



Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μάθημα: Τεχνικές Προγραμματισμού Υπολογιστών. (Εργαστηριακό μάθημα)

Καθηγητής : Πεφάνης Ευάγγελος

4) Εργαστηριακές σημειώσεις στην γλώσσα προγραμματισμού Java Script.

Συναρτήσεις (Λειτουργίες)

Ενας μεγάλος αριθμός από κατασκευές, είτε φυσικές είτε τεχνητές, βασίζεται στην ιδέα της χρησιμοποίησης μικρότερων δομικών μονάδων. Αυτές οι δομικές μονάδες μπορούν να συνδυαστούν μεταξύ τους με πολλούς διαφορετικούς τρόπους, παρέχοντας μια μεγάλη ποικιλία αντικειμένων με περίπλοκες ιδιότητες. Η ιδέα των μικρότερων δομικών μονάδων, απλοποιεί την ανάπτυξη αυτών των αντικειμένων παρέχοντας μικρότερες δομικές λειτουργίες. Επιπρόσθετα, καθίσταται ευκολότερος ο έλεγχος αυτών. Το ίδιο σκεπτικό υλοποιείται και στην **Javascript** χρησιμοποιώντας τις συναρτήσεις.

Οι συναρτήσεις μας επιτρέπουν την ομαδοποίηση κώδικα που επιτελεί μια συγκεκριμένη λειτουργία και, κατά συνέπεια, την ευκολότερη επαναχρησιμοποίησή του. Επιπλέον, επιτρέπουν τον σχεδιασμό του τελικού συστήματος λογισμικού σε ένα υψηλότερο επίπεδο αφαίρεσης, όπου οι λειτουργικότητα παρέχεται μέσω συναρτήσεων.

Συνάρτηση είναι μια ονομαζόμενη ακολουθία δηλώσεων η οποία πραγματοποιεί έναν συγκεκριμένο υπολογισμό. Αποτελούν μια τεχνική αφαίρεσης μέσω της οποίας μπορεί να δοθεί όνομα σε μια σύνθετη λειτουργία και στην συνέχεια να παρέχεται η δυνατότητα να αναφερόμαστε σε αυτή τη σύνθετη λειτουργία ως μονάδα. Αν μια συνάρτηση εφαρμόζεται σε αντικείμενο, ονομάζεται μέθοδος.

Όρισμα είναι μια τιμή η οποία παίρνεται από το πρόγραμμα στην κλήση της συνάρτησης. Μπορεί να χαρακτηριστεί και ως είσοδος της συνάρτησης από το πρόγραμμα από το οποίο καλείται. Μια συνάρτηση μπορεί να επιδέχεται πολλά ορίσματα.

Επιστρεφόμενη τιμή είναι η τιμή η οποία επιστρέφεται από την συνάρτηση στο περιβάλλον που την κάλεσε μέσω μιας δήλωσης `return`. Όταν συναντάται η δήλωση `return`, τερματίζεται η εκτέλεση της συνάρτησης και η ροή εκτέλεσης επιστρέφει στο περιβάλλον από όπου καλέστηκε.

Υπάρχουν δύο τρόποι για τον ορισμό συναρτήσεων σε διαδικασίες 1) με όνομα και 2) ανώνυμη συνάρτηση.

Για να ορίσουμε μια ονομαστική συνάρτηση, χρησιμοποιούμε τη `function` δήλωση ως εξής:

```
function greet(name)
```

```

{
    return "Γειά σου " + name + "!";
}

console.log(greet("Νίκο")); // prints out Γειά σου Νίκο!

```

Σε αυτή τη συνάρτηση, το `name` όρισμα προς την `greet` συνάρτηση χρησιμοποιείται μέσα στη συνάρτηση για να κατασκευάσει μια νέα συμβολοσειρά και να την επιστρέψει χρησιμοποιώντας τη `return` δήλωση.

Για να ορίσουμε μια ανώνυμη συνάρτηση, μπορούμε εναλλακτικά να χρησιμοποιήσουμε την ακόλουθη σύνταξη:

```

var greet = function(name)
{
    return "Hello " + name + "!";
}

console.log(greet("Eric")); // prints out Hello Eric!

```

Παράδειγμα.

```

function myFunction(p1, p2) {
    return p1 * p2; // The function returns the product of p1 and p2
}

```

Σύνταξη συναρτήσεων JavaScript.

Μια συνάρτηση JavaScript ορίζεται με τη λέξη-κλειδί `function`, ακολουθούμενη από ένα όνομα, ακολουθούμενο από παρενθέσεις `()`.

Οι ονομασίες συναρτήσεων μπορούν να περιέχουν γράμματα, ψηφία, υπογράμμιση και δολάριο (οι ίδιοι κανόνες με τις μεταβλητές).

