



JavaScript

ΜΑΘΗΜΑ 4.2

ΔΟΜΕΣ ΕΠΙΛΟΓΗΣ

ΠΕΡΙΕΧΟΜΕΝΑ:

1. Η δομή ελέγχου if-else
 1. Ο τριαδικός τελεστής ?:
 2. Ο τελεστής ??
2. Η δομή switch
 1. break στη switch
3. Εμβέλεια Μεταβλητών

Νικόλαος Β.

Σμαραγδένιος Χορηγός Μαθήματος

- Με τη δομή ελέγχου if μπορούμε να εκτελούμε διαφορετικές εντολές ανάλογα με το αν ισχύει μια συνθήκη.
 - Η συνθήκη συντάσσεται συνήθως με χρήση των λογικών και των σχεσιακών τελεστών που είδαμε στο προηγούμενο μάθημα.
 - Έτσι, με την if μπορούμε να κάνουμε διαφορετικές ενέργειες ανάλογα με το αν ικανοποιούνται συνθήκες που εμείς ορίζουμε.

• Απλή if:

```
if (condition) statement;
```

- Ελέγχεται η συνθήκη και
 - αν είναι true (ή truthy) εκτελείται η εντολή.
 - αν είναι false (ή falsy) δεν εκτελείται η εντολή

• if...else:

```
if (condition) statement1;
else statement2;
```

- Ελέγχεται η συνθήκη και
 - αν είναι true/γ εκτελείται η εντολή1.
 - αν είναι false/γ εκτελείται η εντολή2

• if...else if...else:

```
if (condition1) statement1;
else if (condition2) statement2;
else if (condition3) statement3;
...
else statementN;
```

- Ελέγχεται η συνθήκη1, αν είναι true/γ, τρέχει η εντολή1
- Αλλιώς ελ/ται η συνθήκη2, αν είναι true/γ, τρέχει η εντολή2
- ...
- Αν αποτύχουν όλες οι συνθήκες τρέχει η εντολήN

Προσοχή! όπου statement έχουμε:

- είτε ακριβώς μία εντολή
- είτε πολλές εντολές, οι οποίες εντίθενται σε άγκιστρα.

Παράδειγμα 1: comparison

```
let age=44;

if (age<18) {
  console.log("Price: 10E");
  console.log("... and a candy for free");
}
else if (age<65)
  console.log("Price: 20E");
else
  console.log("Price: 15E");
```

Άσκηση 1: Δύο παίκτες ρίχνουν ένα ζάρι. Κατασκευάστε ένα πρόγραμμα το οποίο να εκτυπώνει “Νίκη του A”, “Νίκη του B” ή “Ισοπαλία” ανάλογα με τις τιμές που έχει το ζάρι τους.

ΜΑΘΗΜΑ 4.2: Δομές Επιλογής

1.1. Ο τριαδικός τελεστής ?:

- Ο τριαδικός τελεστής ?: (ternary operator)

- Είναι μια συντομογραφία της if...else, ειδικά για την περίπτωση που θέλουμε να αποθηκεύουμε μία τιμή σε μία μεταβλητή ανάλογα με το αν ισχύει μία συνθήκη.
- Συντακτικό:

```
variable = condition? YesValue: NoValue;
```

- Η συνθήκη:
 - Αν είναι true/γ, τότε ανατίθεται στη μεταβλητή η YesValue,
 - αλλιώς της ανατίθεται η NoValue
- Είναι μια συντομογραφία του ακόλουθου κώδικα:

```
if (condition)
    variable = YesValue;
else
    variable = NoValue;
```

Παρατήρηση:

- Η αποθήκευση της επιστρεφόμενης τιμής δεν είναι υποχρεωτική.
- Πρακτικά όλη η έκφραση υπολογίζεται στην επιστρεφόμενη τιμή και μπορούμε να την χρησιμοποιήσουμε σε μία πιο σύνθετη παράσταση, π.χ.:

```
console.log(condition? YesValue: NoValue);
```

Παράδειγμα 2: ternary

```
let j = 5;
let res = j%2 == 0? j/2: 3*j+1;
console.log(res);
```

Άσκηση 2:

- Ορίστε μία ακέραια μεταβλητή και αναθέστε της μία τιμή.
- Έπειτα με τον τριαδικό τελεστή τυπώστε την απόλυτή τιμή της.

