



Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μάθημα: Τεχνικές Προγραμματισμού Υπολογιστών. (Εργαστηριακό μάθημα)

Καθηγητής : Πεφάνης Ευάγγελος

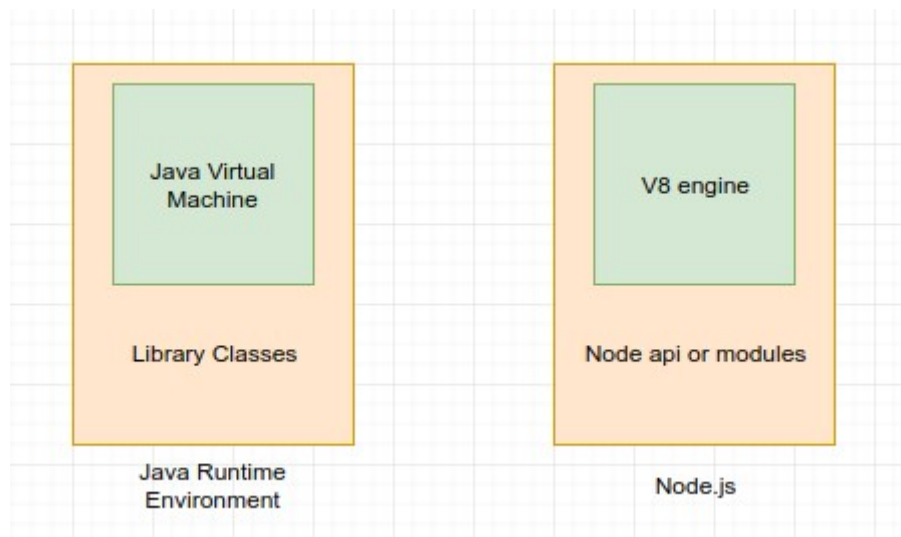
1) Εργαστηριακές σημειώσεις στην γλώσσα προγραμματισμού Java Script.

Η JavaScript είναι μια πολύ ισχυρή γλώσσα. Μπορεί να χρησιμοποιηθεί σε οποιοδήποτε πρόγραμμα περιήγησης στον κόσμο. Επιπλέον, μπορεί να χρησιμοποιηθεί για την εγγραφή κώδικα διακομιστή (Server) με χρήση του `node.js`.

Node.js

Το Node.js είναι ένα περιβάλλον εκτέλεσης του JavaScript. Ακούγεται υπέροχο, αλλά τι σημαίνει αυτό; Πώς λειτουργεί αυτό;

Το περιβάλλον εκτέλεσης Node.js περιλαμβάνει όλα όσα χρειάζεστε για να εκτελέσετε ένα πρόγραμμα γραμμένο σε JavaScript.



Αν γνωρίζετε την Java, σε αυτή την εικόνα φαίνεται μια μικρή αναλογία.

Το Node.js τέθηκε σε λειτουργία όταν οι αρχικοί προγραμματιστές της JavaScript την επέκτειναν από κάτι που θα μπορούσατε να εκτελέσετε μόνο στο πρόγραμμα περιήγησης σε κάτι που θα μπορούσε να τρέξει στο μηχάνημά σας ως αυτόνομη εφαρμογή.

Τώρα μπορείτε να κάνετε πολλά περισσότερα με την JavaScript από το να κάνετε μόνο διαδραστικές ιστοσελίδες.

Η JavaScript έχει τώρα τη δυνατότητα να κάνει πράγματα που μπορούν να κάνουν άλλες γλώσσες δέσμης ενεργειών όπως η Python.

Τόσο το πρόγραμμα περιήγησης JavaScript όσο και το Node.js εκτελούνται στον μηχανισμό εκτέλεσης V8 JavaScript. Αυτός ο μηχανισμός λαμβάνει τον κώδικα JavaScript και τον μετατρέπει σε πιο γρήγορο κώδικα μηχανής. Ο κωδικός μηχανής είναι κώδικας χαμηλού επιπέδου ο οποίος μπορεί να τρέξει στον υπολογιστή χωρίς να χρειάζεται να το ερμηνεύσει για πρώτη φορά.

Γιατί Node.js;

Εδώ είναι ένας επίσημος ορισμός που δίνεται στην επίσημη [ιστοσελίδα](#) Node.js :

Το Node.js® είναι ένα runtime JavaScript που βασίζεται στη μηχανή JavaScript V8 του Chrome.

