



Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μάθημα: Τεχνικές Προγραμματισμού Υπολογιστών. (Εργαστηριακό μάθημα)

Καθηγητής : Πεφάνης Ευάγγελος

2) Εργαστηριακές σημειώσεις στην γλώσσα προγραμματισμού JavaScript.

Συνθήκες

Η **if** δήλωση

Η **if** δήλωση μας επιτρέπει να ελέγξουμε εάν μια έκφραση ισούται με **true** ή **false** και να εκτελέσει διαφορετικό κώδικα ανάλογα με το αποτέλεσμα.

Για παράδειγμα, εάν θέλουμε να ελέγξουμε αν μια μεταβλητή *x* είναι μεγαλύτερη από την τιμή 100 χρησιμοποιούμε την **if** ως εξής:

```
if (x>100)
{
    console.log("value of x is biggest than 100");
} else {
    console.log("value of x is smaller than 100");
}
```

Είναι επίσης δυνατό να παραλείψουμε τη λέξη κλειδί **else** εάν θέλουμε να εκτελέσουμε μόνο ένα μπλοκ κώδικα μόνο αν μια συγκεκριμένη έκφραση είναι αληθής.

Για να αξιολογηθεί εάν δύο μεταβλητές είναι ίσες, χρησιμοποιείται ο τελεστής ισότητας **==**. Υπάρχει επίσης ένας άλλος τελεστής ισότητας στο JavaScript, **===** ο οποίος κάνει μια αυστηρή σύγκριση. Αυτό σημαίνει ότι αυτό θα ισχύει μόνο αν τα δύο πράγματα που συγκρίνετε είναι του ίδιου τύπου και του ίδιου περιεχομένου.

```
console.log("1" == 1); // true
console.log("1" === 1); // false
```

Για παράδειγμα:

```
var myNumber = 42;
if (myNumber == 42)
{
    console.log("Ο αριθμός είναι σωστός.");
}
```

```
}
```

Οι σχεσιακοί τελεστές μπορούν επίσης να χρησιμοποιηθούν για την αξιολόγηση των εκφράσεων. Για παράδειγμα:

```
var Age_Giorgos = 20;
var Age_Nikos = 22;

if (Age_Giorgos < Age_Nikos)
{
    console.log("Ο Γιώργος είναι μικρότερος από τον Νίκο.");
}
```

Δύο ή περισσότερες εκφράσεις μπορούν να συγκριθούν μαζί χρησιμοποιώντας λογικούς τελεστές για να ελέγξουν αν δύο εκφράσεις είναι αληθής `true` ισχύουν δηλαδή και οι δύο μαζί ή τουλάχιστον μία από αυτές.

Για να ελέγξετε αν δύο εκφράσεις είναι αληθής (`true`), χρησιμοποιήστε τον τελεστή AND `&&`.

Για να ελέγξετε αν τουλάχιστον μία από τις εκφράσεις είναι αληθής(`true`), χρησιμοποιήστε τον τελεστή OR `||`.

```
var foo = 1;
var bar = 2;
var moo = 3;

if (foo < bar && moo > bar)
{
    console.log("foo is smaller than bar AND moo is larger than bar.");
}

if (foo < bar || moo > bar)
{
    console.log("foo is smaller than bar OR moo is larger than bar.");
}
```

Ο χειριστής NOT `!` μπορεί επίσης να χρησιμοποιηθεί ομοίως:

```
var notTrue = false;
if (!notTrue)
{
    console.log("not not true is true!");
}
```

Η switch δήλωση

Η `switch` δήλωση είναι παρόμοια με τη `switch` δήλωση από τη γλώσσα προγραμματισμού C, αλλά υποστηρίζει και αλφαριθμητικά.

Η `switch` δήλωση χρησιμοποιείται για την επιλογή μεταξύ περισσότερων από δύο διαφορετικών επιλογών και για την εκτέλεση του ίδιου κώδικα για περισσότερες από μία επιλογές. Για παράδειγμα:

```
var rank = "Διοικητής";
switch(rank)
{
    case "Επισκέπτης":
    case "Υπάλληλος":
        console.log("Δεν έχετε εξουσιοδότηση.");
        break;
    case "Διοικητής":
        console.log("Καλημέρα Διοικητά. Τι μπορώ να κάνω για εσάς σήμερα");
        break;
    case "Προιστάμενος":
        console.log("Καλημέρα σας. Πείτε μου τι θέλετε να κάνω.");
        break;
    default:
        console.log("Δεν σας αναγνωρίζω.");
        break;
}
```

Σε αυτό το παράδειγμα, ο " **Επισκέπτης** " και ο " **Υπάλληλος** " ενεργοποιούν την πρώτη πρόταση, ο " **Διοικητής** " ενεργοποιεί τη δεύτερη πρόταση και ο " **Προιστάμενος** " ενεργοποιεί την τρίτη. Για οποιαδήποτε άλλη τιμή της μεταβλητής **rank**, η **default** λέξη-κλειδί ορίζει τη δράση για αυτήν την περίπτωση (προαιρετικά). Πρέπει να χρησιμοποιήσουμε τη **break** δήλωση μεταξύ κάθε μπλοκ κώδικα για να αποφύγουμε την **switch** εκτέλεση του επόμενου μπλοκ κώδικα.

Η χρήση της **switch** δήλωσης γενικά δεν συνιστάται, επειδή εάν ξεχάσουμε την λέξη κλειδί **break** προκαλούνται πολύ συγκεχυμένα αποτελέσματα.