 253 ή να είναι μικρότερος από $^{-253}$.

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

Αριθμοί JavaScript

- Υπάρχουν περισσότεροι τρόποι να γράψεις έναν αριθμό. Για παράδειγμα, ο προφανής τρόπος εγγραφής 1 δισεκατομμυρίου θα ήταν "1000000000" ή "1_000_000_000", χρησιμοποιώντας την υπογράμμιση ως διαχωριστικό. Σε αυτή την περίπτωση, η υπογράμμιση είναι η «συντακτική ζάχαρη» (syntactic sugar), που σημαίνει ότι καθιστά τον αριθμό πιο ευανάγνωστο. Ο μηχανή της JS αγνοεί την υπογράμμιση μεταξύ των ψηφίων, οπότε είναι ακριβώς το ίδιο, δηλαδή το ένα δισεκατομμύριο της πρώτης περίπτωσης.
- Ωστόσο, στην πραγματικότητα, όλοι σίγουρα θα προσπαθήσουν να αποφύγουν να γράψουν μεγάλες ακολουθίες μηδενικών. Κάτι σαν «1 δις» για ένα δισεκατομμύριο ή «2,5 δις» για 2 δισεκατομμύρια 500 εκατομμύρια φαίνεται πιο λογικό. Η ίδια αρχή ισχύει για τους περισσότερους μεγάλους αριθμούς. Τούτου λεχθέντος, είναι δυνατόν να συντομευθεί ένας αριθμός σε JS, προσθέτοντας το γράμμα "e" σε αυτό και καθορίζοντας την ποσότητα των μηδενικών.

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

Αριθμοί JavaScript

```
1  let billion = 1e9;  // 1 billion, literally: 1 and 9 zeroes
2
3  alert( 7.3e9 );  // 7.3 billions (same as 7300000000 or 7_300_000_000)
```

- Έτσι, το e πολλαπλασιάζει τον αριθμό με το 1 με το δεδομένο αριθμό μηδενικών.

```
1e3 === 1 * 1000; // e3 σημαίνει *1000
1.23e6 == 1.23 * 1000000; // e6 σημαίνει *1000000
```

- Η ίδια αρχή ισχύει και για τους μικρούς αριθμούς. Για παράδειγμα, τι πρέπει να γραφτεί για 1 μικροδευτερόλεπτο (ένα εκατομμυριοστό του δευτερολέπτου).

```
let mcs = 0,000001 ·
```

- Ακριβώς όπως η προαναφερθείσα περίπτωση με τους μεγάλους αριθμούς, χρησιμοποιώντας το "e" μπορεί να είναι χρήσιμο. Για να αποφύγετε να γράψετε τα μηδενικά ρητά, μπορούν να γραφτούν τα ακόλουθα:

```
mcs = 1e-6; // έξι μηδενικά προς τα αριστερά από 1
```

- Υπάρχουν 6 μηδενικά στο 0.000001. Στη συνέχεια, δεν είναι δύσκολο να συναχθεί το συμπέρασμα 1e-6.

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

Αριθμοί JavaScript

Επομένως, ένας αρνητικός αριθμός μετά το "e" υποδηλώνει μια διαίρεση με το 1 με το δοσμένο αριθμό μηδενικών, ως εξής:

```
1  // -3 divides by 1 with 3 zeroes
2  1e-3 === 1 / 1000; // 0.001
3
4  // -6 divides by 1 with 6 zeroes
5  1.23e-6 === 1.23 / 1000000; // 0.00000123
```

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

Αριθμοί JavaScript

Δεκαεξαδικοί, δυαδικοί και οκταδικοί αριθμοί

Οι δεκαεξαδικοί αριθμοί χρησιμοποιούνται συνήθως στην JavaScript για να ενσωματώσουν χρώματα, να κωδικοποιήσουν χαρακτήρες και για πολλούς άλλους σκοπούς. Έτσι, προφανώς, υπάρχει ένας γρηγορότερος τρόπος για να τους γράψετε, πιο συγκεκριμένα προσθέτοντας το πρόθεμα (prefix) 0x (0x) ακολουθούμενο από μια ακολουθία αριθμών, ως εξής:

```
1  alert( 0xff ); // 255
2  alert( 0xFF ); // 255 (the same, case doesn't matter)
```

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

Αριθμοί JavaScript

Δεκαεξαδικοί, δυαδικοί και οκταδικοί αριθμοί

- Τα δυαδικά και οκταδικά αριθμητικά συστήματα δεν χρησιμοποιούνται συχνά, αλλά και αυτά μπορούν να υποστηριχθούν με την χρήση άλλων προθεμάτων όπως το 0b και το 0o (*μηδέν*, *ο*):
- Μόνο τα προαναφερθέντα 3 αριθμητικά συστήματα μπορούν να υποστηριχθούν μέσω της προσθήκης προθεμάτων. Σε άλλα αριθμητικά συστήματα, μπορεί να χρησιμοποιηθεί η συνάρτηση parseInt.

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

Αριθμοί JavaScript

Η μέθοδος toString(base)

- Η μέθοδος num.toString (base) επιστρέφει μια αναπαράσταση της συμβολοσειράς του num στο αριθμητικό σύστημα με την παρεχόμενη βάση (base), ως εξής:

```
1  let num = 255;
2
3  alert( num.toString(16) );  // ff
4  alert( num.toString(2) );   // 11111111
```

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

Αριθμοί JavaScript

Η μέθοδος toString(base)

- Υπάρχει μια συγκεκριμένη αξιοσημείωτη περίπτωση. Όταν εντοπίζονται δύο τελείες στο 123456..toString(36), αυτό δεν είναι τυπογραφικό λάθος. Όταν ο προγραμματιστής θέλει να καλέσει μια μέθοδο απευθείας σε έναν αριθμό, είναι απαραίτητο να τοποθετήσετε δύο τελείες (‘..‘) μετά από αυτό.
- Εάν, όμως, τοποθετηθεί μία τελεία ως εξής [‘123456.toString(36)’], αυτό θα αποτελούσε σφάλμα, καθώς η σύνταξη JavaScript πρέπει να περιέχει το δεκαδικό τμήμα μετά την πρώτη τελεία. Αν ο προγραμματιστής τοποθετήσει μια ακόμα τελεία, τότε η JavaScript ξέρει ότι το δεκαδικό τμήμα είναι κενό και ότι τώρα δεν μπορεί να ολοκληρωθεί η μέθοδος.

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

Αριθμοί JavaScript

Στρογγυλοποίηση (rounding)

Μία από τις πιο εφαρμοσμένες λειτουργίες κατά τη λειτουργία με αριθμούς είναι η **στρογγυλοποίηση**.

Υπάρχουν ορισμένες ενσωματωμένες λειτουργίες για τη στρογγυλοποίηση:

- **Math.floor**

Η λειτουργεία Math.floor στρογγυλοποιεί προς τα κάτω το 3.1 μετατρεποντάς το σε 3, ενώ το -1.1 το μετατρέπει σε -2.

- **Math.ceil**

Η λειτουργία Math.ceil στρογγυλοποιεί το 3.1 μετατρέποντάς το σε 4, ενώ το -1.1 το μετατρέπει σε -1.

- **Math.round**

Η λειτουργία Math.round στρογγυλοποιεί στον πλησιέστερο ακέραιο αριθμό. Για παράδειγμα το 3.1 μετατρέπεται σε 3, το 3.6 σε 4, ενώ το 3.5 (δηλαδή οι μεσσείες τιμές 0,5) στρογγυλοποιείται επίσης σε 4.