Το Node.js χρησιμοποιεί ένα μοντέλο εισόδου / εξόδου μη αποκλεισμού, το οποίο καθιστά ελαφρύ και αποτελεσματικό.

Το οικοσύστημα πακέτων Node.js, **npm**, είναι το μεγαλύτερο οικοσύστημα βιβλιοθηκών ανοικτού κώδικα στον κόσμο.

Η V8 επιταχύνει την JavaScript με τη χρήση C ++

Το V8 είναι μια μηχανή χρόνου εκτέλεσης ανοικτού κώδικα γραμμένη σε C ++, και χρησιμοποιείται στην μηχανή αναζήτησης του browser **Google Chrome**.

JavaScript -> V8 (C ++) -> **Κώδικας μηχανής**

Το V8 υλοποιεί μια δέσμη ενεργειών που ονομάζεται ECMAScript όπως ορίζεται στην ECMA-262. Το ECMAScript δημιουργήθηκε από την Ecma International για την τυποποίηση του JavaScript.

Το Ecma είναι ένας οργανισμός τυποποίησης για συστήματα πληροφοριών και επικοινωνιών. Πήρε το σημερινό της όνομα το 1994, όταν η Ευρωπαϊκή Ένωση Κατασκευαστών Ηλεκτρονικών Υπολογιστών (ECMA) άλλαξε το όνομά της για να αντικατοπτρίζει την παγκόσμια εμβέλεια και τις δραστηριότητες της οργάνωσης. Ο οργανισμός ιδρύθηκε το 1961 για να τυποποιήσει τα συστήματα ηλεκτρονικών υπολογιστών στην Ευρώπη.

Το V8 μπορεί να τρέξει αυτόνομο ή μπορεί να ενσωματωθεί σε οποιαδήποτε εφαρμογή C ++. Έχει συνδέσμους μέσω διαδικασιών που σας επιτρέπουν να γράψετε τον δικό σας κώδικα C ++ που μπορείτε να διαθέσετε στη Javascript.

Αυτό ουσιαστικά σας επιτρέπει να προσθέσετε λειτουργίες σε JavaScript, ενσωματώνοντας τον κώδικα V8 στον κώδικα C ++, έτσι ώστε ο κώδικας C ++ σας να κατανοεί περισσότερα από ό, τι προδιαγράφει το πρότυπο ECMAScript.

Το Node.js είναι μια πλατφόρμα ανάπτυξης λογισμικού (κυρίως διακομιστών) χτισμένη σε περιβάλλον JavaScript. Στόχος του είναι να παρέχει ένα εύκολο τρόπο δημιουργίας κλιμακωτών διαδικτυακών εφαρμογών.

Σε αντίθεση από τα περισσότερα σύγχρονα περιβάλλοντα ανάπτυξης εφαρμογών δικτύων μία διεργασία node δεν στηρίζεται στην πολυνηματικότητα (multithreading) αλλά σε ένα μοντέλο ασύγχρονης επικοινωνίας εισόδου/εξόδου.

Αυτό του είδους το μοντέλο λειτουργίας στοχεύει στη βελτίωση της διεκπεραιωτικής ικανότητας των διαδικτυακών εφαρμογών με πολλές λειτουργίες εισόδου/εξόδου, όπως και εφαρμογών ιστού πραγματικού χρόνου (προγράμματα επικοινωνίας πραγματικού χρόνου, παιχνίδια φυλλομετρητών).

Η αρχιτεκτονική της πλατφόρμας φέρνει τον προγραμματισμό οδηγούμενο από γεγονότα (event – driven programming) στους εξυπηρετητές, επιτρέποντας την ανάπτυξη γρήγορων διακομιστών σε JavaScript.

Γράφοντας Hello World στο Node.js

Κάντε ένα αρχείο app.js και προσθέστε τα ακόλουθα σε αυτό.

```
console.log("Hello World!");
```

Ανοίξτε το τερματικό και εκτελέστε το app.js δίνοντας στο prompt : > `node app.js`.

Μόλις γράψατε το Hello World στο Node.js.

