

# Παιχνίδια σε Javascript

## Μάθημα 1ο – Μια Γρήγορη Εισαγωγή στη Γλώσσα

### Τα Εργαλεία

Την Javascript μπορούμε (όπως και την HTML) να τη γράψουμε σε ένα απλό συντάκτη κειμένου, ή σε ένα περιβάλλον όπως το Bluefish που έχει τη δυνατότητα να συμπληρώνει για εμάς τις εντολές και να τις χρωματίζει. Ωστόσο, σε σχέση με την HTML έχουμε κάποια επιπλέον προβλήματα:

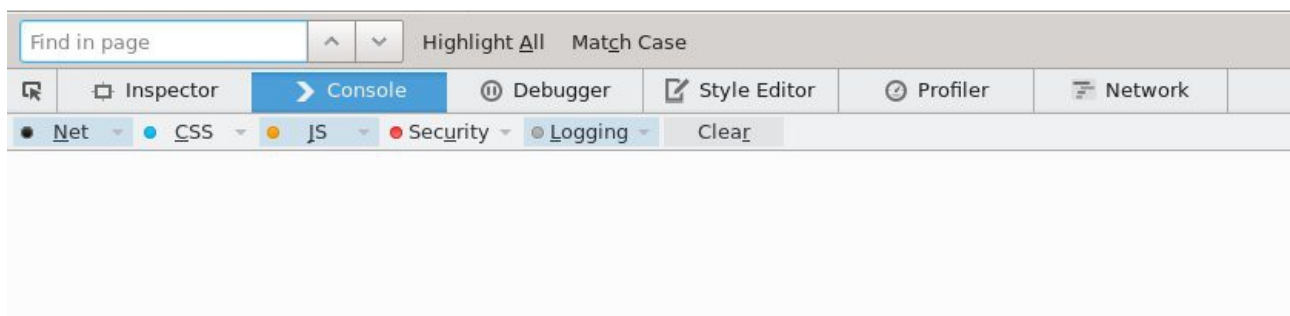
- Η Javascript δεν είναι γλώσσα περιγραφής σελίδας αλλά γλώσσα προγραμματισμού.
- Όταν το πρόγραμμα μας έχει κάποιο λογικό ή συντακτικό λάθος, το αποτέλεσμα στο browser είναι συχνά μια κενή σελίδα.
- Θα πρέπει να έχουμε τον τρόπο να βρούμε το λάθος χρησιμοποιώντας εργαλεία που βρίσκουμε στις άλλες γλώσσες προγραμματισμού. Για παράδειγμα, να βλέπουμε τα μηνύματα λάθους κατά την εκτέλεση ή να μπορούμε να δούμε τις τιμές των μεταβλητών, να εκτελούμε το πρόγραμμα γραμμή – γραμμή κλπ.

Οι παραπάνω δυνατότητες υπάρχουν για κάθε γλώσσα προγραμματισμού με ειδικά προγράμματα που ονομάζονται **debuggers** (εκσφαλματωτές στα Ελληνικά, αλλά αν το λέτε έτσι θα σας κοροϊδεύω).

Στην Javascript όμως, το περιβάλλον εκτέλεσης είναι ο browser! Πως είναι δυνατόν να κάνουμε τα παραπάνω;

### Developer Tools

Και όμως, ο ταπεινός μας browser περιέχει τα εργαλεία για να δούμε τα μηνύματα και να διορθώσουμε τον κώδικα μας σε Javascript (και όχι μόνο). Ενεργοποιούνται αν πιέσετε το F12:



Μπορούμε μάλιστα να εγκαταστήσουμε και μια αρκετά καλύτερη εκδοχή τους, το Firebug:

Ακολουθήστε τις οδηγίες από το <http://getfirebug.com>

## Javascript Crash Course

Μια που ξέρετε PASCAL (έτσι λέω δηλαδή), θα δούμε πολύ γρήγορα τις εντολές της Javascript για να μπορέσουμε να προχωρήσουμε στη συγγραφή ενός απλού παιχνιδιού. Ταυτόχρονα θα δούμε πως μπορούμε να χρησιμοποιήσουμε το περιβάλλον ανάπτυξης (developer tools) για να διορθώσουμε το πρόγραμμα μας.