- **Math.trunc (not supported by Internet Explorer)**

Εξαλείφει οτιδήποτε μετά το δεκαδικό οριοθέτη (την τελεία) χωρίς στρογγυλοποίηση. Για παράδειγμα, το 3.1 μετατρέπεται σε 3 και το -1.1 γίνεται -1.

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

Αριθμοί JavaScript

Στρογγυλοποίηση

Αυτές οι συναρτήσεις καλύπτουν όλους τους δυνητικούς τρόπους διαχείρισης του δεκαδικού μέρους ενός αριθμού. Αλλά τι συμβαίνει όταν θέλουμε να στρογγυλοποιήσουμε έναν αριθμό στη νιοστή του δύναμη μετά από τον δεκαδικό οριοθέτη; Για παράδειγμα, το δεκαδικό μέρος του 1,2345 στρογγυλοποιείται σε δύο ψηφία (1,23).

Υπάρχουν δύο τρόποι για να γίνει αυτό:

1. Πολλαπλασιασμός και διαίρεση.

Π.χ., για να στρογγυλοποιηθεί το δεκαδικό μέρος ενός αριθμού μέχρι το 2ο του ψηφίο, μπορεί να πολλαπλασιαστεί με το 100 (ή με μια μεγαλύτερη δύναμη από το 10), να καλέσει τη συνάρτηση στρογγυλοποίησης και στη συνέχεια να τη διαιρέσει.

2. Η μέθοδος **toFixed (n)** στρογγυλοποιεί τον αριθμό στη νιοστή του δύναμη (δηλαδή σε n= εκθέτης) μετά τον δεκαδικό οριοθέτη (τελεία) και επιστρέφει μια συμβολοσειρά που αναπαριστά το γινόμενο του πολλαπλασιασμού τον n παραγόντων.

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

Αριμοί JavaScript

Ανακριβείς υπολογισμοί

- Σε ένα αριθμητικό σύστημα, ένας αριθμός αντιπροσωπεύεται υπό την μορφή 64-bit (**IEEE-754**). Έτσι λοιπόν ένας αριθμός αποθηκεύεται χρησιμοποιώντας 64 bits: 52 εκ των οποίων χρησιμοποιούνται για την αποθήκευση των ψηφίων του ακέραιου μέρους ενός αριθμού, 11 για την αποθήκευση του δεκαδικού μέρους ενός αριθμού (10 μηδενικά για τις ακέραιες τιμές στο δεκαδικό μέρος, π.χ. 3.10000000000), και 1 bit είναι για τον δεκαδικό οριοθέτη (την τελεία).

- Αν ένας αριθμός είναι πολύ μεγάλος, τότε δεν θα χωρέσει στην αποθήκευση των 64-bit, δίνοντας ενδεχομένως μια άπειρη τιμή.

- Αυτή η απώλεια ακρίβειας είναι αρκετά συνήθης.

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

Αριθμοί JavaScript

Ανακριβείς υπολογισμοί

- Η πιο αξιόπιστη μέθοδος για την αντιμετώπιση της κατάστασης είναι η χρήση της συνάρτησης **toFixed(n)**:

```
1  let sum = 0.1 + 0.2;
2  alert( sum.toFixed(2) ); // 0.30
```

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

Αριθμοί JavaScript

Δοκιμές: isFinite και isNaN

- Μια τιμή **Infinity** (και -infinity) είναι μια ειδική αριθμητική τιμή που είναι μεγαλύτερη (μικρότερη) από οποιαδήποτε άλλη.
- Το **NaN, συντομογραφία** του **not a number,** αντιπροσωπεύει ένα σφάλμα.

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

Αριθμοί JavaScript

Δοκιμές: isFinite και isNaN

Και οι δύο προαναφερθέντες τιμές (του απείρου και του NaN) ενώ ανήκουν σε **έναν** τύπο αριθμών, δεν είναι εντούτοις «κανονικοί» αριθμοί, και γι' αυτό υπάρχουν συγκεκριμένες λειτουργίες για τον έλεγχό τους:

- **το isNaN (τιμή)** μετατρέπει το όρισμά του σε έναν αριθμό και στη συνέχεια τον δοκιμάζει ως **NaN**:

```
1  alert( isNaN(NaN) ); // true
2  alert( isNaN("str") ); // true
```

- το **isFinite (τιμή)** μετατρέπει το όρισμά του σε έναν αριθμό και το επιστρέφει ως ένα αποτέλεσμα που είναι "**Αληθές**" αν είναι κανονικός αριθμός, και όχι αν είναι **NaN/Infinity/-Infinity;**

```
1  alert( isFinite("15") ); // true
2  alert( isFinite("str") ); // false, because a special value: NaN
3  alert( isFinite(Infinity) ); // false, because a special value: Infinity
```

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

Αριθμοί JavaScript

Το **parseInt** και το **parseFloat**

- Η αριθμητική μετατροπή με τη χρήση του συν **+** ή **Αριθμού()** είναι εξονυχιστική. Εάν μια τιμή δεν αναπαριστάνει ακριβώς έναν αριθμό, τότε η αριθμητική μετατροπή είναι αποτυχής.
- Οι μέθοδοι **parseInt** και **parseFloat** χρησιμοποιούνται για την εξαγωγή αριθμητικών τιμών σε απρόσμενες, παράδοξες καταστάσεις (π.χ. όταν τα σύμβολα τοποθετούνται μετά τον αριθμό – όπως στην περίπτωση μιας τιμής 18€).
- «*Διαβάζουν*» έναν αριθμό από μια συμβολοσειρά μέχρι που δεν μπορούν να τον "διαβάσουν". Σε περίπτωση σφάλματος, επιστρέφεται ο αριθμός που συλλέχθηκε. Η μέθοδος parseInt **επιστρέφει** έναν ακέραιο αριθμό, ενώ η **συνάρτηση** parseFloat επιστρέφει έναν αριθμό κινητής υποδιαστολής:

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

Αριθμοί JavaScript

Άλλες μαθηματικές λειτουργίες

Η JavaScript κατέχει ένα ενσωματωμένο αντικείμενο μαθηματικών που περιλαμβάνει μια μικρή βιβλιοθήκη μαθηματικών συναρτήσεων και σταθερών:

- **Math.random()**

  Επιστρέφει έναν τυχαίο αριθμό από το 0 μέχρι 1 (μη συμπεριλαμβανομένου του 1).

- **Math.max(a, b, c...) / Math.min(a, b, c...)**

  Επιστρέφει το μεγαλύτερο/μικρότερο από τον αυθαίρετο αριθμό ορισμάτων.

- **Math.pow(n, ανύψωση ενός αριθμού στη δύναμη ενός εκθέτη)**

  Επιστρέφει το αποτέλεσμα, δηλαδή το γινόμενο στην προκειμένη περίπτωση, της ύψωσης ενός αριθμού στη νιοστή δύναμη, δηλαδή στη δύναμη του n εκθέτη του.

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

JavaScript Δηλώσεις If…Else

Όπως πολλές άλλες γλώσσες προγραμματισμού, η JavaScript επιτρέπει επίσης την εγγραφή κωδίκων που εκτελούν διαφορετικές ενέργειες με βάση τα αποτελέσματα τύπων υπολογισμών, όπως τους λογικούς τελεστές ή τους συγκριτικούς τελεστές και σε χρόνο εκτέλεσης. Έτσι, οι συνθήκες δοκιμής μπορούν να δημιουργηθούν ως εκφράσεις που αξιολογούν είτε το "Αληθές" είτε το "Ψευδές" και, με βάση αυτά τα αποτελέσματα, μπορούν να εκτελεστούν ορισμένες ενέργειες.