Οι παρενθέσεις μπορούν να περιλαμβάνουν ονόματα παραμέτρων διαχωρισμένα με κόμματα: `(parameter1, parameter2, ...)`

Ο κώδικας που πρέπει να εκτελεστεί από τη συνάρτηση τοποθετείται μέσα σε άγκιστρα (curly brackets): `{}`

```

function name(parameter1, parameter2, parameter3) {
    // code to be executed
}

```

Οι παράμετροι των συναρτήσεων εμφανίζονται μέσα στις παρενθέσεις () στον ορισμό της συνάρτησης.

Τα ορίσματα (arguments) είναι οι τιμές που λαμβάνονται από τη συνάρτηση κατά την κλήση. Μέσα στη συνάρτηση, τα arguments (οι παράμετροι) συμπεριφέρονται ως τοπικές μεταβλητές. Μια συνάρτηση είναι πολύ ίδια με μια διαδικασία ή μια υπορουτίνα (Procedure ή Subroutine) σε άλλες γλώσσες προγραμματισμού.

Ενεργοποίηση συναρτήσεων (Function Invocation).

Ο κώδικας μέσα στη συνάρτηση θα εκτελεστεί όταν “κάτι” ενεργοποιεί (καλεί) τη συνάρτηση:
Όταν συμβαίνει ένα γεγονός (όταν ένας χρήστης κάνει κλικ σε ένα κουμπί)
Όταν ενεργοποιείται (καλείται) από τον κώδικα JavaScript αυτόματα (αυτο-επίκληση)

Επιστροφή Συνάρτησης (Function Return).

Όταν η JavaScript φτάσει σε μια δήλωση return, η συνάρτηση θα σταματήσει την εκτέλεση. Εάν η συνάρτηση ενεργοποιήθηκε από μια δήλωση, η JavaScript θα “επιστρέψει” για να εκτελέσει τον κώδικα μετά την εντολή ενεργοποίησης. Οι συναρτήσεις συχνά υπολογίζουν μια τιμή επιστροφής. Η τιμή επιστροφής είναι αυτή που επιστρέφει στον “καλούντα”:

Παράδειγμα

Υπολογίζει το γινόμενο δύο αριθμών και επιστρέφει το αποτέλεσμα:

```
var x = myFunction(4, 3); // Function is called, return value will end up in x
function myFunction(a, b) {
  return a * b;           // Function returns the product of a and b
}
```

Το αποτέλεσμα στο x θα είναι:

12

Γιατί χρησιμοποιούμε συναρτήσεις;

Μπορείτε να επαναχρησιμοποιήσετε τον κώδικα: Καθορίστε τον κώδικα μία φορά και χρησιμοποιήστε τον πολλές φορές.

Μπορείτε να χρησιμοποιήσετε τον ίδιο κώδικα πολλές φορές με διαφορετικά ορίσματα, για να παράγετε διαφορετικά αποτελέσματα.

Παράδειγμα

Μετατροπή Φαρενάιτ σε Κελσίου:

```
function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
}
```

```
Console.log( toCelsius(77));
```

Ο τελεστής () καλεί τη συνάρτηση

Χρησιμοποιώντας το παραπάνω παράδειγμα, το toCelsius αναφέρεται στο αντικείμενο της function και toCelsius() αναφέρεται στο αποτέλεσμα της function.

Η πρόσβαση σε μια συνάρτηση χωρίς () θα επιστρέψει τον ορισμό της function, αντί του αποτελέσματος της συνάρτησης:

Παράδειγμα

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}
```

```
Console.log(toCelsius);
```

Συναρτήσεις που χρησιμοποιούνται ως μεταβλητές τιμές.

Οι συναρτήσεις μπορούν να χρησιμοποιηθούν με τον ίδιο τρόπο που χρησιμοποιούνται οι μεταβλητές, σε όλα τα είδη των τύπων, εκχωρήσεων και υπολογισμών.

Παράδειγμα

Αντί να χρησιμοποιήσετε μια μεταβλητή για να αποθηκεύσετε την τιμή επιστροφής μιας συνάρτησης:

```
var x = toCelsius(77);  
var text = "The temperature is " + x + " Celsius";
```

Μπορείτε να χρησιμοποιήσετε τη function απευθείας, ως μεταβλητή τιμή:

```
var text = "The temperature is " + toCelsius(77) + " Celsius";
```

Τοπικές μεταβλητές

Οι μεταβλητές που δηλώνονται μέσα σε μια function JavaScript, γίνονται ΤΟΠΙΚΕΣ (LOCAL) για τη function.

Οι τοπικές μεταβλητές μπορούν να προσπελαστούν μόνο μέσα από τη function.

Παράδειγμα

```
// code here can NOT use carName  
function myFunction() {  
    var carName = "Volvo";  
    // code here CAN use carName  
}  
// code here can NOT use carName
```

Δεδομένου ότι οι τοπικές μεταβλητές αναγνωρίζονται μόνο μέσα στις συναρτήσεις τους, μεταβλητές με το ίδιο όνομα μπορούν να χρησιμοποιηθούν σε διαφορετικές συναρτήσεις.

Οι τοπικές μεταβλητές δημιουργούνται όταν ξεκινά μια function και διαγράφονται όταν ολοκληρωθεί η function.