Φυσιολογικά, η Javascript εκτελείται ως κομμάτι μιας σελίδας και τα αποτελέσματα της τα δείχνει είτε αλλάζοντας το περιεχόμενο και τη μορφή κάποιας υπάρχουσας ετικέτας HTML είτε “ζωγραφίζοντας” μέσα σε ένα καμβά (HTML5 canvas) – κάτι πολύ χρήσιμο για να φτιάξει κανείς παιχνίδια.

Ωστόσο, αν πρόκειται για πειράματα (που θα χρειαστούν για να μάθουμε βασικές εντολές της γλώσσας) μπορούμε να χρησιμοποιήσουμε το περιβάλλον του browser για να γράψουμε το πρόγραμμα μας και την κονσόλα από τα εργαλεία ανάπτυξης για να δούμε το αποτέλεσμα.

Παράδειγμα:

Ας πειράξουμε λίγο την σελίδα του eshop.gr. Ανοίξτε την στο Firefox και βρείτε ένα από τα πιο ακριβά laptop που σας αρέσει. Π.χ.



**LENOVO Y50 TOUCH 59-421810 15.6" INTEL CORE I7-4700HQ GTX860M 2GB WIN 8.**

PER.910394       Like  0

Κατηγορία **ΦΟΡΗΤΟΙ ΥΠΟΛΟΓΙΣΤΕΣ | LENOVO** στην κατηγορία **ΦΟΡΗΤΟΙ ΥΠΟΛΟΓΙΣΤΕΣ**  
Υποκατηγορία 15

- Εκμεταλλευτείτε την ειδική προσφορά έκπτωσης 10% στα παντίκια notebook με την αγορά φορητού υπολ
- Με την αγορά φορητού υπολογιστή κερδίστε 10% έκπτωση στις τσάντες μεταφοράς notebook
- Δείτε περισσότερα στην Κεντρική σελίδα Laptop
- Επιλέξτε βάσεις με η χωρίς ψύξη για το φορητό υπολογιστή σας
- Δωρεάν αποστολή σε όλες τις μεγάλες πόλεις της Ελλάδας

**Διαθεσιμότητα: Άμεσα διαθέσιμο για online παραγγελία**

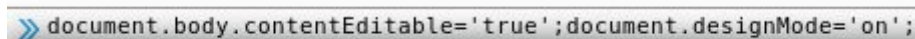
**1599.00**  
**1499.00 €**

 Προσθήκη στο Καλάθι  Προσθήκη στη Wishlist

περιλαμβανομένης ΦΠΑ

Πιέστε το F12 για να ανοίξετε τα developer tools. Πηγαίνετε στην επιλογή console και γράψτε την παρακάτω γραμμή:

```
document.body.contentEditable='true'; document.designMode='on';
```



```
>> document.body.contentEditable='true';document.designMode='on';
```

Κάντε κλικ στην τιμή του laptop και αλλάξτε την από 1499 ευρώ σε 99 ευρώ.

Εκτελέστε την εντολή:

```
document.body.contentEditable='false'; document.designMode='off';
```

Θαυμάστε το αποτέλεσμα:



## LENOVO Y50 TOUCH 59-421810 15.6" INTEL C GTX860M 2GB WIN 8.

PER.910394



e-mail

8+1 0

Like

0



Κατηγορία ΦΟΡΗΤΟΙ ΥΠΟΛΟΓΙΣΤΕΣ | LENOVO στην κατηγορία  
Υποκατηγορία 15

- Εκμεταλλευτείτε την ειδική προσφορά έκπτωσης 10% στα ποντίκια notebook μ
- Με την αγορά φορητού υπολογιστή κερδίστε 10% έκπτωση στις τσάντες μεταφ
- Δείτε περισσότερα στην Κεντρική σελίδα Laptop
- Επιλέξτε βάσεις με η χωρίς ψύξη για το φορητό υπολογιστή σας
- Δωρεάν αποστολή σε όλες τις μεγάλες πόλεις της Ελλάδας