Υπάρχουν αρκετές υπό όρους δηλώσεις στο JavaScript που μπορούν να χρησιμοποιηθούν για τη λήψη αποφάσεων:

- Η δήλωση **if** ·
- Η δήλωση **if...else**;
- Η δήλωση **if...else if....else**;
- Η δήλωση **switch...case**

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

JavaScript Δηλώσεις If…Else

Η δήλωση if

Η δήλωση if χρησιμοποιείται για την εκτέλεση ενός μπλοκ κώδικα μόνο αν η καθορισμένη συνθήκη αξιολογείται ως αληθής. Αυτές είναι οι απλούστερες υπό όρους δηλώσεις της JS και μπορούν να γραφτούν ως:

```
if(condition) {
    // Code to be executed
}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript If Statement</title>
</head>
<body>
    <script>
    var now = new Date();
    var dayOfWeek = now.getDay(); // Sunday - Saturday : 0 - 6

    if(dayOfWeek == 5) {
        document.write("Have a nice weekend!");
    }
    </script>
    <p><strong>Note:</strong> This example will print "Have a nice weekend!" if the current day is Friday.</p>
</body>
</html>
```

Note: This example will print "Have a nice weekend!" if the current day is Friday.

JavaScript Δηλώσεις If…Else

Η δήλωση if…else statement

Η διαδικασία λήψης αποφάσεων της JS μπορεί να ενισχυθεί με την παροχή μιας εναλλακτικής επιλογής μέσω της προσθήκης μιας **άλλης δήλωσης** στη **δήλωση if**. Η δήλωση if…else επιτρέπει την εκτέλεση ενός κώδικα μπλοκ εάν η καθορισμένη συνθήκη αξιολογείται ως *αληθής* και εάν ένας άλλος κώδικας μπλοκ αξιολογείται ως *ψευδής*.

```
if(condition) {
    // Code to be executed if condition is true
} else {
    // Code to be executed if condition is false
}
```

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

JavaScript Δηλώσεις If…Else

Ο Τριμερής Φορέας Εκμετάλλευσης (The Ternary Operator)

Ο τριμερής φορέας εκμετάλλευσης μας επιτρέπει **να γράφουμε τις συντομογραφίες των δηλώσεων if...else**, χρησιμοποιώντας τον χαρακτήρα του ερωτηματικού (‘?’) και χρειάζονται τρεις τελεστέοι: ένας που ελέγχει και συγκρίνει τις τιμές, ένας που επιστρέφει ένα αποτέλεσμα που είναι "Αληθές" και ένας που επιστρέφει το αποτέλεσμα που είναι "Ψευδές”.

Η βασική σύνταξη έχει ως εξής:

```
var result = (condition) ? value1 : value2
```

Αν η συνθήκη αξιολογηθεί ως *αληθής,* η τιμή 1 θα επιστραφεί, αν όχι τότε θα επιστραφεί η τιμή 2.

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

Δηλώσεις Switch…Case στην JavaScript

- Μια δήλωση **switch..case** είναι ένα εναλλακτικό σενάριο της δήλωσης **if...else if...else**, το οποίο κάνει σχεδόν το ίδιο πράγμα.
- Η δήλωση **switch...case** αναλύει μια μεταβλητή ή μια έκφραση έναντι μιας σειράς τιμών μέχρι να βρει μια αντιστοιχία, και στη συνέχεια εκτελεί το μπλοκ του κώδικα που αντιστοιχεί σε αυτή την αντιστοιχία. Η βασική σύνταξη έχει ως εξής:

```
switch(x){
    case value1:
        // Code to be executed if x === value1
        break;
    case value2:
        // Code to be executed if x === value2
        break;
    ...
    default:
        // Code to be executed if x is different from all values
}
```

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

Η δήλωση Switch…Case της JavaScript

- Αυτό το παράδειγμα εμφανίζει το όνομα της ημέρας της εβδομάδας στην οποία βρίσκεται ο αναγνώστης.

- Η δήλωση **switch...case** διαφέρει από την δήλωση **if...else** σε μια σημαντική πτυχή. Η δήλωση switch...case εκτελεί γραμμή προς γραμμή και όταν η JavaScript βρίσκει μια πρόταση (case clause) που αξιολογεί ως *αληθή*, δεν εκτελεί μόνο τον κώδικα που αντιστοιχεί για την πρόταση αυτή, αλλά εκτελεί αυτόματα επίσης όλες τις διαδοχικές προτάσεις μέχρι το τέλος του μπλοκ switch.

- Για να αποφευχθεί αυτό, πρέπει να περιλαμβάνεται μια δήλωση switch μετά από κάθε περίπτωση (όπως μπορούμε να συμπεράνουμε στην *Εικόνα 73*). Η δήλωση **switch...case** ενημερώνει τον διερμηνέα της JS να σπάσει το μπλοκ switch όταν τελειώσει με την εκτέλεση του κώδικα στην περίπτωση που ένα αποτέλεσμα αξιολογείται ως *αληθές*.

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

Συστοιχίες JavaScript

- Οι συστοιχίες είναι πολύπλοκες μεταβλητές που επιτρέπουν την αποθήκευση περισσότερων από μία τιμών ή μιας ομάδας τιμών κάτω από ένα όνομα μεταβλητής. Οι συστοιχίες JavaScript μπορούν να αποθηκεύσουν οποιαδήποτε έγκυρη τιμή, συμπεριλαμβανομένων συμβολοσειρών, αριθμών, αντικειμένων, λειτουργιών και ταυτόχρονα άλλων συστοιχιών, επιτρέποντας έτσι τη δημιουργία πιο σύνθετων δομών δεδομένων, όπως μια συστοιχία αντικειμένων ή μια συστοιχία συστοιχιών.

- Στο παρακάτω παράδειγμα, το όνομα των χρωμάτων θα αποθηκευτεί σε έναν κώδικα JavaScript:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Storing Single Values</title>
</head>
<body>
    <script>
    // Creating variables
    var color1 = "Red";
    var color2 = "Green";
    var color3 = "Blue";

    // Printing variable values
    document.write(color1 + "<br>");
    document.write(color2 + "<br>");
    document.write(color3);
    </script>
</body>
</html>
```

```
Red
Green
Blue
```

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

Συστοιχίες JavaScript

Δημιουργία μιας συστοιχίας

Ο ευκολότερος τρόπος για να δημιουργήσετε μια συστοιχία στο JavaScript είναι να επισυνάψετε μια λίστα τιμών διαχωρισμένων με κόμματα σε αγκύλες ([]), όπως φαίνεται στην ακόλουθη σύνταξη:

```
var myArray = [element0, element1, ..., elementN];
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Creating Arrays in JavaScript</title>
</head>
<body>
    <script>
    // Creating variables
    var colors = ["Red", "Green", "Blue"];
    var fruits = ["Apple", "Banana", "Mango", "Orange", "Papaya"];
    var cities = ["London", "Paris", "New York"];
    var person = ["John", "Wick", 32];

    // Printing variable values
    document.write(colors + "<br>");
    document.write(fruits + "<br>");
    document.write(cities + "<br>");
    document.write(person);
    </script>
</body>
</html>
```