Περιβάλλοντα υλοποίησης JavaScript

1) JSEditor - Επεξεργαστής κώδικα JavaScript σε απευθείας σύνδεση

Link : <https://jseditor.io/>

Το JSEditor σας βοηθά να δημιουργήσετε πρωτότυπο κώδικα JavaScript, HTML και CSS. Στην αριστερή πλευρά του έργου, υπάρχουν τρεις καρτέλες στις οποίες μπορείτε να πληκτρολογήσετε τη σήμανση HTML, τον κώδικα JavaScript και να επεξεργαστείτε φύλλα στυλ. Για να εκτελέσετε τον κώδικα, απλώς κάντε κλικ στο κουμπί εκτέλεσης και θα δείτε τα αποτελέσματα στον δεξιό πίνακα. Η κονσόλα στο κάτω μέρος της οθόνης εμφανίζει τα αποτελέσματα εκτέλεσης, σφάλματα της γλώσσας javascript καθώς και μηνύματα κονσόλας.

2) Visual Studio Code.

Το Visual Studio Code, είναι ένα πρόγραμμα για την συγγραφή και επεξεργασία κώδικα. Είναι μία από τις προσπάθειες της Microsoft, να συμμετάσχει στον χώρο του Open Source. Βασίζεται στον Atom και πρόκειται για την δική της εκδοχή ενός πολύ καλού text editor.

Ο κώδικας του Visual Studio είναι ένας ελαφρύς αλλά ισχυρός επεξεργαστής πηγαίου κώδικα ο οποίος εκτελείται στην επιφάνεια εργασίας σας και διατίθεται για Windows, MacOS και Linux. Έχει ενσωματωμένη υποστήριξη για JavaScript, TypeScript και Node.js και έχει ένα πλούσιο οικοσύστημα επεκτάσεων για άλλες γλώσσες (όπως C ++, C #, Java, Python, PHP, Go) και runtimes (όπως .NET και Unity).

Περιβάλλον εργασίας:

Το περιβάλλον εργασίας του προγράμματος, είναι ιδιαίτερα καλαίσθητο και λειτουργικό. Το μεγαλύτερο μέρος του καταλαμβάνεται από τον χώρο συγγραφής και επεξεργασίας κώδικα. Περιμετρικά αυτού του χώρου, μπορούν να εμφανίζονται διάφορα πλαίσια με πληροφορίες και βοηθητικά εργαλεία.

Μπορείτε να προσαρμόσετε την διάταξη τους, δημιουργώντας έτσι το περιβάλλον εργασίας που προτιμάτε. Επιπλέον, υποστηρίζεται η χρήση χρωματικών θεμάτων (themes) με τα οποία μπορείτε να αλλάξετε και την εμφάνισή του.

Χαρακτηριστικά και λειτουργίες:

Το πρόγραμμα αν και ανήκει στην κατηγορία των επεξεργαστών κειμένου, διαθέτει αρκετά χαρακτηριστικά τα οποία συναντώνται σε πιο σύνθετα και ολοκληρωμένα προγράμματα.

Ένα από αυτά, είναι το IntelliSense το οποίο επιταχύνει την πληκτρολόγηση με το να προτείνει σχετικές επιλογές για την αυτόματη συμπλήρωση του κώδικα. Είναι ιδιαίτερα αποτελεσματικό και υποστηρίζει διάφορες γλώσσες προγραμματισμού.

Η ανάγνωση του κώδικα βελτιώνεται ιδιαίτερα, χάρη στο syntax highlighting, το οποίο χρωματίζει τα διάφορα μέρη του κώδικα τονίζοντας τα χαρακτηριστικά της κάθε γλώσσας, κάνοντάς τον έτσι πιο ευδιάκριτο και ευανάγνωστο.

Ο ενσωματωμένος debugger, βοηθάει στον εντοπισμό λαθών κατά την εκτέλεση του κώδικα. Επιτρέπει την προσθήκη breakpoints, στα οποία μπορείτε να παρακολουθείτε την λειτουργία του.

Ένα άλλο εργαλείο το οποίο είναι ενσωματωμένο, είναι αυτό για το source control management. Υποστηρίζει τους όλους τους providers, από τους οποίους ο πιο γνωστός και διαδεδομένος είναι το Git. Αυτό σας δίνει την δυνατότητα να εκτελείται τις αντίστοιχες εντολές όπως για παράδειγμα push και pull, απευθείας από το περιβάλλον του προγράμματος, χωρίς να απαιτείται κάποιο επιπλέον πρόγραμμα.

Η υποστήριξη extensions, τα οποία προσθέτουν λειτουργίες και αυξάνουν τις δυνατότητές του, σε συνδυασμό με τις πολλές επιλογές και ρυθμίσεις, κάνουν το πρόγραμμα ιδιαίτερα ευέλικτο.