**Διαθεσιμότητα: Άμεσα διαθέσιμο για online παραγγελία**

**99 €**

περιλαμβάνει ΦΠΑ



Προσθήκη στο  
**Καλάθι**



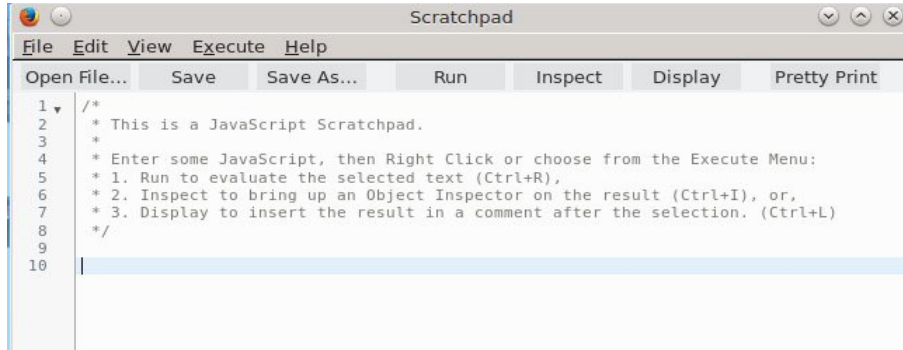
Προσθήκη στη  
**Wishlist**

Παραγγείλτε το laptop και ψάξτε να βρείτε τα υπόλοιπα 1400 ευρώ. 99 ευρώ κάνει σε αυτή τη σελίδα και στη φαντασία μας μόνο!

Στο console μπορούμε να εκτελέσουμε εντολές αλλά και να γράψουμε επίσης χρησιμοποιώντας την εντολή `console.log()`. Θα γράψουμε τώρα μερικά μικρά προγράμματα για να γνωρίσουμε τις βασικές εντολές της Javascript.

## Scratchpad Editor

Στο περιβάλλον του Firefox, και με το console ανοικτό (αν δεν είναι πιάστε το F12), πιάστε το Shift+F4. Θα εμφανιστεί ο Scratchpad, ένας απλός editor για να κάνουμε άμεσα δοκιμές με την Javascript:



Θα δούμε τώρα τα βασικά της Javascript. Τα αποτελέσματα θα εμφανίζονται στο console ή/και στο Scratchpad.

## Τύποι Μεταβλητών

Η Javascript αναγνωρίζει τα παρακάτω είδη:

Αριθμούς	5, 12, 3.464
Λογικές τιμές	true ή false
Strings (μη γελάτε... πάλι)	"Game programming"

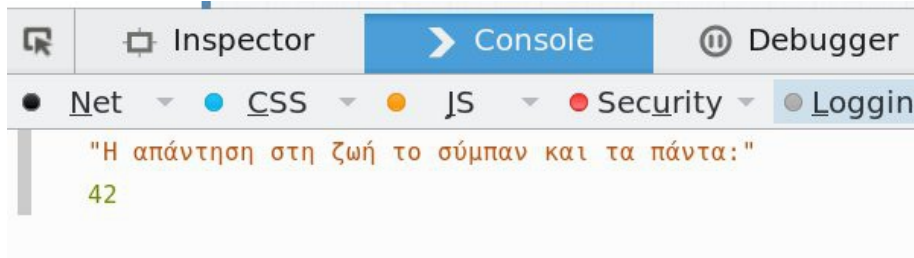
Και την ειδική τιμή null που σημαίνει... τίποτα!

Προσοχή: Η Javascript διακρίνει μεταξύ μικρών και κεφαλαίων. Το Null δεν είναι ίδιο με το null και το NULL. Το σωστό είναι null. Πολύ κακό για το null (τίποτα!)