Άσκηση 3:

- Γράψτε έναν ισοδύναμο κώδικα με τον ακόλουθο, χρησιμοποιώντας τον τριαδικό τελεστή και όχι την if..else

```
let x = -4;
let sign;

if (x<0)
    sign=-1;
else if (x===0)
    sign=0;
else
    sign=1;

console.log(sign);
```

Ο τελεστής ??

- Αναφέρεται και ως «πρώτα ορισμένο» (first defined)
- Συντάσσεται ως:

`value1 ?? value2`

- Αν το value1 δεν είναι null ή undefined, τότε το αποτέλεσμα είναι value1
- Αλλιώς είναι value2

Παρατήρηση:

- Είναι μία συντομογραφία του ακόλουθου κώδικα:

```
if (value1 !== null && value1 !== undefined)
    result = value1;
else
    result = value2;
```

Παράδειγμα 3: first defined

```
let x = null;
let y = undefined;
let z = 1;
let w = "value";
console.log(x ?? y ?? z ?? w);
```

Παρατήρηση:

- Ο τελεστής ?? είναι ελαφρά παραλλαγή του τελεστή || όταν δουλεύουμε με truthy - falsy τιμές.
- Έτσι η παράσταση:

`value1 || value2 || value3 || ...`

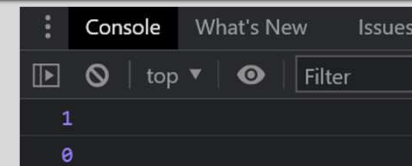
- Θα επιστρέψει την πρώτη τιμή που δεν είναι: false, 0, "", null, undefined
- Ενώ η παράσταση:

`value1 ?? value2 ?? value3 ?? ...`

- Θα επιστρέψει την πρώτη τιμή που δεν είναι: null, undefined

Παράδειγμα 4: first defined2

```
let x = 0;
let y = 1;
console.log(x || y);
console.log(x ?? y);
```



- Η **δομή switch** χρησιμοποιείται όταν θέλουμε να κάνουμε διαφορετικές ενέργειες ανάλογα με την τιμή που έχει μια έκφραση.
 - Ξέρουμε ήδη να το κάνουμε αυτό με διαδοχικές else if, όπου στην κάθε μία ελέγχουμε την τιμή της έκφρασης.
 - Ωστόσο η εντολή switch είναι πιο εύκολα αναγνώσιμη και συνίσταται όταν οι επιλογές που έχουμε είναι περισσότερες από δύο.

- **Συντακτικό switch:**

```
switch(expression) {  
  case expr1:  
    statements1  
    break;  
  case expr2:  
    statements2  
    break;  
  case expr3:  
    statements3  
    break;  
  ...  
  default:  
    statements  
}
```

- **Ανάλογα με την τιμή της έκφρασης expression**
 - Ελέγχεται αν έχει αυστηρά ίση (===) τιμή με την expr1. Αν ναι, εκτελούνται οι εντολές statements1 και έπειτα εξέρχεται από τη switch.
 - ...
 - Αν δεν είναι αυστηρά ίση με κάποια από τις τιμές value1,value2,... τότε τρέχουν οι εντολές που είναι στο default και εξέρχεται της switch.

Παράδειγμα 5: switch.html

```
let value = 3;  
let s = "";  
  
switch(value) {  
  case 1:  
    s = "one";  
    break;  
  case 2:  
    s = "two";  
    break;  
  case 3:  
    s = "three";  
    break;  
  default:  
    s = "other";  
}  
  
console.log(s);
```

Σημείωση:

- Αν και στο συντακτικό, οι expression, expr1, expr2, ... μπορούν να είναι οποιαδήποτε έκφραση, συνηθίζεται στη switch να έχουμε μια μεταβλητή και στις case μία τιμή (literal)

- Αν παραλείψουμε το break σε ένα case:

- και προκύψει η εκτέλεση αυτού του case,
- τότε η εκτέλεση θα συνεχίσει στο επόμενο case (μέχρι να συναντήσει ένα break, ή βγει από τον κώδικα του switch)