Έτσι μπορεί να το προσαρμοστεί στο προσωπικό στυλ του κάθε προγραμματιστή ή ακόμα και στις ανάγκες διαφορετικών projects.

Εκμάθηση και χρήση:

Το πρόγραμμα δεν είναι ιδιαίτερα δύσκολο καθώς οι περισσότερες από τις λειτουργίες του είναι εύκολα αντιληπτές και κατανοητές. Στο site του υπάρχει ένα αναλυτικό εγχειρίδιο χρήσης, καθώς και διάφορες άλλες πληροφορίες.

Εγκατάσταση και απαιτήσεις:

Για να εγκαταστήσετε το πρόγραμμα στον υπολογιστή σας, δεν θα χρειαστεί να κάνετε τίποτα περισσότερο από το να ακολουθήσετε τα βήματα του αντίστοιχου οδηγού.

Το πρόγραμμα διατίθεται και σε portable έκδοση, την οποία μπορείτε να κατεβάσετε, να την τοποθετήσετε σε ένα USB Flash Disk και να το χρησιμοποιείτε σε οποιονδήποτε υπολογιστή.

JavaScript

Όταν χρησιμοποιούμε την JavaScript μέσα στο πρόγραμμα περιήγησης, μπορούμε να αλλάξουμε τον τρόπο εμφάνισης μιας ιστοσελίδας και τον τρόπο με τον οποίο συμπεριφέρεται. Θα επικεντρωθούμε κυρίως στην εκμάθηση της ίδιας της γλώσσας και συνεπώς θα χρησιμοποιήσουμε μόνο μία λειτουργία για να εκτυπώσουμε τα αποτελέσματα αυτή την λειτουργία – εντολή ονομάζουμε "console.log".

Μεταβλητές

Στην Javascript δεν γίνεται διαχωρισμός των τύπων των μεταβλητών, αλλά χρησιμοποιείται από κοινού η δέσμευση χώρου στην μνήμη ανεξαρτήτου τύπου περιεχομένου που θα χρησιμοποιήσουμε, μπορούμε δηλαδή να δηλώσουμε μια αόριστη μεταβλητή στην αρχή να της καταχωρήσουμε ακέραιο αριθμό, ύστερα να την χρησιμοποιήσουμε σαν String ή Boolean και αργότερα αλλιώς.

Μία μεταβλητή δηλώνεται με την εντολή «var» και ένα όνομα μεταβλητής το οποίο θα πρέπει αναγκαστικά να είναι σε λατινικούς χαρακτήρες. Μια μεταβλητή πρέπει να χρησιμοποιείται ακριβώς όπως έχει δηλωθεί, έτσι το «var y» δεν είναι ίδιο με το «var Y».

Το «var» προέρχεται από το «variable» που σε ελεύθερη μετάφραση σημαίνει «μεταβλητή».

```
var a,b,c;  
var B;
```

```
a=11;
```

```
b=873.123;  
B="Καλησπέρα!";  
c=true;
```

```
var e=66;  
var f="Καλησπέρα!";
```

Κάθε μεταβλητή έχει ημερομηνία λήξεως και γεννιέται όταν την δηλώνουμε. Αν μια μεταβλητή δηλωθεί μέσα σε μια μέθοδο τότε «ζει» για όσο εκτελείται η μέθοδος, μόλις ολοκληρωθεί, η μεταβλητή διαγράφεται από την μνήμη. Στην περίπτωση οι μεταβλητές αυτές λέγονται local (τοπικές), ενώ αν μια μεταβλητή είναι δηλωμένη εκτός των μεθόδων τότε λέγεται global (παγκόσμιες).

Τελεστές

Η χρησιμότητα των μεταβλητών και άλλων στοιχείων που δημιουργούνται μέσω μιας γλώσσας προγραμματισμού, γίνεται μέσα από την μεταξύ τους αλληλεπίδραση μέσω των πράξεων που μπορούν να γίνουν. Έστω πως έχουμε μια μεταβλητή x και θέλουμε να την πολλαπλασιάσουμε με την μεταβλητή y. Αυτό που θέλουμε να κάνουμε είναι να καταχωρήσουμε το αποτέλεσμα της αλληλεπίδρασης των μεταξύ μεταβλητών x και y που πραγματοποιείται με τον τελεστή του πολλαπλασιασμού.

