

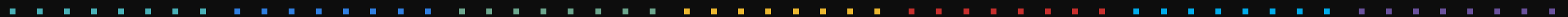
WMNFE2110 | FRONT-END DEVELOPMENT

JavaScript #5

Functions



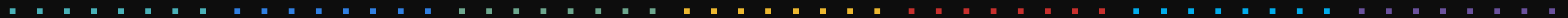
ΠΕΡΙΕΧΟΜΕΝΑ



Περιεχόμενα

- Γενικά
 - Returning / Non-returning
 - Παράμετροι
- Built-in / API
 - Global
 - Window
 - Math
 - String / Array
 - Document
 - Console

FUNCTIONS



Functions

Μια συνάρτηση

- είναι ένα βασικό δομικό στοιχείο ενός προγράμματος.
- είναι ένα σύνολο από εντολές, που μπορούν να κληθούν με την χρήση από διάφορα σημεία του κώδικα
- βοηθά στην οργάνωση του κώδικα.
- μπορεί να εκτελεί συγκεκριμένες πράξεις, ελέγχους ή διαδικασίες.
- μπορεί να δέχεται παραμέτρους και να δίνει ανάλογα αποτελέσματα.

Functions

Σύνταξη

```
function functionName(<params>) {
    // κώδικας της συνάρτησης
}

// ES5 - function expression / anonymous function
const functionName = function() {
    // κώδικας της συνάρτησης
};

// ES6 - arrow function
const functionName = () => {
    // κώδικας της συνάρτησης
};
```

Functions

Συναρτήσεις τύπου **void** / **non-returning**

- Οι συγκεκριμένες συναρτήσεις ορίζονται και καλούνται μέσα στον κώδικα μας. Δεν επιστρέφουν κάποιο αποτέλεσμα πράξεων, απλά εκτελούν μια σειρά από ενέργειες. Σε ορισμένες γλώσσες προγραμματισμού, μια τέτοια συνάρτηση ονομάζεται **διαδικασία**.
- Π.χ. **Window.alert()** κ.α.

Functions

Συναρτήσεις τύπου **returning**

- Εκτελούν συγκεκριμένες πράξεις και επιστρέφουν ένα αποτέλεσμα μέσω της εντολής **return** (συχνά στο τέλος της συνάρτησης).
- Οι συναρτήσεις αυτές είναι συνήθως αυτόνομες (δεν κάνουν κλήση σε κώδικα ή αναφορά σε μεταβλητές έξω από αυτές) και έχουν σκοπό να επιστρέψουν μια τιμή.
- Π.χ. **Window.prompt()**, **Number.isNaN()** κ.α.

Functions

Παράμετροι

- Οι συναρτήσεις, ανεξάρτητα από το είδος τους, μπορούν να πάρουν ως είσοδο μία ή περισσότερες παραμέτρους.
- Οι παράμετροι αυτοί ορίζονται μέσα στις παρενθέσεις που βρίσκονται δίπλα στο όνομα της συνάρτησης.
- Οι παράμετροι που ορίζονται στις παρενθέσεις αντιστοιχούν με τα δεδομένα (μεταβλητές, τιμές) που θα χρησιμοποιήσουμε όταν καλούμε την συνάρτηση.
- Το πλήθος των παραμέτρων μπορεί να είναι, θεωρητικά, άπειρο.
- Καμία παράμετρος δεν είναι υποχρεωτική, όποια δεν πάρει τιμή θα παραμείνει ως **undefined**.
- Η σειρά με την οποία ορίσαμε τις παραμέτρους, είναι και η σειρά που θα περάσουμε τις τιμές.

Functions

Παράδειγμα #1

```
function greet(name) {
  alert('Καλωσήλθες ' + name + '!');
}

greet('John Doe');
greet('Joan Doe');
```

Functions

Παράδειγμα #2

```
function multiply(a, b) {
  return a * b;
}

let x = multiply(4, 3);
let y = multiply(5, 9);
```

Functions

Παράδειγμα #3

```
function greetBob() {
  let person = prompt("Please enter your name:");

  if (person === "Bob") {
    alert("Welcome Bob!");
  } else {
    alert("I don't know you...");
  }
}

greetBob();
```

Functions

Παράδειγμα #4

```
function average(x, y, z) {  
  return (x + y + z) / 3;  
}  
  
console.log(average(5, 7, 4));
```

Functions

Παράδειγμα #5

```
function myMin(x, y) {
  if (x > y) {
    return x;
  } else {
    return y;
  }
}

// Here be dragons
// const myMin = (x, y) => x < y ? x : y;

console.log(myMin(5, 7));
```

