

Τεχνολογίες Διαδικτύου

Εισαγωγή στη javascript
Εντολές υπό συνθήκη και Βρόχοι επανάληψης,
αντικείμενα

Καθηγητής Χρήστος Δουληγέρης
Δρ. Ρόζα Μαυροπόδη

Υπό συνθήκη εντολές (1/?)

Αποτελούν εργαλείο στη λήψη αποφάσεων και επηρεάζουν έτσι την εξέλιξη της εκτέλεσης του σεναρίου κώδικα της εφαρμογής.

Οι υπό συνθήκη εντολές επιτρέπουν τον έλεγχο ορισμένων συνθηκών και ανάλογα με το αν οι συνθήκες είναι αληθείς ή ψευδείς εκτελούν κάποιες εντολές.

Οι συνθήκες είναι συγκρίσεις μεταβλητών και δεδομένων. Για παράδειγμα:

- Είναι το A μεγαλύτερο από το B;
- Είναι το X ίσο με το Y;
- Είναι το M όχι ίσο με το N;

Στις συγκρίσεις αυτές χρησιμοποιούνται οι τελεστές σύγκρισης και οι λογικοί τελεστές.

Υπό συνθήκη εντολές (2/?)

Η JavaScript υποστηρίζει τις υπο-συνθήκη εντολές: **if**, **if...else** και **switch**.

Οι **if** και **if...else** χρησιμοποιούνται στην περίπτωση που έχουμε να ελέγξουμε ορισμένες συνθήκες και ανάλογα με τα αποτελέσματα να εκτελέσουμε κάποιες εντολές.

```
if (συνθήκη) {  
    μια ακολουθία εντολών αν η σύγκριση είναι  
    αληθής  
}  
if (συνθήκη) {  
    μια ακολουθία εντολών αν η σύγκριση είναι  
    αληθής  
}  
else {  
    ακολουθία εντολών αν η σύγκριση είναι ψευδής  
}
```

Υπό συνθήκη εντολές (3/?)

Στον έλεγχο της 'αλήθειας' της συνθήκης χρησιμοποιούνται οι τελεστές σύγκρισης (==, ===, !=, !==, >, >=, <, <=). Οι πολλαπλές συνθήκες συνδέονται με κάποιον λογικό τελεστή (&&, ||, !) , π.χ.

```
var myAge = 28;
if ((myAge >= 0 && myAge < 3) || myAge > 90) {
    console.log('Μικρό ή μεγάλο παιδί.');
```

Λάθος χρήση του τελεστή ανάθεσης = μέσα σε συνθήκη

```
var myAge = 28;  
if ((myAge = 2) || myAge > 90) {  
    console.log('Μικρό ή μεγάλο παιδί.');}  
console.log (myAge);
```

Συνθήκες πλασματικά αληθείς ή ψευδείς

Truthy: Είναι μια τιμή που θεωρείται 'αληθής' όταν αξιολογείται στο πλαίσιο Boolean, πχ.

```
if (true) { /*εκτελείται πάντα ο κώδικας*/}  
if ({} ) { /*εκτελείται πάντα ο κώδικας*/}  
if ([]) { /*εκτελείται πάντα ο κώδικας*/}  
if (42) { /*εκτελείται πάντα ο κώδικας*/}
```

Falsy: Είναι μια τιμή που θεωρείται 'ψευδής' όταν αξιολογείται στο πλαίσιο Boolean, πχ.

```
if (false) { /*δε θα εκτελεστεί ο κώδικας */}  
if (null) { /*δε θα εκτελεστεί ο κώδικας */}  
if (undefined) { /*δε θα εκτελεστεί ο κώδικας */}  
if (0) { /*δε θα εκτελεστεί ο κώδικας */}  
if (NaN) { /*δε θα εκτελεστεί ο κώδικας */}  
if ('') { /*δε θα εκτελεστεί ο κώδικας */}  
if ("" ) { /*δε θα εκτελεστεί ο κώδικας */}
```

Συνθήκες πλασματικά αληθείς ή ψευδείς: Παράδειγμα

```
//var globalResult;  
function addNumbers(num1, num2) {  
var globalResult;  
    globalResult = num1 + num2;  
    console.log("The global result is: " +  
globalResult);  
}  
addNumbers(5, 7);  
console.log(globalResult);  
if (globalResult) {console.log("Χα δεν θα  
εκτελεστεί");}
```

Σύγκριση αλφαριθμητικών τιμών 1/?

```
var myName = "paul";  
if (myName == "Paul"){  
alert("Η σύγκριση των αλφαριθμητικών κάνει  
διάκριση πεζών ΚΕΦΑΛΑΙΩΝ");  
}
```

```
if ("A" < "B"){  
alert("Σύμφωνα με την αλφαβητική σειρά το A  
είναι πριν το B");  
}
```

```
if ("a" < "B"){  
alert(" Τα πεζά γράμματα είναι μετά τα ΚΕΦΑΛΑΙΑ");  
}
```


Σύγκριση αλφαριθμητικών τιμών 2/?