Για να ορίσουμε μια μεταβλητή χρησιμοποιούμε την **var**:

```
var result=42;  
console.log("Η απάντηση στη ζωή το σύμπαν και τα πάντα:");  
console.log(result);|
```

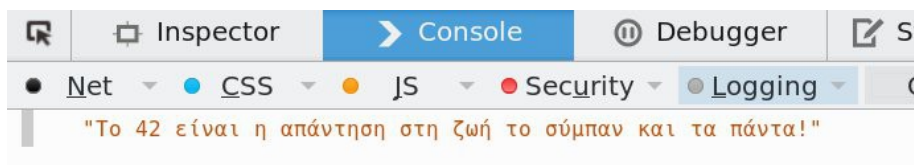
Πιέστε το Run στο Scratchpad και θα δείτε στο console:



Σε αντίθεση με άλλες γλώσσες, δεν δηλώνουμε τα είδη των μεταβλητών όταν τις ορίζουμε. Η Javascript τα καταλαβαίνει μόνη της. Επίσης μπορεί να μετατρέψει από ένα είδος σε ένα άλλο χωρίς να κάνουμε τίποτα:

```
var result=42;  
var answer = "Το "+result+" είναι η απάντηση στη ζωή το σύμπαν και τα πάντα!";  
console.log(answer);
```

Σχεδόν σε οποιαδήποτε άλλη γλώσσα θα είχε το αποτέλεσμα “Προσπαθείς να προσθέσεις ένα αριθμό με ένα string. Μήπως είσαι πωμένος;” (εντάξει, λίγο πιο ευγενικά). Στη Javascript όμως εκτελείται κανονικά, απλώς ο αριθμός μετατρέπεται σε string και ενώνεται με το άλλο:



Τέλος, είναι δυνατόν να αλλάξουμε τύπο σε μια μεταβλητή καθώς το πρόγραμμα εκτελείται:

```
var a=42;  
a="Ε τώρα το θέλω string! :)";
```

Σημειώστε ότι στα strings μπορούμε να χρησιμοποιήσουμε μονά ή διπλά εισαγωγικά (αρκεί να είναι και από τις δύο μεριές τα ίδια!)

Μπορούμε επίσης να χρησιμοποιήσουμε σταθερές για να δώσουμε ονομασία σε τιμές που δεν αλλάζουν:

```
const pi=3.1415;
```

Μπορούμε να ορίσουμε μια μεταβλητή χωρίς ταυτόχρονα να τις δώσουμε τιμή:

```
var x;
```

Στην περίπτωση αυτή έχει την ειδική τιμή “undefined”

## Αποφάσεις και Βρόχοι

Η Javascript έχει τις γνωστές εντολές if – else για κατά συνθήκη εκτέλεση όπως και τις εντολές while και for για βρόχους. Θα τις δούμε με παραδείγματα.

### Εντολή if

Γράψτε το παρακάτω μικρό πρόγραμμα στο Scratchpad. Δείτε τα αποτελέσματα στο console.

```
var temp=prompt("Δωσε θερμοκρασία");
t=parseInt(temp);
if (t>=30) {
    console.log("Κάνει ζέστη!");
} else {
    console.log("Κάνει κρύο...");
}
```

Μπορούμε ακόμα να συνδυάσουμε πολλές if μαζί αν χρειάζεται:

```
var temp=prompt("Δωσε θερμοκρασία");
t=parseInt(temp);
if (t>=30) {
    console.log("Κάνει ζέστη!");
} else if (t>=25) {
    console.log("Καλά είναι");
} else {
    console.log("Μάλλον θα ξεπαγιάσουμε!");
}
```

**Σημείωση:** Με το prompt ζητάμε από το χρήστη να μας δώσει μια θερμοκρασία. Επειδή το αποτέλεσμα αποθηκεύεται στην μεταβλητή temp ως string, το μετατρέπουμε σε αριθμό με την parseInt που φαίνεται στην επόμενη γραμμή και το αποθηκεύουμε στην μεταβλητή t.

Τελεστές σύγκρισης που χρησιμοποιούμε:

Τελεστής Σύγκρισης	Σημασία
>	Μεγαλύτερο
<	Μικρότερο
>=	Μεγαλύτερο ή ίσο

<=	Μικρότερο ή ίσο
==	Ισότητα
===	Ίδια τιμή και ίδιος τύπος μεταβλητής
!=	Ανισότητα

## Βρόχος While

Όπως ήδη ξέρετε από τον προγραμματισμό που κάνετε ήδη, η εντολή `while` επαναλαμβάνει τις εντολές που βρίσκονται μέσα στο σώμα της όσο ισχύει η συνθήκη που έχει. Για παράδειγμα, γράψτε το παρακάτω απλό πρόγραμμα:

```
var n=0;
while (n<=100) {
  console.log(n);
  n=n+1;
}
```

Το πρόγραμμα θα εκτυπώσει τους αριθμούς από 0 ως 100.