- π.χ. στο ακόλουθο τμήμα κώδικα:

```
switch(expression) {  
  case 1:  
    statements1  
  case 2:  
    statements2  
    break;  
  default:  
    statements  
}
```

- αν expression===1, τότε θα εκτελεστούν οι statements1 και έπειτα οι statements2

Άσκηση 4:

Χρησιμοποιώντας αποκλειστικά τη switch (και όχι την if...else)

- Ορίστε μια ακέραια μεταβλητή
- Αν η τιμή της είναι από το 0...5, να εκτυπώνει το μήνυμα "less or equal than 5"
- Αν η τιμή της είναι από το 6..10, να εκτυπώνει το μήνυμα "greater than 5, less or equal than 10"
- Σε κάθε άλλη περίπτωση να βγάζει το μήνυμα "invalid value"

Άσκηση 5: if...else

Επαναλάβετε την προηγούμενη άσκηση αυτή τη φορά χρησιμοποιώντας τη δομή ελέγχου if...else.

Άσκηση 6: Μορφοποιημένη απεικόνιση ώρας

(Χρησιμοποιήστε όποια δομή συνθήκης κρίνετε σκόπιμη για την ευκολότερη διεκπεραίωση της άσκησης)

- Ορίστε τρεις ακέραιες μεταβλητές με ονόματα: hour, minute, second
- Αναθέστε τιμές στις μεταβλητές
- Αρχικά να ελέγχεται ότι οι τιμές είναι έγκυρες (π.χ. η ώρα να είναι από 0 έως 23)
- Να εκτυπώνεται μορφοποιημένα η ώρα π.χ.:

02:23:58

Εμβέλεια μεταβλητών:

- Με τον όρο εμβέλεια (scope) ενός ονόματος εννοούμε:
 - Την περιοχή του προγράμματος στην οποία έχουμε πρόσβαση σε αυτό το όνομα.

Μπλοκ κώδικα και χρόνος ζωής μεταβλητής (με let/const):

- Κάθε ζεύγος άγκιστρων ορίζει και ένα μπλοκ κώδικα.
- Μία μεταβλητή μπορεί να οριστεί οπουδήποτε σε ένα μπλοκ κώδικα και είναι προσβάσιμη μόνο σε αυτό το μπλοκ (και σε ενδεχόμενα μπλοκ που αυτό περιέχει) από το σημείο της δήλωσης της και μετά (βλ. παράδειγμα scope)

```
let x = 0;
if (true) {
  let y = 1;
  console.log(x);
  console.log(y);
}
console.log(x);
// console.log(y); // error
```

- Λέμε ότι η καθολική μεταβλητή x έχει καθολική εμβέλεια (global scope)
- Λέμε ότι η τοπική μεταβλητή y έχει τοπική εμβέλεια μπλοκ (local scope ή block scope)

Σημείωση:

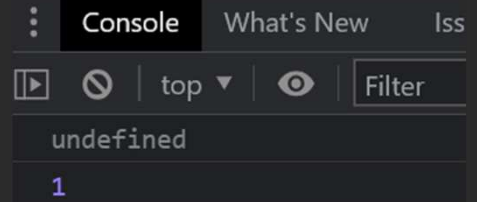
- Σε περίπτωση σύγκρουσης ονομάτων επικρατεί το όνομα που έχει δηλωθεί στο πιο «εσωτερικό» μπλοκ (π.χ. αν η y του παραπάνω παραδείγματος μετονομαστεί σε x, τότε οι εκτυπώσεις θα ήταν 1 1 0)

Ιδιαιτερότητες της JS (< ES6):

- Το strict mode δεν επηρεάζει τη συμπεριφορά της var αναφορικά με την εμβέλεια σε μπλοκ κώδικα. Έτσι:
 - Δεν υπάρχει εμβέλεια μπλοκ.
 - Άρα οι μεταβλητές που δηλώνονται με var σε μπλοκ είναι καθολικές
 - Λόγω του hoisting, είναι σαν να έχουν δηλωθεί στην αρχή του προγράμματος (προσοχή, μόνο δήλωση, όχι αρχικοποίηση)

Παράδειγμα 6: var scope

```
if (true) {
  console.log(x);
  var x = 1;
}
console.log(x);
```



Παράδειγμα 7: var scope2

```
var x = 2;
{
  var x = 1;
}
console.log(x);
```