character	UTF-8	name
	(hex.)	
B	42	LATIN CAPITAL LETTER B
C	43	LATIN CAPITAL LETTER C
D	44	LATIN CAPITAL LETTER D
E	45	LATIN CAPITAL LETTER E
.....
a	61	LATIN SMALL LETTER A
b	62	LATIN SMALL LETTER B
c	63	LATIN SMALL LETTER C
d	64	LATIN SMALL LETTER D
e	65	LATIN SMALL LETTER E
f	66	LATIN SMALL LETTER F
g	67	LATIN SMALL LETTER G
h	68	LATIN SMALL LETTER H

Έλεγχος συνθήκης: αριστερά -> δεξιά

Η javascript ελέγχει μια συνθήκη από αριστερά προς τα δεξιά και σταματά μόλις νομίσει ότι γνωρίζει την απάντηση. Έτσι εφόσον:

`(false && anything) => false`

`(true || anything) => true`

```
var nominator = 5;
```

```
var denominator = 0;
```

```
if (denominator != 0 && (nominator/denominator > 0))  
{  
    console.log('Thats a valid, positive  
fraction');  
}
```

Υπό συνθήκη εντολές if ... else παράδειγμα

Ας υποθέσουμε ότι έχουμε μία φόρμα αίτησης για μία υπηρεσία. Οι χρήστες συμπληρώνουν τα στοιχεία τους και την ηλικία τους και επιλέγουν το πλήκτρο παράδοσης της φόρμας (submit). Θέλουμε οι μικρότεροι σε ηλικία πελάτες (<24) να έχουν έκπτωση 20%.

```
function ShowPrice(age) {  
    if (age<24){  
        alert("Τιμή υπηρεσίας με έκπτωση 20%: 80  
            ευρώ")  
    }  
    else{  
        alert("Τιμή υπηρεσίας: 100 ευρώ ")  
    }  
}
```

```
ShowPrice(30);
```

Υπό συνθήκη εντολές if/else if/else ...παράδειγμα

Ο συνδυασμός if/else μπορεί να χρησιμοποιηθεί όσες φορές απαιτείται

```
function ShowPrice (age) {  
    if (age<24) {  
        alert ("Τιμή υπηρεσίας με έκπτωση 20%: 80  
                ευρώ")  
    }  
    else if (age<30) {  
        alert ("Τιμή υπηρεσίας με έκπτωση 30%:  
                70 ευρώ ")  
    }  
}  
  
ShowPrice (40) ;
```

Υπό συνθήκη εντολές switch

Η switch χρησιμοποιείται στην περίπτωση που ισχύει μία μόνο συνθήκη από ένα σύνολο πολλών συνθηκών.

```
switch (μεταβλητή) {  
  case τιμή1:  
    /*κώδικας */  
    break;  
  case τιμή2:  
    /*κώδικας */  
    break;  
  default:  
    /*κώδικας */  
}
```

switch : μεταβλητή προς έλεγχο
case: έλεγχος συνθήκης
break: τετρατισμός
default: όταν όλα τα υπόλοιπα αποτύχουν

Υπό συνθήκη εντολές switch: παράδειγμα 1/?

Το παράδειγμα αυτό ελέγχει το βαθμό ενός φοιτητή και ανάλογα με την επίδοσή του εμφανίζει σχετικό μήνυμα:

```
var grade = '5';
switch (grade) {
case '6':
result="Moderate";
break;
case '7':
result="Good";
break;
case '8':
result="Very good";
break;
case '9':
result="Excellent!";
break;
default: result="You failed"; }
console.log(result);
```

Υπό συνθήκη εντολές switch: παράδειγμα 2/?

Το παράδειγμα αυτό δεν υπάρχει κώδικας για την περίπτωση όπου ο βαθμό είναι 6. Εφόσον ισχύει η περίπτωση '6' τότε θα εκτελεστεί το τμήμα του κώδικα που περιέχεται στην αμέσως επόμενη περίπτωση:

```
var grade = '6';  
switch (grade) {  
  case '6':  
  case '7':  
    result="Good";  
    break;  
  case '8':  
  case '9':  
    result="Excellent!";  
    break;  
  default: result="You failed"; }  
console.log(result);
```

Υπό συνθήκη εντολές switch: παράδειγμα 3/?

Εφόσον ισχύει η περίπτωση '6' θα εκτελεστεί το τμήμα κώδικά της έως ότου βρει το λεκτικό break.

```
var grade = '6';
switch (grade) {
case '6':
result="Moderate";
case '7':
result="Good";
break;
case '8':
result="Very good";
break;
                case '9':
result="Excellent!";
break;
default: result="You failed"; }
                console.log(result);
```


Βρόχοι επανάληψης

Οι βρόχοι (loops) επιτρέπουν σε μία συνάρτηση να επαναλάβει την ίδια ενέργεια περισσότερες από μία φορές

Για την υλοποίησή τους σε JavaScript μπορούμε να χρησιμοποιήσουμε τις δηλώσεις: **for**, **while** και **do...while**.