**Άσκηση:** Μετατρέψτε το ώστε να υπολογίζει και να εκτυπώνει το άθροισμα των αριθμών από το 0 ως το 100. (Η σωστή απάντηση είναι 5050)

Υπάρχει και η παραλλαγή `do-while` όπου ο έλεγχος γίνεται στο τέλος του βρόχου αντί για την αρχή του. Έτσι είμαστε σίγουροι ότι το σώμα του βρόχου θα εκτελεστεί τουλάχιστον μια φορά!

```
var n=0;
do {
  console.log(n);
  n=n+1;
} while (n<=100);
```

## Βρόχος For

Ο βρόχος `For` επαναλαμβάνεται μέχρι η συνθήκη που έχουμε καθορίσει να γίνει ψευδής. Η γενική σύνταξη του βρόχου φαίνεται στο παράδειγμα παρακάτω (ίδιο αποτέλεσμα με την `while` που δείξαμε πριν):

```
var n;
for (n=0; n<=100; n++) {
  console.log(n);
}
```

Το `n` ξεκινάει από την τιμή 0 και ο βρόχος τερματίζεται όταν περάσει την τιμή 100 (όταν δηλαδή

σταματήσει να ισχύει η συνθήκη  $n \leq 100$ ). Η τρίτη παράμετρος δείχνει πως μεταβάλλεται η τιμή του  $n$  σε κάθε ανακύκλωση: το  $n++$  σημαίνει ότι η τιμή αυξάνεται κάθε φορά κατά 1. Είναι ισοδύναμο με το να γράφαμε:

$$n = n + 1$$

Έχοντας δει τα βασικά στοιχεία της γλώσσας, μπορούμε να δοκιμάσουμε να γράψουμε το πρώτο μας παιχνίδι: Βρες τον αριθμό!

## Το 1ο παιχνίδι: Βρες τον Αριθμό!

Σκοπός του παιχνιδιού είναι να σκεφτεί ο υπολογιστής ένα αριθμό και εμείς να τον βρούμε. Άρα μπορούμε να περιγράψουμε την διαδικασία ως εξής:

- Ο υπολογιστής σκέφτεται ένα τυχαίο αριθμό από 1 ως 100 και τον αποθηκεύει σε μια μεταβλητή
- Μας ρωτάει να μαντέψουμε και αποθηκεύει ότι του δώσουμε σε μια άλλη μεταβλητή
- Συγκρίνει αυτές τις δύο και μας λέει σε μήνυμα αν αυτό που σκεφτήκαμε είναι μικρότερο ή μεγαλύτερο από αυτό που έχει σκεφτεί αυτός.
- Η διαδικασία των ερωτήσεων συνεχίζεται μέχρι να βρούμε τον αριθμό
- Όταν βρούμε τον αριθμό μας δίνει το μήνυμα “Μπράβο το βρήκες”
- Προαιρετικά μπορεί να μετράει και πόσες προσπάθειες χρειαστήκαμε και στο τέλος να μας κάνει κριτική (του στυλ, η γιαγιά μου παίζει πιο καλά. Ναι αυτή με τα μεγάλα εικονίδια στο control panel).

### Πρόβλημα 1ο – Υπολογιστής και Τυχαίοι αριθμοί

Μπορούμε να κάνουμε ένα υπολογιστή να παράγει τυχαίους αριθμούς; Ναι. Μπορούμε να κάνουμε κάτι τέτοιο στη Javascript; Και πάλι ναι!

Η Javascript διαθέτει τη συνάρτηση `random` στη βιβλιοθήκη `Math`. Η συνάρτηση παράγει τυχαίους αριθμούς (με άφθονα δεκαδικά ψηφία!) από 0 ως 0.99999999...