```
Red,Green,Blue
Apple,Banana,Mango,Orange,Papaya
London,Paris,New York
John,Wick,32
```

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

Συστοιχίες JavaScript

Πρόσβαση στα στοιχεία μιας συστοιχίας

- Τα στοιχεία πίνακα μπορούν να προσπελαστούν από τον δείκτη τους χρησιμοποιώντας τη σημειογραφία τετραγωνικής παρένθεσης. Ένας δείκτης είναι ένας αριθμός που αντιπροσωπεύει τη θέση ενός στοιχείου σε μια συστοιχία.
- Οι δείκτες των συστοιχιών βασίζονται στο μηδέν. Αυτό σημαίνει ότι το πρώτο στοιχείο μιας συστοιχίας αποθηκεύεται στο δείκτη 0, όχι 1, το δεύτερο στοιχείο αποθηκεύεται στο δείκτη 1, και ούτω καθεξής. Οι δείκτες της συστοιχίας αρχίζουν από το 0 και φτάνουν μέχρι τον αριθμό των στοιχείων μείον 1. Έτσι, η σειρά των πέντε στοιχείων θα έχει δείκτες από το 0 έως το 4.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Access Individual Elements of an Array</title>
</head>
<body>
    <script>
    var fruits = ["Apple", "Banana", "Mango", "Orange", "Papaya"];

    document.write(fruits[0] + "<br>"); // Prints: Apple
    document.write(fruits[1] + "<br>"); // Prints: Banana
    document.write(fruits[2] + "<br>"); // Prints: Mango
    document.write(fruits[fruits.length - 1]); // Prints: Papaya
    </script>
</body>
</html>
```

Apple
Banana
Mango
Papaya

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

JavaScript

Βρόχοι στοιχείων μιας συστοιχίας.

Για να αποκτήσετε πρόσβαση σε όλα τα στοιχεία μιας συστοιχίας με διαδοχική σειρά, μπορείτε να χρησιμοποιήσετε τους βρόχους, ως εξής:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Loop Through an Array Using For Loop</title>
</head>
<body>
    <script>
    var fruits = ["Apple", "Banana", "Mango", "Orange", "Papaya"];

    // Iterates over array elements
    for(var i = 0; i < fruits.length; i++){
        document.write(fruits[i] + "<br>"); // Print array element
    }
    </script>
</body>
</html>
```

Apple
Banana
Mango
Orange
Papaya

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

Συστοιχίες JavaScript

Προσθήκη νέων στοιχείων σε μια συστοιχία

Για να προσθέσετε ένα νέο στοιχείο στο τέλος ενός πίνακα, απλά χρησιμοποιήστε τη μέθοδο push(), ως εξής:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Add a New Element at the End of an Array</title>
</head>
<body>
    <script>
    var colors = ["Red", "Green", "Blue"];
    colors.push("Yellow");

    document.write(colors + "<br>"); // Prints: Red,Green,Blue,Yellow
    document.write(colors.length); // Prints: 4
    </script>
</body>
</html>
```

```
Red,Green,Blue,Yellow
4
```

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

JavaScript Arrays

Removing Elements from an Array

To eliminate the last element from an array you can use the pop() method. This method returns the value that was popped out.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Remove the Last Element from an Array</title>
</head>
<body>
    <script>
    var colors = ["Red", "Green", "Blue"];
    var last = colors.pop();

    document.write(last + "<br>"); // Prints: Blue
    document.write(colors.length); // Prints: 2
    </script>
</body>
</html>
```

```
Blue
2
```

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

Συστοιχίες JavaScript

Προσθήκη ή αφαίρεση στοιχείων σε οποιαδήποτε θέση

Η μέθοδος splice() είναι μια πολύ ευέλικτη μέθοδος συστοιχίας που επιτρέπει την προσθήκη ή αφαίρεση στοιχείων από οποιοδήποτε ευρετήριο, χρησιμοποιώντας τη διάταξη σύνταξης `splice(startIndex, deleteCount, elem1, ..., elemN)`

Αυτή η μέθοδος παίρνει τρεις παραμέτρους: η πρώτη είναι ο δείκτης στον οποίο θα ξεκινήσει η σύζευξη της συστοιχίας, απαιτείται. Η δεύτερη είναι ο αριθμός των στοιχείων που πρέπει να αφαιρεθούν (το 0 πρέπει να χρησιμοποιηθεί σε περίπτωση που ο προγραμματιστής δεν θέλει να αφαιρέσει κανένα στοιχείο), είναι προαιρετική. Και η τρίτη παράμετρος είναι ένα σύνολο στοιχείων αντικατάστασης, είναι επίσης προαιρετική.

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

Συστοιχίες JavaScript

Προσθήκη ή αφαίρεση στοιχείων σε οποιαδήποτε θέση

Η μέθοδος splice() επιστρέφει μια συστοιχία των διαγραμμένων στοιχείων, ή μια κενή συστοιχία αν δεν διαγράφηκαν στοιχεία, όπως φαίνεται στην *Εικόνα 81*. Εάν το δεύτερο όρισμα απουσιάζει, όλα τα στοιχεία από την αρχή έως το τέλος του πίνακα αφαιρούνται. Σε αντίθεση με τις μεθόδους slice() και concat(), η μέθοδος splice() τροποποιεί τη συστοιχία στην οποία γίνεται κλήση.

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

ΣυστοιχίεςJavaScript

Δημιουργία συμβολοσειράς από συστοιχία

- Μπορεί να υπάρχουν καταστάσεις στις οποίες ένας προγραμματιστής απλά σκοπεύει να δημιουργήσει μια συμβολοσειρά με τη σύνδεση των στοιχείων μιας συστοιχίας. Για να γίνει αυτό, μπορεί να χρησιμοποιήσει τη μέθοδο join(). Αυτή η μέθοδος παίρνει μια προαιρετική παράμετρο η οποία είναι μια διαχωριστική συμβολοσειρά που προστίθεται ανάμεσα σε κάθε στοιχείο. Αν παραλείψετε το διαχωριστικό, τότε η JavaScript θα χρησιμοποιήσει κόμμα (,) από προεπιλογή.

- Μια συστοιχία μπορεί επίσης να μετατραπεί σε μια συμβολοσειρά διαχωρισμένη με κόμμα χρησιμοποιώντας το toString(). Αυτή η μέθοδος δεν επιτρέπει την παράμετρο διαχωρισμού ως join().

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

Συστοιχίες JavaScript

Συγχώνευση δύο ή περισσότερων συστοιχιών

- Η μέθοδος concat() μπορεί να χρησιμοποιηθεί για τον συνδυασμό δύο ή περισσότερων συστοιχιών. Αυτή η μέθοδος δεν αλλάζει τις επικρατούσες συστοιχίες, αντ 'αυτού επιστρέφει μια νέα συστοιχία.
- Η μέθοδος concat() μπορεί να λάβει οποιονδήποτε αριθμό παραμέτρων πίνακα, έτσι ένας πίνακας μπορεί να δημιουργηθεί από οποιονδήποτε αριθμό άλλων πινάκων, όπως φαίνεται στο ακόλουθο παράδειγμα:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Merge Multiple Arrays</title>
</head>
<body>
    <script>
    var pets = ["Cat", "Dog", "Parrot"];
    var wilds = ["Tiger", "Wolf", "Zebra"];
    var bugs = ["Ant", "Bee"];

    // Creating new array by combining pets, wilds and bugs arrays
    var animals = pets.concat(wilds, bugs);
    document.write(animals); // Prints: Cat,Dog,Parrot,Tiger,Wolf,Zebra,Ant,Bee
    </script>
</body>
</html>
```

```
Cat,Dog,Parrot,Tiger,Wolf,Zebra,Ant,Bee
```

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

Συστοιχίες JavaScript

Αναζήτηση μέσα σε μια συστοιχία

- Οι μέθοδοι indexOf() και lastIndexOf() μπορούν να χρησιμοποιηθούν για την αναζήτηση μιας συστοιχίας για μια συγκεκριμένη τιμή. Εάν βρεθεί η τιμή, και οι δύο μέθοδοι επιστρέφουν ένα δείκτη που αντιπροσωπεύει το στοιχείο πίνακα. Αν δεν βρεθεί η τιμή, επιστρέφεται -1.
- Η μέθοδος indexOf() επιστρέφει την πρώτη που βρέθηκε, αν και η lastIndexOf() επιστρέφει την τελευταία που βρέθηκε. Και οι δύο μέθοδοι αναγνωρίζουν επίσης μια προαιρετική ακέραια παράμετρο από το ευρετήριο, η οποία καθορίζει το δείκτη εντός του πίνακα στον οποίο θα ξεκινήσει η αναζήτηση.