Βρόχος for:

```
for ( count=1; count <=5; count++ ) {  
    alert(count); }
```

Η δήλωση διαβάζεται: "Για count ίσο με ένα, όσο το count είναι μικρότερο ή ίσο με πέντε, το count αυξάνει κατά ένα"

Η παραπάνω δήλωση χρησιμοποιεί τη μεταβλητή/μετρητή "count".

1. Αρχικά, το count θα είναι ίσο με 1
2. Ο υπολογιστής θα εκτελέσει τις εντολές που είναι μεταξύ των αγκίστρων, μέχρι να φτάσει στο τέλος του βρόχου όπου το count θα αυξηθεί κατά 1
3. Εάν η νέα τιμή του count είναι μικρότερη ή ίση του 5, ο βρόχος θα επαναληφθεί
4. Διαφορετικά, δηλαδή εάν $count > 5$, τότε το πρόγραμμα θα εξέλθει από το βρόχο και θα συνεχίσει στις υπόλοιπες εντολές

Βρόχος for – Παράδειγμα

Το παρακάτω παράδειγμα επαναλαμβάνει τη λέξη της μεταβλητής txt1 για 5 φορές περιλαμβάνοντας ένα κενό διάστημα ανάμεσα.

```
function repeat(txt1) {  
  var txt2=" ";  
  for (count=0; count<5; count++) {  
    txt2=txt2+txt1+" "; }  
  console.log(txt2);  
}  
repeat ('internet');
```

Βρόχος break

`break`: διακόπτει την επανάληψη

```
function repeat(txt1) {  
  var txt2=" ";  
  for (count=0; count<5; count++) {  
    if (count==3) break;  
    txt2=txt2+txt1+" ";  
  }  
  console.log(txt2);  
}  
repeat ('internet');
```

Βρόχος `continue`

`continue`: παραλείπει τον τρέχοντα κύκλο της επανάληψης

```
function repeat(txt1) {  
  var txt2=" ";  
  for (count=0; count<5; count++) {  
    if (count==3) continue;  
    txt2=txt2+txt1+" ";  
  }  
  console.log(txt2);  
}  
repeat ('internet');
```

Βρόχος while

Ο βρόχος while επαναλαμβάνει ένα τμήμα κώδικα εφόσον ισχύει η συνθήκη. Η γενική μορφή του είναι:

```
while (συνθήκη) {  
    // κώδικας προς επανάληψη  
}
```

Βρόχος while – Παράδειγμα 1/?

Το παρακάτω παράδειγμα επαναλαμβάνει τη λέξη της μεταβλητής txt1 για 5 φορές περιλαμβάνοντας ένα κενό διάστημα ανάμεσα.

```
function repeat(txt1) {  
  var txt2=" ";  
  var count=0;  
  while (count<5) {  
    txt2=txt2+txt1+" ";  
    count++;  
  }  
  console.log(txt2);  
}  
  
repeat ('internet');
```

Βρόχος while – Παράδειγμα 2/?

Ατέρμονη επανάληψη. Παραλήφθηκε η αύξηση του μετρητή.

```
function repeat(txt1) {  
  var txt2=" ";  
  var count=0;  
  while (count<5) {  
    txt2=txt2+txt1+" ";  
    // count++;  
  }  
  console.log(txt2);  
}  
  repeat ('internet');
```

Πίνακες 1/?

Αποθήκευση μιας σειράς από τιμές οποιουδήποτε είδους. Ορισμός:

```
var arrayName = [element0, element1, ...];
```

```
var rainbowColors = ['Red', 'Orange',  
'Yellow', 'Green', 'Blue', 'Indigo',  
'Violet'];
```

```
var raceWinners = [33, 72, 64];  
var myFavoriteThings = ['Broccoli',  
60481, 'Love Actually'];
```

```
console.log(rainbowColors.length);
```


Πίνακες αντιγραφή τιμών σε μεταβλητές

Γενικός τύπος:

```
var arrayItem = arrayName[indexNum];
```

```
var rainbowColors = ['Red', 'Orange',  
'Yellow', 'Green', 'Blue', 'Indigo',  
'Violet'];
```

```
var firstColor = rainbowColors[0];
```

```
var lastColor = rainbowColors[6];
```

Πίνακες αλλαγή τιμών

Αλλαγή συγκεκριμένης τιμής με βάση τη θέση της:

```
var myFavoriteThings = ['Broccoli',  
60481, 'Love Actually'];  
myFavoriteThings[0] = 'Celery Root';
```

Προσθήκη στο τέλος:

```
myFavoriteThings[4] = 'Playgrounds';  
myFavoriteThings[myFavoriteThings.length]  
= 'Playgrounds';
```

Με τη χρήση της push:

```
myFavoriteThings.push('Dancing');
```

Πίνακες πρόσβαση τιμών 1/?