Υπάρχουν τριών ειδών τελεστές:

- Αριθμητικοί τελεστές : χρησιμοποιούνται με την πράξη μεταξύ δυο μεταβλητών.
- Συγκριτικοί τελεστές : χρησιμοποιούνται με την μεταξύ σύγκριση δυο μεταβλητών η τιμών. Όταν η σύγκριση είναι αληθής επιστρέφεται η τιμή true ενώ όταν δεν είναι επιστρέφεται η τιμή false.
- Λογικοί τελεστές : χρησιμοποιούνται για την σύγκριση δυο λογικών προτάσεων. Παρομοίως το αποτέλεσμα θα είναι true ή false.

Σε αυτή την ενότητα θα μελετήσουμε και τα τρία είδη τελεστών με αναλυτικά παραδείγματα και επεξήγηση πάνω στην λειτουργία τους.

Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές χρησιμοποιούνται για την αλληλεπίδραση μεταξύ δυο μελών είτε είναι μεταβλητών, είτε οποιαδήποτε άλλου τύπου δεδομένων. Είναι το κατώτερο επίπεδο αλληλεπίδρασης μεταξύ στοιχείων. Παρακάτω θα μελετήσουμε τους πιο σημαντικούς.

Πρόσθεση (+)

Από τα μαθηματικά γνωρίζουμε πως αν έχουμε να κάνουμε με δυο αριθμούς, τότε από αυτούς θα προκύψει το άθροισμα τους. Εάν όμως έχουμε να κάνουμε με άλλο τύπο για παράδειγμα αν και τα δυο μέλη είναι κείμενο, τότε το ένα κείμενο θα προστεθεί στο τέλος του άλλου, αν το ένα μέλος είναι αριθμός και το άλλο κείμενο, τότε ο αριθμός θα θεωρηθεί ως κείμενο. Γενικώς όταν ένας αριθμός βρίσκεται μέσα σε εισαγωγικά θεωρείται κείμενο.

Παράδειγμα:

```
var a=5,b=6;  
console.log((a+b));  
var c="Γεια", d="σας!";  
console.log(c+d);  
console.log(a+c);
```

Αποτέλεσμα:

```
11
Γεια σας!
5Γειά
```

Αφαίρεση (-)

Αφαιρεί δυο αριθμούς. Ο τελεστής χρησιμοποιείται αποκλειστικά για αριθμητικές πράξεις ακόμη κι αν ο αριθμός βρίσκεται σε εισαγωγικά.

Παράδειγμα:

```
var a=5,b=6;
console.log((a-b));
var c="20", d="30";
console.log(c-d);
```

Αποτέλεσμα:

```
-1
-10
```

Πολλαπλασιασμός (*)

Χρησιμοποιείται αποκλειστικά για τον πολλαπλασιασμό αριθμών.

Παράδειγμα:

```
var a=5,b=6;
console.log((a*b));
var c="20", d="30";
console.log(c*d);
```

Αποτέλεσμα:

```
30
600
```

Διαίρεση (/)

Χρησιμοποιείται αποκλειστικά για την διαίρεση αριθμών το αποτέλεσμα μπορεί να είναι ακέραιος αριθμός ή δεκαδικός.

Παράδειγμα:

```
var a=5,b=6;
console.log((a/b));
var c="20", d="30";
console.log(c/d);
```

Αποτέλεσμα:

```
0.8333333333333334
0.6666666666666666
```

Υπόλοιπο (%)

Χρησιμοποιείται αποκλειστικά για το υπόλοιπο διαίρεσης μεταξύ δυο αριθμών.

Παράδειγμα:

```
var a=5,b=6;  
console.log((a%b));  
var c="20", d="30";  
console.log(c%d);
```

Αποτέλεσμα:

```
5  
20
```

Μείωση (--) και Αύξηση (++)

Χρησιμοποιούνται αποκλειστικά σε αριθμούς. Αυξάνουν ή μειώνουν αντίστοιχα την τιμή ενός αριθμού κατά ένα. Μπορεί να βρίσκεται αριστερά της μεταβλητής ή δεξιά αυτής. Όταν βρίσκεται αριστερά τότε πρώτα αυξάνεται κατά ένα και ύστερα δίνεται το αποτέλεσμα, αν βρίσκεται δεξιά της μεταβλητής τότε πρώτα δίνεται το αποτέλεσμα με την υπάρχουσα τιμή και ύστερα αυξάνεται η τιμή της κατά ένα.