Παράδειγμα:

```
var n = Math.random();  
console.log(n);
```

Θα τυπώσει κάτι σαν:

0.7272229133553736

Μπορούμε να το μετατρέψουμε ώστε να παράγει αριθμούς από το 1 ως το 100;

Και πάλι ναι. Αρκεί να κάνουμε ένα κόλπο σαν αυτό:



```
var n = Math.floor((Math.random() * 10) + 1);
```

Πολλαπλασιάζοντας με το 10, θα πάρουμε τιμές από 0 ως 9. Προσθέτοντας 1, θα έχουμε τιμές από 1 ως 10. Χρησιμοποιούμε και τη συνάρτηση floor για να κόψουμε τα δεκαδικά. Τέλειο!

Είμαστε έτοιμοι για το παιχνίδι μας.

## Βρες τον Αριθμό

Αυτή είναι μια πρώτη εκδοχή και ακολουθούν ασκήσεις!

```
/* Βρες τον αριθμό! */  
var n = Math.floor((Math.random()*100)+1);  
do {  
  g = prompt("Μάντεψε τον αριθμό:");  
  guess = parseInt(g);  
  if (guess>n) {  
    console.log("Έδωσες μεγαλύτερο αριθμό!");  
  } else if (guess<n) {  
    console.log("Έδωσε μικρότερο αριθμό!");  
  }  
} while (guess!=n);  
console.log("Μπράβο το βρήκες!");
```

**Σημείωση:** Όποιος πει ότι δεν του αρέσει θα τον βάλω να υπολογίζει κλιμακωτές χρεώσεις στην Pascal.

Ασκήσεις:

1. Προσθέστε του μηνύματα που να κριτικάρουν την απόδοση του παίκτη, μετρώντας τις προσπάθειες που έκανε για να βρει τον αριθμό.
2. Χρησιμοποιήστε την εντολή alert για να εμφανίζονται τα μηνύματα σε παραθυράκι αντί για την κονσόλα
3. Ξαναγράψτε το πρόγραμμα ώστε να χρησιμοποιεί την εντολή while αντί για do-while
4. Φτιάξτε το πρόγραμμα ώστε να δείχνει το περιεχόμενο των μηνυμάτων σε σελίδα αντί για την κονσόλα.



## Μάθημα 2ο – Συναρτήσεις

### Ορισμός Συνάρτησης

Μια συνάρτηση είναι ένα αυτόνομο κομμάτι κώδικα το οποίο εκτελεί μια συγκεκριμένη λειτουργία (κάνει κάποιο υπολογισμό ή εμφανίζει κάτι στην οθόνη κλπ). Σκοπός μας είναι να το γράψουμε μια φορά και να το χρησιμοποιήσουμε αρκετές, μέσα στο ίδιο πρόγραμμα. Αποφεύγουμε έτσι να επαναλαμβάνουμε τα ίδια κομμάτια κώδικα συνέχεια.

Σε μια συνάρτηση:

- Υπάρχει ένα σημείο μέσα στο πρόγραμμα που την ορίζουμε. Το σημείο αυτό ξεκινάει με την εντολή `function`, το όνομα της συνάρτησης, μια παρένθεση με πιθανές παραμέτρους και ακολουθούν οι εντολές που βρίσκονται στο σώμα της συνάρτησης.
- Αν η συνάρτηση επιστρέφει κάποια τιμή, αυτό γίνεται με την εντολή `return`.

Ένα παράδειγμα θα μας βοηθήσει να καταλάβουμε:

```
function square(x){
  var y = x*x;
  return y;
}

var a=10;
b=square(a);
console.log(b);
```

Στο παράδειγμα μας, ορίζουμε την συνάρτηση `square` η οποία παίρνει ως παράμετρο το `x`. Στη συνέχεια το υψώνει στο τετράγωνο ( $y=x*x$ ) και το επιστρέφει.