Χρήση του βρόχου for :

```
var rainbowColors = ['Red', 'Orange',  
'Yellow', 'Green', 'Blue', 'Indigo',  
'Violet'];  
  
for (var i = 0; i < rainbowColors.length;  
i++) {  
    console.log(rainbowColors[i]);  
}
```

Πίνακες πρόσβαση τιμών 2/?

Ειδική μορφή του βρόχου for...in :

```
var rainbowColors = ['Red', 'Orange',  
'Yellow', 'Green', 'Blue', 'Indigo',  
'Violet'];  
for (var i in rainbowColors) {  
    console.log(rainbowColors[i]);  
}
```

Αλφαριθμητικά ως πίνακες

```
var rainbowColorsLetters = 'ROYGBIV';  
for (var i in rainbowColorsLetters) {  
    console.log(rainbowColorsLetters[i]);  
}  
console.log(rainbowColorsLetters.length);
```

Αντικείμενα

Τα αντικείμενα αποτελούν έναν τύπο δεδομένου στον οποίο μπορούμε να αποθηκεύσουμε ιδιότητες και μεθόδους.

Οι ιδιότητες δέχονται διακεκριμένες τιμές.

Οι μέθοδοι είναι συναρτήσεις των αντικειμένων και πραγματοποιούν κάποια ενέργεια.

Αντικείμενα ορισμός

```
var objectName = {  
    propertyName: propertyValue,  
    propertyName: propertyValue,  
    ...  
    methodName: function MyMethod () {  
        /*κώδικας μεθόδου*/  
    }  
};
```

Αντικείμενα ορισμός

Οι ιδιότητες χρησιμοποιούνται με τη μορφή όνομαιδιότητας: τιμήιδιότητας. Ένα αντικείμενο μπορεί να περιέχει όσα ζευγάρια όνομαιδιότητας: τιμήιδιότητας απαιτεί, αρκεί να χωρίζονται με κόμμα

```
var aboutMe = {  
    hometown: "Pasadena, CA",  
    hair: "brown"  
};
```


Αντικείμενα ορισμός

Οι τιμές των ιδιοτήτων μπορεί να είναι οποιοσδήποτε τύπος δεδομένων (number, string, array,boolean) ακόμα και άλλο αντικείμενο

```
var lizzieTheCat = {  
  age: 18,  
  furColor: "grey",  
  likes: ["catnip", "milk"],  
  birthday: {"month": 7, "day": 17, year: 1994}  
};
```

Αντικείμενα πρόσβαση στις ιδιότητες και τις τιμές 1/?

Οι πρόσβαση στις τιμές των ιδιοτήτων γίνεται ως εξής:

όνομαΑντικειμένου ■ όνομαιδιότητας

```
var aboutMe = {  
    hometown: "Pasadena, CA",  
    hair: "brown"  
};
```

```
var myHometown = aboutMe.hometown;
```

Αντικείμενα πρόσβαση στις ιδιότητες και τις τιμές 2/?

ή ως εξής:

όνομαΑντικειμένου [όνομαιδιότητας]
(όπως στους πίνακες)

```
var myHair = aboutMe["hair"];
```

Στην περίπτωση όπου η ιδιότητα δεν υπάρχει θα επιστραφεί η τιμή `undefined`

Αντικείμενα πρόσβαση στις ιδιότητες και τις τιμές 3/?

Για τις ιδιότητες μπορεί να οριστούν μεταβλητές οι οποίες λειτουργούν ως 'ψευδώνυμα' των ίδιων των ιδιοτήτων

```
var myProperty = "hair";  
var myHair = aboutMe[myProperty];
```

Αντικείμενα αλλαγή ιδιοτήτων και τις τιμών

Αλλαγή τιμών υπαρχόντων ιδιοτήτων

```
var aboutMe = {  
  hometown: "Pasadena, CA",  
  hair: "brown"  
};
```

```
aboutMe.hair = "blue"; ή  
aboutMe['hair'] = "blue";
```

Προσθήκη νέας ιδιότητας:

```
aboutMe.gender = "female";
```

Διαγραφή ιδιότητας:

```
delete aboutMe.gender;
```

Αντικείμενα σε πίνακες

Αφού οι πίνακες μπορεί να περιέχουν οποιονδήποτε τύπο δεδομένων. Μπορεί να περιέχουν και αντικείμενα.

```
var myCats = [  
  {name: "Lizzie", age: 18},  
  {name: "Daemon", age: 1}  
];
```

```
for (var i = 0; i < myCats.length; i++) {  
  console.log(myCats[i].name + ' είναι' +  
  myCats[i].age + ' χρονών.');
```

Αντικείμενα και μέθοδοι

```
var lizzieTheCat = {  
  age: 18,  
  furColor: 'γκρι',  
  meow: function() {  
    console.log('μιάου μιάου');  
  },  
  eat: function(food) {  
    console.log('Μου αρέσουν ' + food);  
  },  
  sleep: function(numMinutes) {  
    for (var i = 0; i < numMinutes; i++) {  
      console.log('z');  
    }  
  }  
};
```