Παράδειγμα:

```
var a=1;  
console.log("a++: "+(a++));  
console.log("++a: "+(++a));  
console.log("a: "+a);  
console.log("--a: "+(--a));  
console.log("a--: "+(a--));  
console.log("a: "+a);
```

Αποτέλεσμα:

```
a++: 1  
++a: 3  
a: 3  
--a: 2  
a--: 2  
a: 1
```

Συγκριτικοί τελεστές

Οι συγκριτικοί τελεστές χρησιμοποιούνται για να ελέγχουν συγκριτικά αν κάτι ισχύει ή δεν ισχύει. Κάθε τελεστής αποτελείται από αριστερό και δεξιό μέλος τα οποία μεταξύ τους συγκρίνονται με βάση των εκάστοτε τελεστή. Αν η σύγκριση βγει αληθής τότε επιστρέφεται τιμή «true» αν δεν είναι, τότε επιστρέφεται τιμή «false». Παρακάτω δίδεται σχετικός πίνακας με όλους τους συγκριτικούς τελεστές και ένα απλό παράδειγμα χρήσης τους.

Σύμβολο	Επεξήγηση
>	Μεγαλύτερο
<	Μικρότερο
>=	Μεγαλύτερο ή ίσο
<=	Μικρότερο ή ίσο
==	Ίσον
===	Ίσον (με τον ίδιο τύπο μεταβλητής)

```
var a=5,b=2;  
console.log(a>b);  
console.log(a<b);  
console.log(a>=b);
```

```
console.log(a<=b);  
console.log(a==b);  
console.log(a===b);
```

Λογικοί τελεστές

Οι λογικοί τελεστές χρησιμοποιούνται σε συνδυασμό με τους συγκριτικούς όπου μπορούν να συνδυαστούν μεταξύ τους περιπλοκοί και συνδυαστικοί έλεγχοι.

Λογικό ΚΑΙ (&&) - AND

Όταν και τα δυο μέλη είναι true τότε το αποτέλεσμα είναι true αλλιώς είναι false.

```
var a=5,b=6,c=2,d=2;  
console.log((a<b && c==d));
```

Το αποτέλεσμα που μας επιστρέφεται στην κονσόλα από την εκτέλεση του παραπάνω κώδικα είναι:

True

Λογικό Ή (||) - OR

Όταν τουλάχιστον ένα από τα δυο μέλη είναι true τότε το αποτέλεσμα είναι true διαφορετικά το αποτέλεσμα είναι false.

```
var a=5,b=6,c=2,d=2;  
console.log((a<b || c>d));
```

Το αποτέλεσμα που μας επιστρέφεται στην κονσόλα από την εκτέλεση του παραπάνω κώδικα είναι:

True

Λογικό ΌΧΙ (!) - NOT

Όταν εφαρμόζεται σε ένα μέλος αλλάζει την λογική τιμή του. Πχ όταν είναι true γίνεται false και όταν είναι false γίνεται true.

```
var a=5,b=6,c=2,d=2;  
console.log(!(a<b && c==d));
```

Το αποτέλεσμα που μας επιστρέφεται στην κονσόλα από την εκτέλεση του παραπάνω κώδικα είναι:

False

Μαθηματικές πράξεις

Εκτός από τις βασικές πράξεις μεταξύ αριθμών (πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση, υπόλοιπο) υπάρχουν και κάποιες επιπλέον πράξεις οι οποίες αντιπροσωπεύουν μαθηματικές πράξεις όπως ύψωση σε δύναμη, ρίζα κτλ.

Στην javascript για να χρησιμοποιήσουμε αυτού του είδους τις μεταβλητές θα πρέπει να χρησιμοποιήσουμε την βιβλιοθήκη «Math» όπου περιέχει μέσα τις ένα σύνολο από μεθόδους που κάνουν τέτοιου είδους μαθηματικές πράξεις. Για να χρησιμοποιήσουμε μια από τις παρακάτω δυνατότητες θα πρέπει να έχουμε εισάγει αριστερά αυτών το πρόθεμα «Math.» και ύστερα το όνομα της μεθόδου και τις παραμέτρους τις.