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

Συστοιχίες JavaScript

Αναζήτηση μέσα σε μια συστοιχία

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Search an Array for a Specific Value</title>
</head>
<body>
    <script>
    var fruits = ["Apple", "Banana", "Mango", "Orange", "Papaya"];

    document.write(fruits.indexOf("Apple") + "<br>"); // Prints: 0
    document.write(fruits.indexOf("Banana") + "<br>"); // Prints: 1
    document.write(fruits.indexOf("Pineapple")); // Prints: -1
    </script>
</body>
</html>
```

```
0
1
-1
```

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

Συστοιχίες JavaScript

Αναζήτησε μέσα σε μια συστοιχία

Η μέθοδος include() μπορεί επίσης να χρησιμοποιηθεί για να μάθετε αν ένας πίνακας περιλαμβάνει ένα συγκεκριμένο στοιχείο ή όχι. Αυτή η μέθοδος παίρνει τις ίδιες παραμέτρους με τις μεθόδους indexOf() και lastIndexOf(), αλλά επιστρέφει τις τιμές αληθές ή ψευδές αντί για αριθμητικούς δείκτες.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Find Whether an Array Includes a Certain Value</title>
</head>
<body>
    <script>
    var arr = [1, 0, 3, 1, false, 5, 1, 4, 7];

    document.write(arr.includes(1) + "<br>"); // Prints: true
    document.write(arr.includes(6) + "<br>"); // Prints: false
    document.write(arr.includes(1, 2) + "<br>"); // Prints: true
    document.write(arr.includes(3, 4)); // Prints: false
    </script>
</body>
</html>
```

```
true
false
true
false
```

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

Συστοιχίες JavaScript Ar

Αναζήτηση μέσα σε μια συστοιχία

- Για να αναζητήσετε μια συστοιχία με βάση μια συγκεκριμένη κατάσταση μπορεί να χρησιμοποιηθεί η μέθοδος JavaScript find(), η οποία έχει εισαχθεί πρόσφατα στο ES6. Αυτή η μέθοδος επιστρέφει την τιμή του πρώτου στοιχείου στη συστοιχία που πληροί την παρεχόμενη λειτουργία δοκιμής. Αν όχι, επιστρέφει την τιμή undefined.
- Η μέθοδος find() αναζητά απλά το πρώτο στοιχείο που πληροί την παρεχόμενη λειτουργία δοκιμής. Ακόμα, εάν ο προγραμματιστής σκοπεύει να ανακαλύψει όλα τα αντιστοιχισμένα στοιχεία, μπορεί να χρησιμοποιήσει τη μέθοδο filter(). Η μέθοδος αυτή δημιουργεί μια νέα συστοιχία με όλα τα στοιχεία που περνούν επιτυχώς τη δοσμένη δοκιμή, όπως μπορεί να εξετασθεί σε αυτό το παράδειγμα.

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

Συστοιχίες JavaScript Sorting

Ταξινόμηση συστοιχιών αντικειμένων JavaScript

- Η ταξινόμηση είναι μια δημοφιλής εργασία όταν εργάζεστε με συστοιχίες.
- Μπορεί να χρησιμοποιηθεί, για παράδειγμα, για την εμφάνιση των ονομάτων των πόλεων ή των χωρών με αλφαβητική σειρά.
- Το αντικείμενο JavaScript Array έχει μια ενσωματωμένη μέθοδο sorting() για την ταξινόμηση στοιχείων μιας συστοιχίας με αλφαβητική σειρά.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Sort an Array Alphabetically</title>
</head>
<body>
    <script>
    var fruits = ["Banana", "Orange", "Apple", "Papaya", "Mango"];
    var sorted = fruits.sort();

    document.write(fruits + "<br>"); // Outputs: Apple,Banana,Mango,Orange,Papaya
    document.write(sorted); // Outputs: Apple,Banana,Mango,Orange,Papaya
    </script>
</body>
</html>
```

```
Apple,Banana,Mango,Orange,Papaya
Apple,Banana,Mango,Orange,Papaya
```

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

Ταξινόμηση συστοιχιών αντικειμένων JavaScript

- Για να **αντιστραφεί η σειρά των στοιχείων** ενός πίνακα, μπορεί να χρησιμοποιηθεί η reverse() μέθοδος. Αντιστρέφει μια συστοιχία με τέτοιο τρόπο ώστε το πρώτο στοιχείο συστοιχίας να γίνεται το τελευταίο, και αντίστροφα. Η μέθοδος sort() και reverse() αλλάζει την αρχική συστοιχία και επιστρέφει μια αναφορά (reference) στην ίδια συστοιχία, όπως μπορεί να εξετασθεί στο παράδειγμα εδώ.

- Για **τη ταξινόμηση αριθμητικών συστοιχιών**, δεν συνιστάται η χρήση της μεθόδου sort(), καθώς μπορεί να παράγει απροσδόκητα αποτελέσματα. Για το σκοπό αυτό, οι προγραμματιστές θα πρέπει να περνούν μια συνάρτηση σύγκρισης (compare function), καθώς όταν καθορίζεται μια συνάρτηση σύγκρισης, τα στοιχεία πίνακα ταξινομούνται σύμφωνα με την τιμή επιστροφής της συνάρτησης σύγκρισης.

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

Συστοιχίες JavaScript Sorting

- Για **την εύρεση της μέγιστης και ελάχιστης τιμής σε μι συστοιχία**, μπορεί να χρησιμοποιηθεί η μέθοδος apply() σε συνδυασμό με τα Math.max() και Math.min (), όπως παρουσιάζεται στο παράδειγμα αυτό. Η μέθοδος apply() προσφέρει έναν προσβάσιμο τρόπο για να περάσετε τις τιμές πίνακα ως ορίσματα σε μια συνάρτηση που δέχεται πολλαπλά ορίσματα με έναν τρόπο που μοιάζει με πίνακα, αλλά όχι έναν πίνακα. Στη συνέχεια, η δήλωση που προκύπτει Math.max.apply(null, numbers) στο παραπάνω παράδειγμα είναι ισοδύναμη με το Math.max(3, -7, 10, 8, 15, 2).

- Τέλος, για την **ταξινόμηση μιας σειράς αντικειμένων** μπορεί να χρησιμοποιηθεί η μέθοδος sort(). Σε αυτό το παράδειγμα, μπορείτε να δείτε πώς να ταξινομείτε μια σειρά αντικειμένων με τιμές ιδιοτήτων.

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

Βρόχοι JavaScript

Οι βρόχοι εφαρμόζονται για την εκτέλεση του ίδιου μπλοκ κώδικα ξανά και ξανά εάν συναντάται μια συγκεκριμένη προϋπόθεση. Η βασική ιδέα πίσω από ένα βρόχο είναι να μηχανοποιήσει τις επαναλαμβανόμενες εργασίες μέσα σε ένα πρόγραμμα για να εξοικονομήσει χρόνο και προσπάθεια. Η JavaScript υποστηρίζει πλέον πέντε διαφορετικούς τύπους βρόχων:

while — βρόχος μέσω ενός μπλοκ κώδικα, αν μια συγκεκριμένη προϋπόθεση *είναι αληθής*

do...while — βρόχος μέσω ενός μπλοκ κώδικα μια φορά; τότε η κατάσταση αξιολογείται. Εάν η προϋπόθεση είναι *αληθής* τότε η δήλωση επαναλαμβάνεται.

for — βρόχος μέσω ενός μπλοκ κώδικα για έναν αριθμό επαναλήψεων.

for…in — βρόχος μέσα από τις ιδιότητες ενός αντικειμένου.

for… — of βρόχος μέσω των τιμών ενός επαναλαμβανόμενου αντικειμένου όπως πίνακες, συστοιχίες κ.λπ.

Βρόχοι JavaScript

Ο βρόχος while

Αυτή είναι η ευκολότερη δήλωση βρόχου που παρέχεται από την JS. Ο βρόχος μέσω ενός μπλοκ κώδικα αν μια καθορισμένη προϋπόθεση *είναι αληθής* Εάν μια προϋπόθεση είναι ψευδής, τότε ο βρόχος διακόπτεται. Η γενική σύνταξη του βρόχου while είναι:

```
while(condition) {
// ο κώδικας προς εκτέλεση
}
```

Στο παράδειγμα αυτό, ένας βρόχος συνεχίζει να λειτουργεί για όσο διάστημα διαρκεί η μεταβλητή $i \leq 5$. Το θα προσαυξάνεται κατά 1 κάθε φορά που τρέχει ο βρόχος. Οι προγραμματιστές πρέπει να διασφαλίζουν ότ μια προϋπόθεση που καθορίζεται στο βρόχο μπορεί τελικά να επιστραφεί ως ψευδής. Αν όχι, ο βρόχος δεν θα σταματήσει ποτέ να επαναλαμβάνεται (άπειρος βρόχος/infinite loop).

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

Βρόχοι JavaScript

Ο βρόχος while

Ο βρόχος do-while είναι μια παραλλαγή του βρόχου while, η οποία αξιολογεί την κατάσταση στο τέλος κάθε επανάληψης βρόχου. Με ένα βρόχο do…while, το μπλοκ του κώδικα εκτελείται μία φορά, και στη συνέχεια η κατάσταση αξιολογείται. Εάν η προϋπόθεση είναι *αληθής*, η δήλωση τότε επαναλαμβάνεται εάν πληρούται η καθορισμένη προϋπόθεση. Η κοινή του σύνταξη είναι:

```
do {
// Code to be executed
}
while(condition);
```

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

Βρόχοι JavaScript

Ο βρόχος do…while

Ο κώδικας JavaScript στο παράδειγμα αυτό ορίζει έναν βρόχο που ξεκινά με *i=1*. Στη συνέχεια θα παράγει την έξοδο και θα αυξήσει την τιμή της μεταβλητής i κατά 1. Στη συνέχεια αξιολογείται η κατάσταση και ο βρόχος θα συνεχίσει να λειτουργεί εάν το $i \leq 5$.

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

Βρόχοι JavaScript

Ο βρόχος for

Επαναλαμβάνει ένα μπλοκ κώδικα εάν πληρούται μια συγκεκριμένη προϋπόθεση. Χρησιμοποιείται συνήθως για την εφαρμογή ενός μπλοκ κώδικα για ορισμένες φορές. Η σύνταξή του είναι:

```
for(initialization; condition; increment) {
    // Code to be executed
}
```

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

Βρόχοι JavaScript

Ο βρόχος for…in

- Ο μετρητής βρόχων, δηλαδή η μεταβλητή στον βρόχο for-in, είναι μια συμβολοσειρά και όχι ένας αριθμός. Περιλαμβάνει το όνομα της παρούσας ιδιότητας ή το δείκτη του τρέχοντος στοιχείου συστοιχίας.
-
- Αυτό το παράδειγμα δείχνει πώς να περιηγηθείτε σε όλες τις ιδιότητες ενός αντικειμένου JS.

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

Βρόχοι JavaScript

Ο βρόχος for...of

- Το ES6 παρουσιάζει μια νέα δήλωση for-of που επιτρέπει τη δημιουργία βρόχων πάνω σε δομές δεδομένων που είναι επαναληπτικές όπως οι συστοιχίες ή άλλα επαναλαμβανόμενα αντικείμενα (π.χ. συμβολοσειρές) με πολύ απλό τρόπο. Επιπλέον, ο κώδικας μέσα στον βρόχο εκτελείται για κάθε στοιχείο του επαναλαμβανόμενου αντικειμένου.
- Αυτό το παράδειγμα δείχνει πώς να δημιουργήσετε βρόχους πάνω σε δομές δεδομένων όπως οι συστοιχίες και οι συμβολοσειρές.

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

Συναρτήσεις JavaScript

Μια συνάρτηση είναι ένα σύνολο δηλώσεων που εκτελούν συγκεκριμένα ενέργειες και μπορούν να διατηρηθούν ανεξάρτητα. Οι συναρτήσεις παρέχουν τη δυνατότητα δημιουργίας επαναχρησιμοποιήσιμων πακέτων κώδικα που είναι πιο διαχειρίσιμα και πιο εύκολο να αποσφαλματωθούν.

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

Συναρτήσεις JavaScript

Μερικά πλεονεκτήματα της χρήσης συναρτήσεων είναι τα ακόλουθα:

- **Οι συναρτήσεις μειώνουν την επανάληψη του κώδικα μέσα σε ένα πρόγραμμα** — Μια συνάρτηση επιτρέπει την εξαγωγή ενός συχνά χρησιμοποιούμενο μπλοκ κώδικα σε ένα ενιαίο στοιχείο/αντικείμενο. Με αυτόν τον τρόπο είναι δυνατό να εκτελεστεί η ίδια εργασία καλώντας μια συνάρτηση όπου χρειάζεται μέσα σε ένα σενάριο χωρίς να χρειάζεται να αντιγράψετε και να επικολλήσετε το ίδιο μπλοκ κώδικα ξανά και ξανά.
- **Οι συναρτήσεις καθιστούν πολύ ευκολότερη τη διατήρηση του κώδικα τον κώδικα** — Δεδομένου ότι μια συνάρτηση που δημιουργείται μία φορά μπορεί να εφαρμοστεί πολλές φορές, ενώ οποιεσδήποτε αλλαγές γίνονται μέσα σε μια συνάρτηση εφαρμόζονται αυτόματα σε όλα τα μέρη χωρίς να επηρεάζουν τα αντίστοιχα αρχεία.
- **Οι συναρτήσεις καθιστούν ευκολότερη την απαλλαγή από την εμφάνιση σφαλμάτων** — Όταν το πρόγραμμα χωρίζεται σε συναρτήσεις, εάν προκύψει οποιοδήποτε σφάλμα, ο προγραμματιστής ξέρει ακριβώς ποια συνάρτηση προκαλεί το σφάλμα και πού να το εντοπίσει. Κατά συνέπεια, η διόρθωση σφαλμάτων γίνεται πολύ απλούστερη.

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

Συναρτήσεις JavaScript

Ο ορισμός και η κλήση μιας συνάρτησης

Η δήλωση μιας συνάρτησης ξεκινά με τη λέξη-κλειδί της συνάρτησης, ακολουθούμενη από το όνομα της συνάρτησης που θα δημιουργηθεί, έπειτα από παρενθέσεις και τέλος από τον κώδικα της συνάρτησης μεταξύ αγκύλων {}.

Για να δηλώσετε μια συνάρτηση, ισχύει η ακόλουθη σύνταξη:

```
function functionName() {
    // Code to be executed
}
```

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

Συναρτήσεις JavaScript

Ο ορισμό και η κλήση μιας συνάρτησης

- Αυτό το απλό παράδειγμα εμφανίζει ένα μήνυμα "Hello".
-
- Μόλις οριστεί μια συνάρτηση, μπορεί να κληθεί από οπουδήποτε στο έγγραφο, πληκτρολογώντας το όνομά της ακολουθούμενο από ένα σύνολο παρενθέσεων, όπως το sayHello() στο προαναφερθέν παράδειγμα.

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

Συναρτήσεις JavaScript

Η προσθήκη παραμέτρων στις συναρτήσεις

- Η συνάρτηση displaySum() σε αυτό το παράδειγμα λαμβάνει δύο αριθμούς ως ορίσματα, απλά προσθέστε τους ως ένα και μετά εμφανίστε το αποτέλεσμα στο πρόγραμμα περιήγησης. Δεν υπάρχει όριο για τον καθορισμό των παραμέτρων.
- Ωστόσο, για κάθε καθορισμένη παράμετρο, ένα αντίστοιχο όρισμα χρειάζεται να περάσει στη συνάρτηση όταν καλείται, αν όχι η τιμή της γίνεται απροσδιόριστη.

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

Συναρτήσεις JavaScript

Προεπιλεγμένες τιμές για παραμέτρους συνάρτησης

Με το ES6, είναι δυνατόν να καθοριστούν προεπιλεγμένες τιμές για τις παραμέτρους λειτουργίας. Αυτό σημαίνει ότι εάν δεν παρέχονται ορίσματα για να λειτουργήσουν όταν η ES6 καλείται, τότε θα χρησιμοποιηθούν οι προεπιλεγμένες τιμές παραμέτρων. Αυτό το παράδειγμα είναι αρκετά επεξηγηματικό σχετικά με το πόσο χρήσιμη είναι αυτή η δυνατότητα, δεδομένου ότι για να επιτευχθεί το ίδιο αποτέλεσμα, η προηγούμενη διαδικασία ήταν η ίδια.

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

Συναρτήσεις JavaScript

Επιστροφή τιμών από μια συνάρτηση

- Μια συνάρτηση μπορεί να επιστρέψει μια τιμή πίσω στο σενάριο που κάλεσε τη συνάρτηση ως αποτέλεσμα, χρησιμοποιώντας τη δήλωση επιστροφής. Η τιμή μπορεί να είναι οποιουδήποτε τύπου (π.χ. συστοιχίες και αντικείμενα). Η δήλωση επιστροφής (return statement) τυπικά τοποθετείται στην τελευταία γραμμή της συνάρτησης πριν από το κλείσιμο της αγκύλης και τελειώνει με ένα ερωτηματικό, όπως φαίνεται στο παράδειγμα που παρατίθεται εδώ.
- Μια συνάρτηση δεν μπορεί να επιστρέψει πολλαπλές τιμές. Παρόλα αυτά, παρόμοια αποτελέσματα μπορούν να ληφθούν επιστρέφοντας μια σειρά τιμών, όπως παρουσιάζεται εδώ.

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

Συναρτήσεις JavaScript

Δουλεύοντας με τις εκφράσεις συναρτήσεων

- Η σύνταξη που χρησιμοποιήθηκε πριν για τη δημιουργία συναρτήσεων ονομάζεται **δήλωση συναρτήσεων (function declaration)**. Υπάρχει μια άλλη σύνταξη για την οικοδόμηση μιας συνάρτησης – έκφρασης συνάρτησης. Μόλις αποθηκευτεί σε μια μεταβλητή, η μεταβλητή μπορεί να χρησιμοποιηθεί ως συνάρτηση.
- Η σύνταξη της δήλωσης συνάρτησης (function declaration) και της έκφρασης λειτουργίας (function expression) φαίνεται πολύ παρόμοια, αλλά ποικίλλουν στον τρόπο με τον οποίο αξιολογούνται. Όπως παρατηρήθηκε στο προηγούμενο παράδειγμα, η έκφραση συνάρτησης έδωσε μια εξαίρεση όταν έγινε επίκληση πριν οριστεί, αλλά η δήλωση συνάρτησης εκτελέστηκε αποτελεσματικά.
- Η JS αναλύει τη δήλωση συνάρτησης πριν από την εκτέλεση του προγράμματος. Ως εκ τούτου, δεν κάνει καμία διαφορά εάν το πρόγραμμα επικαλείται τη λειτουργία πριν από τον ορισμό της, καθώς η JavaScript έχει ανυψώσει τη λειτουργία στην κορυφή του τρέχοντος πεδίου στο παρασκήνιο. Μια έκφραση συνάρτησης δεν αξιολογείται μέχρι να ανατεθεί σε μια μεταβλητή. Κατά συνέπεια, εξακολουθεί να είναι απροσδιόριστη όταν γίνεται επίκληση.

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

Συναρτήσεις JavaScript

Κατανόηση του εύρους μιας μεταβλητής

- Οι μεταβλητές μπορούν να δηλωθούν οπουδήποτε στο JS. Ωστόσο, η τοποθεσία της δήλωσης θα καθορίσει την έκταση της διαθεσιμότητας μιας μεταβλητής εντός του προγράμματος JavaScript – αυτή η διαδικασία ονομάζεται επίσης **μεταβλητό πεδίο εφαρμογής**.
- Από προεπιλογή, οι μεταβλητές που δηλώνονται εντός μιας συνάρτησης έχουν τοπική εμβέλεια, πράγμα που σημαίνει ότι δεν μπορούν να προβληθούν ή να ελεγχθούν εκτός αυτής της συνάρτησης, όπως φαίνεται εδώ.
- Αν και, οποιεσδήποτε μεταβλητές που δηλώνονται σε ένα πρόγραμμα εκτός μιας συνάρτησης έχουν καθολική εμβέλεια, όπου αυτό το σενάριο βρίσκεται σχετικά με τη συνάρτηση, όπως μπορεί να ελεγχθεί εδώ. (??)

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

Αντικείμενα JavaScript

- Η JavaScript είναι μια γλώσσα που βασίζεται σε αντικείμενα και σχεδόν τα πάντα είναι αντικείμενα ή ενεργούν σαν αντικείμενα. Έτσι, για να συνεργαστούν με τον JS με επιτυχία και αποτελεσματικότητα, οι προγραμματιστές πρέπει να κατανοήσουν πώς λειτουργούν τα αντικείμενα, καθώς και πώς να δημιουργήσουν αντικείμενα και να τα χρησιμοποιήσουν.
-
- Ένα αντικείμενο JavaScript είναι απλά μια συλλογή από ονομαστικές τιμές. Συνήθως αναφέρονται ως ιδιότητες του αντικειμένου. Όντας ένας πίνακας με μια συλλογή τιμών, στην οποία κάθε τιμή έχει ένα δείκτη (ένα αριθμητικό κλειδί) ξεκινώντας από το μηδέν και αυξάνοντας κατά ένα για κάθε τιμή. Ένα αντικείμενο είναι σαν μια συστοιχία, αλλά η διαφορά είναι ότι ο προγραμματιστής ορίζει τα κλειδιά, (όνομα, ηλικία, φύλο, κ.λπ.).

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

Αντικείμενα JavaScript

Δημιουργία ενός αντικειμένου

- Οι προγραμματιστές μπορούν να δημιουργήσουν αντικείμενα με αγκύλες, συμπεριλαμβανομένης μιας προαιρετικής λίστας ιδιοτήτων επίσης. Μια ιδιότητα μπορεί να είναι ζεύγος “key:value”, στο οποίο το κλειδί (ή το *όνομα της ιδιότητας*) είναι πάντα μια συμβολοσειρά, και η τιμή (ή η *τιμή της ιδιότητας*) μπορεί να είναι οποιοσδήποτε τύπος δεδομένων (συμβολοσειρές, αριθμοί, Booleans, πίνακες, λειτουργίες κ.λπ.).