Αντικείμενα και κλήση μεθόδων

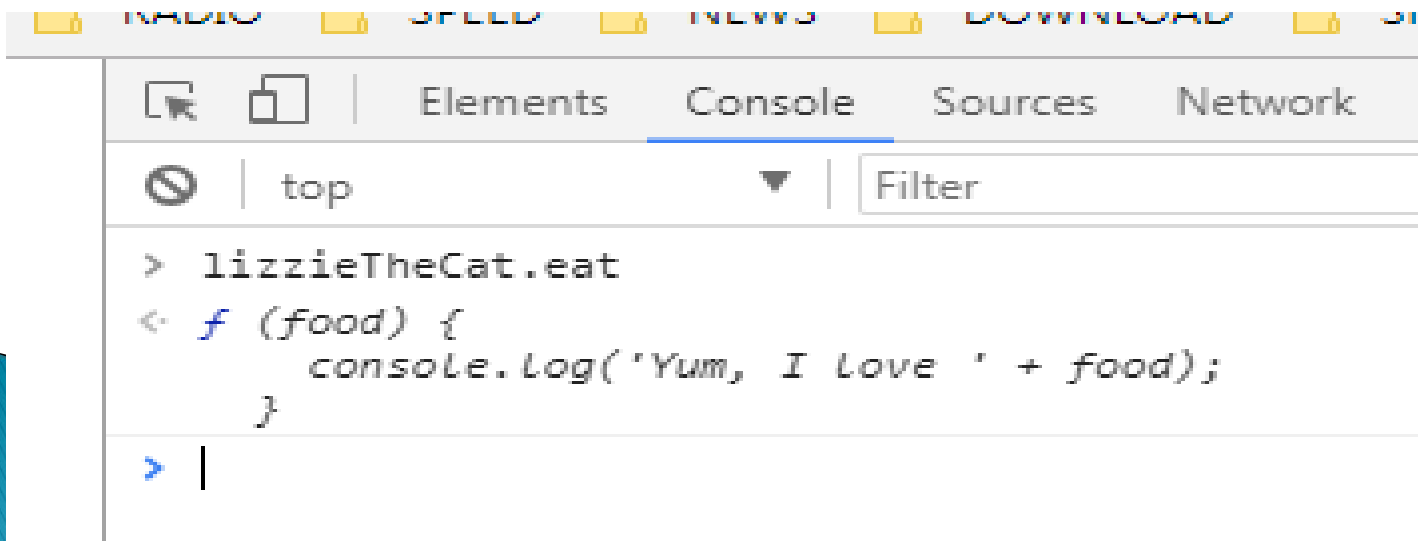
```
lizzieTheCat.meow();
```

```
lizzieTheCat.eat('ψαροκροκέτες');
```

```
lizzieTheCat.sleep(10);
```

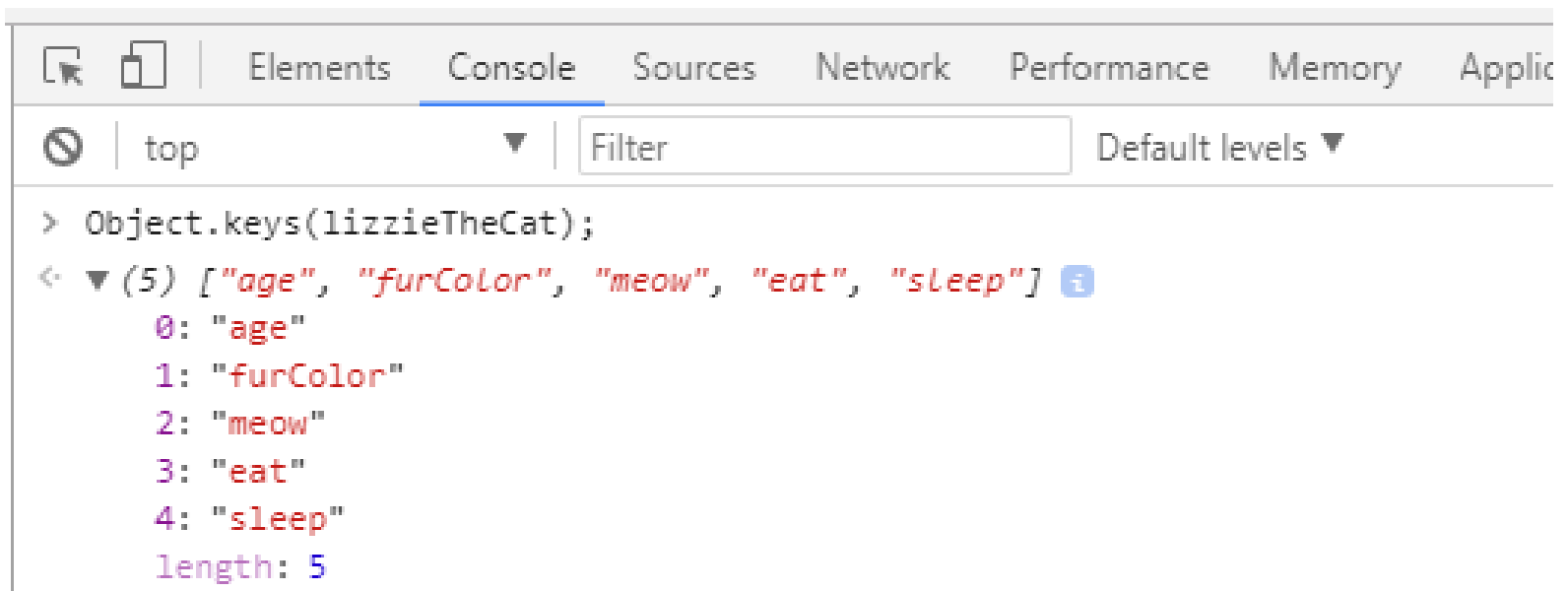
Εφόσον δεν χρησιμοποιήσουμε τις παρενθέσεις θα επιστραφεί ο ορισμός της μεθόδου, π.χ.

```
lizzieTheCat.eat
```