Πράξη	Επεξήγηση
pow()	Ύψωση σε δύναμη.
round()	Στρογγυλοποίηση.
ceil()	Στρογγυλοποίηση προς τα πάνω στον κοντινότερο ακέραιο.
floor()	Στρογγυλοποίηση προς τα κάτω στον κοντινότερο ακέραιο.
abs()	Απόλυτη τιμή.
max()	Επιστρέφει την μεγαλύτερη τιμή.
min()	Επιστρέφει την μικρότερη τιμή.
random()	Επιστρέφει έναν ψευδό-τυχαίο αριθμό μεταξύ 0 και 1.
PI()	Επιστρέφει την τιμή του αριθμού π.
E()	Επιστρέφει την τιμή του αριθμού e.
sqrt()	Επιστρέφει την τετραγωνική ρίζα.
sin()	Ημίτονο.
cos()	Συνημίτονο.
atan()	Τόξο εφαπτομένης.
tan()	Εφαπτομένη.
tan()	Εφαπτομένη.
log()	Ο φυσικός λογάριθμος του αριθμού μέσα στις παρενθέσεις.
log()/Math.LN10	Λογάριθμος με βάση το 10.
log()/Math.LN2	Λογάριθμος με βάση το 2.
exp()	Η ύψωση κυβικής δύναμης της τιμής του e.

```

console.log("Δύναμη: "+Math.pow(4,3));
console.log("Στρογγυλοποιημένος: "+Math.round(2.5));
console.log("Προς τα πάνω: "+Math.ceil(2.2));
console.log("Προς τα κάτω: "+Math.floor(2.8));
console.log("Απόλυτος: "+Math.abs(-7.25));
console.log("Μεγαλύτερος: "+Math.max(2,5,10,6,7));
console.log("Μικρότερος: "+Math.min(2,5,10,6,7));
console.log("Τυχαίος: "+Math.random());
console.log("Π: "+Math.PI);
console.log("e: "+Math.E);
console.log("Τετραγωνική ρίζα: "+Math.sqrt(9));
console.log("Ημίτονο: "+Math.sin(3));
console.log("Συνημίτονο: "+Math.cos(3));
console.log("Τόξο εφαπτομένης: "+Math.atan(2));
console.log("Εφαπτομένη: "+Math.tan(9));
console.log("Λογάριθμος: "+Math.log(2));
console.log("Λογάριθμος βάση 10: "+Math.log(25)/Math.LN10);
console.log("Λογάριθμος βάση 2: "+Math.log(30)/Math.LN2);
console.log("e υψωμένο: "+Math.exp(3));

```

Συν άπειρο, πλήν άπειρο

Η javascript δεν εμφανίζει σφάλματα σε περίπτωση υπερχείλισης ή διαίρεσης με το μηδέν. Όταν ένα αποτέλεσμα μιας αριθμητικής/μαθηματικής πράξης είναι μεγαλύτερο από τα άνω όρια που

μπορεί να αναπαρασταθεί, τότε το αποτέλεσμα που επιστρέφει η javascript είναι το λεκτικό «infinity». Παρομοίως όταν το αρνητικό αποτέλεσμα μιας τιμής ξεπεράσει τα κάτω όρια που μπορεί να αναπαρασταθεί, τότε το αποτέλεσμα που επιστρέφει η javascript είναι «-infinity». Οι τιμές αυτές μπορούν να χρησιμοποιηθούν όπως χρησιμοποιούνται και στην καθημερινότητα, να τις προσθέσουμε, να τις διαιρέσουμε, να τις πολλαπλασιάσουμε κ.α.

Όταν μια τιμή που προκύπτει βρίσκεται κοντά στο μηδέν αλλά μακριά από την μικρότερη δυνατή αναπαράσταση αριθμού σε αυτό το εύρος, τότε η javascript επιστρέφει μηδέν. Αν μια τιμή που προκύπτει βρίσκεται κοντά στο μηδέν από αρνητικές τιμές, αλλά μακριά από την μεγαλύτερη δυνατή αναπαράσταση αριθμού σε αυτό το εύρος, τότε η javascript επιστρέφει το λεγόμενο αρνητικό μηδέν. Αυτή η περίπτωση είναι τελείως δυσδιάκριτη από ένα απλό μηδενικό και συνήθως δεν χρησιμοποιείται.

Η διαίρεση με το μηδέν δεν αποτελεί σφάλμα στην javascript, έτσι επιστρέφεται συνήθως άπειρο ή πλην άπειρο. Παρόλα αυτά υπάρχει μια περίπτωση στην οποία η javascript δεν επιστρέφει τιμή, κι αυτή είναι όταν διαιρούμε μηδέν με το μηδέν. Ως αποτέλεσμα επιστρέφεται η τιμή «NaN» (Not a Number), στην προκειμένη δεν μπορούμε να κάνουμε πράξεις με αυτή την τιμή όπως στην περίπτωση του άπειρου κι έτσι ένα script μπορεί να προκύψει προβληματικό λόγω αυτού του λάθους.