- Επιπλέον, οι ιδιότητες με συναρτήσεις όπως οι τιμές τους συχνά ονομάζονται μέθοδοι, για μπορούμε να τις διακρίνουμε εύκολα από άλλα χαρακτηριστικά. Ένα αντικείμενο JS μπορεί να έχει ως εξής. Αυτό το παράδειγμα δημιουργεί ένα αντικείμενο που ονομάζεται person, το οποίο έχει 3 ιδιότητες (όνομα, ηλικία και φύλο) και τη μέθοδο displayName().

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

Αντικείμενα JavaScript

Δημιουργία ενός αντικειμένου

- Αυτή η μέθοδος εμφανίζει την τιμή του this.name, το οποίο συμφωνεί με το person.name. Αυτός είναι ο ευκολότερος και ιδανικότερος τρόπος για να δημιουργήσετε ένα νέο αντικείμενο στο JF, το οποίο είναι γνωστό ως **object literals syntax**.
- Τα ονόματα των ιδιοτήτων γενικά δεν χρειάζεται να αναφέρονται εκτός αν είναι δεσμευμένες λέξεις, ή αν περιέχουν κενά ή ειδικούς χαρακτήρες (οτιδήποτε άλλο εκτός από γράμματα, αριθμούς και τους χαρακτήρες _ και $), ή αν ξεκινούν με έναν αριθμό, όπως φαίνεται εδώ.

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

Αντικείμενα JavaScript

Πρόσβαση στις ιδιότητες ενός αντικειμένου

- Για την πρόσβαση ή τη λήψη της τιμής μιας ιδιότητας, μπορεί να εφαρμοστεί η τελεία (.), καθώς και η σημειογραφία αγκύλων ([]), όπως φαίνεται στο παράδειγμα αυτό. Η εγγραφή και ανάγνωση της σημειογραφίας της τελείας είναι απλούστερη, αλλά δεν μπορεί να χρησιμοποιηθεί. Εάν το όνομα της ιδιότητας δεν είναι έγκυρο (για παράδειγμα, εάν περιέχει ειδικούς χαρακτήρες ή κενά), δεν μπορεί να χρησιμοποιηθεί ο σημειογραφία της τελείας, αλλά η σημειογραφία της αγκύλης.

- Παρουσιάζει πολύ μεγαλύτερη ευελιξία από τη σημειογραφία της τελείας και επιπλέον επιτρέπει τον εντοπισμό των ονομάτων των ιδιοτήτων (property names) ως μεταβλητές για την αντικατάσταση των γραμμάτων συμβολοσειράς (string literals).

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

Αντικείμενα JavaScript

Δημιουργία Βρόχων μέσα από τις Ιδιότητες ενός Αντικειμένου

Οι προγραμματιστές μπορούν να επαναλάβουν ενέργειες μέσα από τα ζεύγη κλειδιών-τιμών ενός αντικειμένου χρησιμοποιώντας το είδος βρόχου for...in. Ο βρόχος for...in υποστηρίζει την επανάληψη εργασιώ μέσα από τις ιδιότητες ενός αντικειμένου, όπως φαίνεται στο παράδειγμα εδώ.

Ορισμός ιδιοτήτων αντικειμένου

Παρομοίως, μπορούν να οριστούν νέες ιδιότητες ή να ενημερωθεί η υπάρχουσα χρησιμοποιώντας τη σημειογραφία dot (.) ή bracket ([]), όπως φαίνεται στο παράδειγμα εδώ.

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

Αντικείμενα JavaScript

Διαγραφή ιδιότητας αντικειμένου

- Ο τελεστής διαγραφής (delete operator) μπορεί να εφαρμοστεί για την πλήρη διαγραφή ιδιοτήτων από ένα αντικείμενο. Η αφαίρεση είναι ο μόνος τρόπος για να ξεφορτωθείς μια ιδιότητα. Η ρύθμιση μιας ιδιότητας ως απροσδιόριστης ή μηδενικής απλώς αλλάζει την τιμή της, όμως δεν αφαιρεί την ιδιότητα από το αντικείμενο.

- Έτσι, δεν έχει καμία επίδραση στις μεταβλητές ή στις δηλωμένες συναρτήσεις. Ωστόσο, οι προγραμματιστές πρέπει να αποφεύγουν τον τελεστή διαγραφής για σκοπούς διαγραφής στοιχείων ενός πίνακα, καθώς δεν τροποποιεί το μήκος της συστοιχίας, απλά ρίχνει μια τρύπα σε αυτήν.

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

Μέθοδοι Κλήσης Αντικειμένου

Η μέθοδος κλήσης ενός αντικειμένου μπορεί να προσπελαστεί με τον ίδιο τρόπο που γίνονται προσβάσιμες οι ιδιότητες - χρησιμοποιώντας δηλαδή τη σημειογραφία της τελείας ή εφαρμόζοντας τη σημειογραφία των αγκύλων, όπως μπορεί να εξετασθεί στο παράδειγμα που παρατίθεται εδώ .

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

Αντικείμενα JavaScript

Χειρισμός βάσει Τιμής (Value) έναντι Αναφοράς (Reference)

Τα αντικείμενα JS είναι τύποι αναφοράς, πράγμα που σημαίνει ότι όταν ένας προγραμματιστής δημιουργεί τα αντίγραφά τους, αντιγράφουν ουσιαστικά τις αναφορές αυτών των αντικειμένων, ενώ οι πρωτόγονες τιμές (primitive values) όπως οι συμβολοσειρές και οι αριθμοί εκχωρούνται ή αντιγράφονται ως ακέραιες τιμές. Αυτό το παράδειγμα είναι αρκετά ενδεικτικό της έννοιας αυτής.

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

Χειρισμός βάσει Τιμής (Value) έναντι Αναφοράς (Reference)

- Όπως μπορούμε να παρατηρήσουμε στο παράδειγμα αυτό, δημιουργήθηκε ένα αντίγραφο ενός μεταβλητού μηνύματος το οποίο άλλαξε την τιμή του ίδιου του αντίγραφού του. Και οι δύο μεταβλητές παραμένουν διακριτές και ξεχωριστές. Ωστόσο, εάν η ίδια αρχή εφαρμοστεί σε ένα αντικείμενο, τότε το αποτέλεσμα θα είναι διαφορετικό.
- Έτσι, οποιεσδήποτε αλλαγές γίνονται στην μεταβλητή χρήστη (variable user) παρεμβαίνουν επίσης στη μεταβλητή ατόμου (object variable), καθώς και οι δύο μεταβλητές αναφέρονται στο ίδιο αντικείμενο. Επομένως, μια απλή αντιγραφή του αντικειμένου δεν το κλωνοποιεί πραγματικά, αλλά αντιγράφει ουσιαστικά την αναφορά σε αυτό το αντικείμενο.

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

ΑΣ ΕΞΑΣΚΗΘΟΥΜΕ ΛΟΙΠΟΝ!

Ανοίξτε τον ακόλουθο σύνδεσμο για να εξασκηθείτε σε μερικές από τις έννοιες που έχετε αποκτήσει μέχρι τώρα:

https://www.w3resource.com/javascript-exercises/javascript-basic-exercises.php

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

code4sp
coding for social promotion

ΕΥΧΑΡΙΣΤΟΥΜΕ!

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