Αντικείμενα η συνάρτηση keys()

```
Object.keys(lizzieTheCat);  
/* ["age", "furColor", "meow", "eat",  
"sleep"] */
```



```
> Object.keys(lizzieTheCat);  
< ▼ (5) ["age", "furColor", "meow", "eat", "sleep"] ⓘ  
  0: "age"  
  1: "furColor"  
  2: "meow"  
  3: "eat"  
  4: "sleep"  
 length: 5
```

Γενικά για τη JavaScript

Η Javascript είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού και έτσι υποστηρίζει μια πληθώρα από έτοιμα αντικείμενα, όπως τα :

Array, Number, Date, Math

και τον ίδιο τον browser.

JavaScript σε σχέση με ιστοσελίδα

Μέχρι τώρα είδαμε την Javascript ως **αυτόνομη** γλώσσα προγραμματισμού.

Στη συνέχεια θα δούμε πώς αλληλοεπιδρά με τον φυλλομετρητή ώστε να **ενσωματωθεί** σε κάποια ιστοσελίδα.

JavaScript το αντικείμενο του browser

Η Javascript ως αντικειμενοστραφής γλώσσα προγραμματισμού αντιμετωπίζει το παράθυρο του φυλλομετρητή, αλλά και όλα όσα βρίσκονται πάνω του, ως αντικείμενα. Έτσι αναγνωρίζει το αντικείμενο

- ▶ window του παραθύρου του φυλλομετρητή.
- ▶ document της ιστοσελίδας HTML.
- ▶ img μιας εικόνας που βρίσκεται στην ιστοσελίδα
- ▶ link ενός συνδέσμου της ιστοσελίδας
- ▶ κλπ.

Παραδείγματα αντικειμένων

Η κλάση του παραθύρου (**window**) έχει:

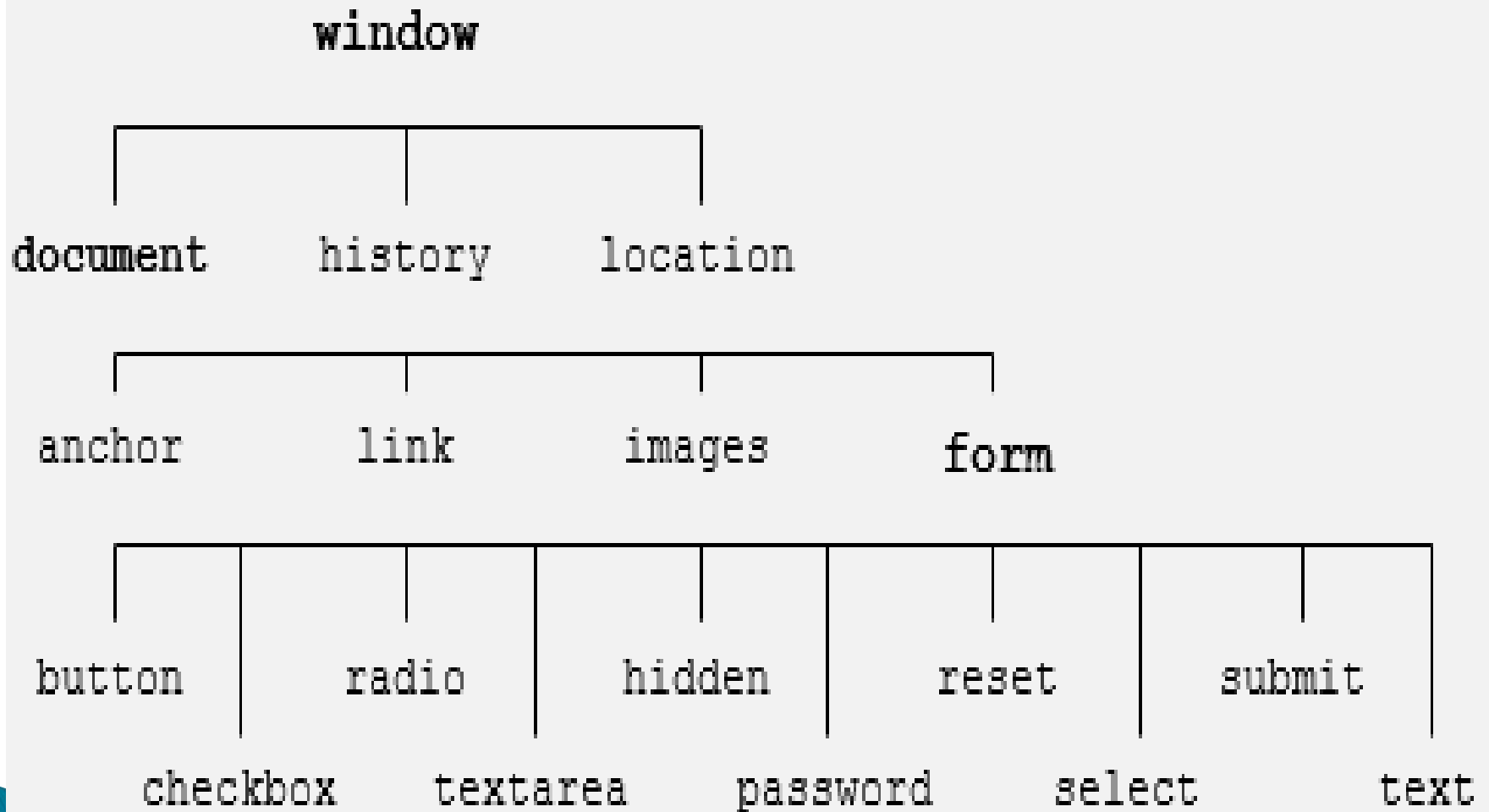
- ▶ ιδιότητες όπως ένα **url**, το όνομα του παραθύρου (**name**), το πλάτος (**width**), το ύψος (**height**) και
- ▶ μεθόδους όπως την **open()** και την **close()**.