Το σημείο που χρησιμοποιούμε την συνάρτηση είναι το παρακάτω:

```
b=square(a);
```

Όταν χρησιμοποιούμε μια συνάρτηση λέμε ότι την **καλούμε**. Μια συνάρτηση που επιστρέφει μια τιμή πρέπει να την λάβουμε εμείς σε κάποια μεταβλητή όπως κάναμε παραπάνω. Θα μπορούσαμε επίσης να την τυπώσουμε κατευθείαν:

```
console.log(square(a));
```

Για την ακρίβεια, το παραπάνω πρόγραμμα θα μπορούσε να συντομευθεί ιδιαίτερα:

```
function square(x){
  return x*x;
}

var a=10;
console.log(square(a));
```

## Επικοινωνία με το Τρέχον Έγγραφο

Καθώς φαντάζεστε δεν έχει νόημα να γράφουμε Javascript για να τυπώνουμε μηνύματα μόνο στην κονσόλα. Η Javascript κανονικά χρησιμοποιείται για να δημιουργεί ή να τροποποιεί μια σελίδα HTML. Για να γίνει αυτό, θα πρέπει:

- Να μπορεί να διαβάζει δεδομένα από τη σελίδα. Π.χ. να διαβάσει ένα αριθμό που έχουμε γράψει σε ένα πεδίο κειμένου μιας φόρμας.
- Να μπορεί να εκτελείται αντιδρώντας π.χ. στη πίεση ενός πλήκτρου που έχουμε τοποθετήσει πάνω στη φόρμα. Για παράδειγμα φανταστείτε το παιχνίδι “Μάντεψε τον αριθμό” που γράφουμε. Θα πρέπει να έχει ένα πλήκτρο Play που να το πατάμε και να μας εμφανίζει το μήνυμα.
- Να μπορεί να γράφει κείμενο πάνω στη σελίδα (και αν θέλουμε φυσικά να το μορφοποιεί κλπ)

Σε μια σελίδα HTML μπορούμε να γράψουμε Javascript τόσο στο head όσο και στο body. Για να μπορούμε όμως να εκτελέσουμε κάποιο συγκεκριμένο κομμάτι του προγράμματος όταν ο χρήστης πιάζει κάποιο button πάνω στη φόρμα, θα το γράψουμε σε μορφή συνάρτησης.

Για το παρακάτω παράδειγμα χρειαζόμαστε πλέον ένα έγγραφο HTML. Δεν θα συνεχίσουμε με το Scratchpad. Ανοίξτε το Bluefish και γράψτε το παρακάτω:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Προσθέστε δύο αριθμούς</title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <form style="width: 300px;">
      <fieldset>
        <legend>Πρόσθεση Αριθμών</legend>
        <label>Αριθμός 1:</label><input type="text" id="number1"><br>
        <label>Αριθμός 2:</label><input type="text" id="number2"><br>
        <input type="button" value="Πρόσθεσε" onclick="addNumbers()">
        <p id="result">Αποτέλεσμα:</p>
      </fieldset>
    </form>
    <script type="text/javascript">
      function addNumbers() {
        var number1=document.getElementById("number1").value;
        var number2=document.getElementById("number2").value;
        result = parseInt(number1)+parseInt(number2);
        document.getElementById("result").innerHTML = "Αποτέλεσμα: "+ result;
      }
    </script>
  </body>
</html>
```

Μόλις συνέλθετε από το σοκ του μεγέθους ενός προγράμματος που απλά προσθέτει δύο αριθμούς, δείτε το αποτέλεσμα:

**Πρόσθεση Αριθμών**

Αριθμός 1:

Αριθμός 2:

Αποτέλεσμα: 30

Μη γκρινιάζετε ότι δεν είναι όμορφο. Μπορείτε να το φτιάξετε όσο όμορφο ή άσχημο θέλετε βελτιώνοντας την HTML και προσθέτοντας φυσικά κατάλληλο CSS. Αυτό δεν μας απασχολεί για την ώρα. Ας δούμε λίγο πως λειτουργεί.

Έχουμε δύο πεδία κειμένου με αναγνωριστικά (id) number1 και number2. Όταν πιάζουμε το πλήκτρο πρόσθεσε, καλείται η συνάρτηση addNumbers(). Αυτό γίνεται μέσω της onclick:

```
<input type="button" value="Πρόσθεσε" onclick="addNumbers()">
```

Η addNumbers δεν δέχεται καμιά παράμετρο, έτσι υπάρχουν απλά οι παρενθέσεις... άδειες!

Η ίδια η συνάρτηση πρέπει να διαβάσει τις τιμές από τα πεδία:

```
var number1=document.getElementById("number1").value;
var number2=document.getElementById("number2").value;
```

Για να διαβάσει η Javascript χρησιμοποιεί τα id των πεδίων. Διαδοχικά, πηγαίνει από το έγγραφο (document) στο πεδίο μέσω του id (getElementById(...)) και τελικά στην τιμή (value).

Έπειτα γίνεται η πρόσθεση:

```
var result = number1 + number2;
```

Μόνο που το παραπάνω, με τους αριθμούς που φαίνονται στο παράδειγμα θα έδινε ως αποτέλεσμα το... 1020. Αυτό γιατί τα πεδία είναι κείμενο, και οι μεταβλητές number1 και number2 καταλήγουν να είναι strings. Χρειαζόμαστε την parseInt που είδαμε και σε προηγούμενο παράδειγμα:

```
var result = parseInt(number1) + parseInt(number2);
```

Τέλος, η συνάρτηση γράφει το αποτέλεσμα στην παράγραφο με id result:

```
document.getElementById("result").innerHTML = "Αποτέλεσμα: "+ result;
```

Η ιδέα πίσω από το innerHTML είναι να γράφει η Javascript ανάμεσα στα <p> και </p> tags!

### Ασκήσεις:

- Μετατρέψτε το πρόγραμμα ώστε να πολλαπλασιάζει δύο αριθμούς
- Μετατρέψτε το πρόγραμμα ώστε να λύνει μια πρωτοβάθμια εξίσωση! Θα πρέπει να ζητά τους αριθμούς a και b και να υπολογίζει την τιμή του x (συνάρτηση  $a*x+b=0$ ). Αν δεν λύνεται θα πρέπει να βγάζει αντίστοιχο μήνυμα. Θα χρειαστείτε την if που είδαμε στο

πρώτο μάθημα

- Γράψτε ένα πρόγραμμα που να ζητάει ένα αριθμό και να υπολογίζει και να τυπώνει το άθροισμα από το 0 μέχρι τον αριθμό αυτό. Π.χ. για τον αριθμό 100 θα δείχνει το 5050.
- Γράψτε ένα πρόγραμμα που να υπολογίζει κλιμακωτή χρέωση της ΔΕΗ. Όχι, πλάκα κάνω :)

## Βρες τον Αριθμό, Εκδοχή 2

Τώρα που είδαμε τις συναρτήσεις και πως μπορούμε να επικοινωνήσουμε με το τρέχον έγγραφο, είναι ώρα να ξαναγράψουμε μια καλύτερη εκδοχή του παιχνιδιού μας:

```
<!DOCTYPE HTML>
<head>
<meta charset="utf-8">
<script type="text/javascript">
  var n = Math.floor((Math.random()*100)+1);
</script>
</head>
<body>
<form style="width:300px;">
<fieldset>
<legend>Βρες τον Αριθμό!</legend>
<label>Μαντεψιά:</label><input type="text" id="guess"/><br>
<input type="button" onclick="getGuess()" value="Παίξε"/><br>
<p id="message"></p>
</fieldset>
</form>
<script type="text/javascript">
function getGuess() {
  guess = document.getElementById('guess').value;
  message = document.getElementById('message');
  if (guess>n) {
    message.innerHTML="Έδωσες μεγαλύτερο αριθμό!";
  } else if (guess<n) {
    message.innerHTML="Έδωσες μικρότερο αριθμό!";
  } else if (guess==n) {
    message.innerHTML="Μπράβο το βρήκες!";
  }
}
</script>
</body>
</html>
```

Παρατηρήστε ότι ο βρόχος δεν υπάρχει πλέον καθώς έχει αντικατασταθεί από το πάτημα του πλήκτρου “Παίξε”. Ο τυχαίος αριθμός δημιουργείται στο head και είναι διαθέσιμος σε όλο το πρόγραμμα (ακόμα και στο script που είναι στο body). Όμως για να ξαναπαίξει ο χρήστης θα πρέπει να κάνει refresh τη σελίδα.