Πίνακες

Η JavaScript μπορεί να κρατήσει μια σειρά από μεταβλητές σε ένα αντικείμενο Array (Πίνακας). Στην JavaScript, ένας πίνακας λειτουργεί επίσης ως λίστα.

Για να ορίσετε έναν πίνακα, χρησιμοποιήστε τα σύμβολα [] στον ορισμό ή τη σημείωση αντικειμένου πίνακα:

```
var myArray = [1, 2, 3];  
var theSameArray = new Array(1, 2, 3);
```

Αντιστοίχιση

Μπορούμε να χρησιμοποιήσουμε τον χειριστή αγκύλης για να απευθυνθούμε σε ένα συγκεκριμένο κελί στον πίνακα μας. Η αντιστοίχιση χρησιμοποιεί μηδενικούς δείκτες, για παράδειγμα στο 2ο μέλος μπορεί να αντιστοιχηθεί με το δείκτη 1. Ένα από τα οφέλη από τη χρήση μιας δομής δεδομένων είναι ότι έχετε σταθερή αναζήτηση χρόνου, αν γνωρίζετε ήδη τον δείκτη του στοιχείου προσπαθείτε να έχετε πρόσβαση.[]myArray

```
console.log(myArray[1]); // prints out 2
```

Οι πίνακες του JavaScript είναι δυναμικοί, πράγμα που σημαίνει ότι μπορούμε επίσης να αντιστοιχίσουμε μεταβλητές σε τυχαίες τοποθεσίες, παρόλο που τα προηγούμενα κελιά ήταν απροσδιόριστα. Για παράδειγμα:

```
var myArray = []  
myArray[3] = "hello"  
console.log(myArray);
```

Θα εκτυπωθεί:

```
[undefined, undefined, undefined, "hello"]
```

Στοιχεία πίνακα

Επειδή οι πίνακες στην JavaScript είναι απλά ειδικοί τύποι αντικειμένων, μπορείτε να αποθηκεύσετε στοιχεία διαφορετικών τύπων στον ίδιο πίνακα. Το παρακάτω παράδειγμα είναι ένας πίνακας με μια συμβολοσειρά, έναν αριθμό και ένα κενό αντικείμενο.

```
var myArray = ["string", 10, {}]
```

Προσπέλαση πινάκων

Για να μπορέσουμε να αποκτήσουμε πρόσβαση στα στοιχεία ενός πίνακα χρησιμοποιούμε τις συντεταγμένες του εκάστοτε στοιχείου. Συντεταγμένη ενός στοιχείου μέσα σε ένα πίνακα, είναι η θέση στην οποία βρίσκεται μέσα σε αυτόν. Στους πίνακες δεν έχουμε ποτέ αρνητικές θέσεις παρά μόνο θετικές. Έτσι αν γνωρίζουμε ότι ένα στοιχείο βρίσκεται στην θέση 1 μπορούμε να χρησιμοποιήσουμε το περιεχόμενο του καλώντας το όνομα του πίνακα και την θέση-συντεταγμένη μέσα σε άγκιστρα:

```
var pinakas = ["Στοιχείο 1", "Στοιχείο 2"];  
var timi = pinakas[0];
```

Στο προηγούμενο παράδειγμα χρησιμοποιούμε την πρώτη τιμή του πίνακα «pinakas» για να την εισάγουμε σε μια νέα μεταβλητή με όνομα «timi». Αυτό που θα εισαχθεί ως τιμή στην μεταβλητή αυτή είναι το αλφαριθμητικό «Στοιχείο 1». Με παρόμοια λογική μπορούμε να αλλάξουμε τιμές στα στοιχεία ενός πίνακα ή ακόμη και να προσθέσουμε νέα στοιχεία:

```
var pinakas = ["Στοιχείο 1", "Στοιχείο 2"];  
pinakas[1] = "Stoixeio 2";  
i = 2;  
pinakas[i] = "Stoixeio 3";  
pinakas[i+1] = "Stoixeio 4";
```

Το ιδιαίτερο με τους πίνακες στην javascript είναι πως ανάλογα με το πόσα στοιχεία έχουμε μια δεδομένη στιγμή, αν προσθέσουμε καινούργια αυτόματα το μήκος (length) του πίνακα αυτού θα προσαρμοστεί στο νέο πλήθος.