Με την εντολή **window.open (“http://www.unipi.gr”)** δημιουργείται και ανοίγει ένα στιγμιότυπο (instance) παραθύρου, που δείχνει στη διεύθυνση <http://www.unipi.gr>

Το αντικείμενο του εγγράφου (**document**) έχει:

- ▶ ιδιότητες όπως τη **bgcolor** που προσδιορίζει το χρώμα του φόντου της σελίδας (π.χ. **document.bgcolor= “red”**) και
- ▶ μεθόδους όπως την **open()** η οποία ανοίγει ένα νέο έγγραφο, τη **write()** που γράφει μέσα στο έγγραφο (π.χ. **document.write(“Web”)**) και την **close()** που κλείνει ένα έγγραφο όταν τελειώσει η επεξεργασία του.

Ιεραρχία αντικειμένων



Προσπέλαση αντικειμένων 1/?

Αρχικά η προσπέλαση των ιδιοτήτων γινόταν με έναν 'σειριακό' τρόπο, με βάση το χαρακτηριστικό name των στοιχείων της HTML, π.χ.

- ▶ `window.document.form1.text1.value`
- ▶ `document.form1.text1.value`
- ▶ `document.form1.images[0].src`
- ▶ `document.form1.images.myimage.width`

Προσπέλαση αντικειμένων 1/?

Παλαιότερος τρόπος προς αποφυγήν.

- ▶ ~~window.document.form1.text1.value~~
- ▶ ~~document.form1.text1.value~~
- ▶ ~~document.form1.images[0].src~~
- ▶ ~~document.form1.images.myimage.width~~

Προσπέλαση αντικειμένων 2/?

Ο τρόπος προσπέλασης των αντικειμένων πραγματοποιείται με:

- ▶ **document.getElementById(id)** Αναφέρεται σε συγκεκριμένο στοιχείο, λόγο του id: `var myElement = document.getElementById("intro");`
- ▶ **document.getElementsByTagName(name).** Αναφέρεται σε μια συλλογή/είδος στοιχείων: `var x = document.getElementsByTagName("p");`

Προσπέλαση αντικειμένων 3/?

- ▶ **document.getElementsByClassName(name).**

Αναφέρεται σε συγκεκριμένη συλλογή στοιχείων με ορισμένες ιδιότητες μορφής: `var x =`

```
document.getElementsByClassName("intro");
```

- ▶ **document.querySelectorAll(css selector)** Αναφέρεται

σε συγκεκριμένη συλλογή στοιχείων με βάση τις

ιδιότητες μορφής τους: `var x =`

```
document.querySelectorAll("p.intro");
```

Προσπέλαση αντικειμένων παράδειγμα 4/?

Προσπέλαση στοιχείων HTML

Προσπέλαση αντικειμένων παράδειγμα 5/?

Η ιδιότητα "id" προσδιορίζει την ταυτότητα ενός στοιχείου σε ένα έγγραφο HTML.

Παραδείγματα:

- **document.getElementById("form1")**

Επιστρέφει ολόκληρο το αντικείμενο της φόρμας με id="form1"

```
var myForm1 =document.getElementById("form1");
```

- **document.getElementById("form1").element[0].value**

Επιστρέφει την τιμή (value) ενός συγκεκριμένου (του πρώτου) στοιχείου της φόρμας.

```
myForm1.element[0].value;
```

```
var myText1 = document.getElementById("form1").element[0];
```

```
myText1.value;
```

- **document.getElementById("text1").value**

Επιστρέφει την τιμή (value) του στοιχείου με id="text1". Το στοιχείο αυτό μπορεί να βρίσκεται και εντός μιας φόρμας.

```
myText1.value;
```

Έξοδος αποτελεσμάτων

Η JavaScript μπορεί να παρουσιάσει πληροφορία με διάφορους τρόπους:

- ✓ Μέσα στην console του φυλλομετρητή, με τη χρήση της μεθόδου **console.log()**.
- ✓ Μέσα σε κάποιο παράθυρο μηνύματος HTML, με τη χρήση των μεθόδων **window.alert()**, **window.prompt()**, **window.confirm()**.
- ✓ Μέσα σε ένα στοιχείο της HTML, με τη χρήση της ιδιότητας **innerHTML**.
- ✓ Μέσα σε ένα έγγραφο HTML, με τη χρήση της μεθόδου **document.write()**.

Έξοδος αποτελεσμάτων παραδείγματα

innerHTML vs innerText

innerHTML: επιτρέπει την εισαγωγή κειμένου με μορφοποίηση, ενώ η

innerText: επιτρέπει την εισαγωγή απλού κειμένου.

Έξοδος αποτελεσμάτων παραδείγματα

part2.html

file:///H:/ROSA120/1Rosa/book_2014/classes/proptuxiako/1ex%20Technologies_web/my_presentation/javascript/part2.html

Γειά σου κόσμε!!!!

Παράδειγμα επιλογής στοιχείου με τη χρήση της μεθόδου **getElementById** !

Κείμενο από την παράγραφο με id=intro: Γειά σου κόσμε!!!!

Η Javascript είναι εύκολη.

Παράδειγμα επιλογής στοιχείου με τη χρήση της μεθόδου **getElementsByName** !

Η πρώτη παράγραφος (index 0) είναι: Γειά σου κόσμε!!!!

Πρώτη παράγραφος η οποία χρησιμοποιεί class="intro".

Παράδειγμα επιλογής στοιχείου με τη χρήση της μεθόδου **getElementsByClassName** !

Η πρώτη παράγραφος (index 0) με class="intro" είναι: Πρώτη παράγραφος η οποία χρησιμοποιεί class="intro".

Πρώτο στοιχείο div το οποίο χρησιμοποιεί class="intro"

Παράδειγμα επιλογής στοιχείου με τη χρήση της μεθόδου **querySelectorAll** !

Η πρώτη παράγραφος (index 0) με class="intro" είναι: Πρώτη

```
<!DOCTYPE html>
<html>
<head>/head>
<body>
  <p id="intro">Γειά σου κόσμε!!!!</p>
  <p id="demo">Κείμενο από την παράγραφο με id=intro: Γειά σου κόσμε!!!!</p> == $0
  <br>
  <br>
  "*****"
  <br>
  <br>
  <p>H Javascript είναι εύκολη.</p>
  <p id="demo2">Η πρώτη παράγραφος (index 0) είναι: Γειά σου κόσμε!!!!</p>
  <br>
  <br>
  "*****"
  <br>
  <br>
  <p class="intro">Πρώτη παράγραφος η οποία χρησιμοποιεί class="intro".</p>
  <p class="intro">...</p>
  <p id="demo3">...</p>
  <br>
  <br>
  "*****"
  <br>
  <br>
  <div class="intro">Πρώτο στοιχείο div το οποίο χρησιμοποιεί class="intro"</div>
  <p class="intro">...</p>
  <p id="demo4">...</p>
  <script>...</script>
</body>
```

Styles

- clientLeft: 0
- clientTop: 0
- clientWidth: 428
- contentEditable: "inherit"
- dataset: DOMStringMap {}
- dir: ""
- draggable: false
- firstChild: text
- firstElementChild: null
- hidden: false
- id: "demo"
- innerHTML: "Κείμενο από την παράγραφο με id=intro: Γειά σου κόσμε!!!!"
- innerText: "Κείμενο από την παράγραφο με id=intro: Γειά σου κόσμε!!!!"
- isConnected: true
- isContentEditable: false
- lang: ""
- lastChild: text
- lastElementChild: null
- localName: "p"
- namespaceURI: "http://www.w3.org/1999/xhtml"
- nextElementSibling: br
- nextSibling: text
- nodeName: "P"
- nodeType: 1
- nodeValue: null
- nonce: ""
- offsetHeight: 18
- offsetLeft: 8
- offsetParent: body
- offsetTop: 102

Έξοδος αποτελεσμάτων παραδείγματα

Η χρήση του `document.write()` θα διαγράψει όλα τα περιεχόμενα του εγγράφου HTML και θα εμφανίσει το αναγραφόμενο κείμενο.

`document.write`