Functions

Παράδειγμα #6

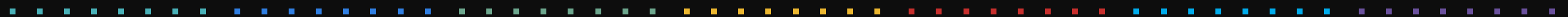
```
function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
}

// Here be dragons
// const toCelsius = f => (5/9) * (f-32);

const input = prompt('Temperature in fahrenheit:');
const f = parseInt(input);

alert(`Temperature in celsius: ${toCelsius(f)}`);
```

GLOBAL FUNCTIONS



Global Functions

Αριθμητικές συναρτήσεις

- **parseInt**
Αναλύει μια συμβολοσειρά και επιστρέφει έναν ακέραιο.
- **parseFloat**
Αναλύει μια συμβολοσειρά και επιστρέφει έναν αριθμό κινητής υποδιαστολής.
- **isNaN**
Ελέγχει εάν μια τιμή είναι μη έγκυρος αριθμός.

Σημ.: **isNaN** \neq **Number.isNaN**

Global Functions

Παράδειγμα #1

```
let a = parseInt('10');
let b = parseInt('10.00');
let c = parseInt('10.33');
let d = parseInt('34 45 66');
let e = parseInt(' 60 ');
let f = parseInt('40 years');
let g = parseInt('He was 40');

console.log(a, b, c, d, e, f, g);
```

Global Functions

Παράδειγμα #2

```
let a = parseFloat('10');
let b = parseFloat('10.00');
let c = parseFloat('10.33');
let d = parseFloat('34 45 66');
let e = parseFloat(' 60 ');
let f = parseFloat('40 years');
let g = parseFloat('He was 40');

console.log(a, b, c, d, e, f, g);

console.log(isNaN(a));
console.log(isNaN(g));
```

WINDOW FUNCTIONS

Window Functions

Συναρτήσεις πλαισίων διαλόγου

- **window.alert**
Εμφανίζει ένα πλαίσιο προειδοποίησης.
- **window.prompt**
Εμφανίζει ένα πλαίσιο διαλόγου, που ζητά από το χρήστη να εισάγει κάποιο κείμενο
- **window.confirm**
Εμφανίζει ένα πλαίσιο διαλόγου, με τις επιλογές 'OK' και 'Cancel'

Window Functions

Παράδειγμα

```
const answer = prompt('What is your age?', 18);
const age = parseInt(answer);

if (age >= 18) {
  if (confirm('Do you promise you are over 18?')) {
    alert('Welcome');
  }
}
```

Window Functions

Συναρτήσεις χρονόμετρου

- **window.setTimeout**
Καλεί μια συνάρτηση (ή εκτελεί μια "παράσταση") **μετά** από έναν καθορισμένο αριθμό milliseconds
- **window.setInterval**
Καλεί μια συνάρτηση (ή εκτελεί μια "παράσταση") **ανά** έναν καθορισμένο αριθμό milliseconds

Window Functions

Παράδειγμα

```
function alarm() {
    alert('Wake up!');
}

// One-shot timer
setTimeout(alarm, 2000);

// Interval timer
let snoozer = setInterval(alarm, 5000);

// Επαναλαμβάνεται μέχρι να καλέσουμε:
clearInterval(snoozer);
```


MATH FUNCTIONS

Math Functions

Μαθηματικές συναρτήσεις

- **Math.abs** - Απόλυτη τιμή
- **Math.pow** - Ύψωση σε δύναμη
- **Math.sqrt** - Τετραγωνική ρίζα
- **Math.min** - Ελάχιστο στοιχείο
- **Math.max** - Μέγιστο στοιχείο

Math Functions

Παράδειγμα #1

```
let radius = prompt('Radius of the circle: ', 0);  
  
let circumference = 2 * Math.PI * radius;  
let area = Math.PI * Math.pow(radius, 2);  
  
console.log(circumference, area);
```

Math Functions

Παράδειγμα #2

```
console.log(Math.abs(-101));    // result is 101

let max = Math.max(21, 22);    // result is 22
let min = Math.min(30.1, 30.2); // result is 30.1

console.log(max, min);

// Something extra
let a = 5.000001, b = 4.999999;
let floatEqual = Math.abs(a - b) < 0.001;

console.log(Math.max(a, b), Math.min(a, b));
```

Math Functions

Μαθηματικές συναρτήσεις

- **Math.random**
Τυχαία τιμή μεταξύ **0** και **1** (χωρίς το **1**)
- **Math.round**
Στρογγυλοποίηση στον κοντινότερο ακέραιο
- **Math.floor**
Στρογγυλοποίηση σε ακέραιο, "προς τα κάτω"
- **Math.ceil**
Στρογγυλοποίηση σε ακέραιο, "προς τα πάνω"

Σημ.: **Number.prototype.toFixed**

Στρογγυλοποίηση σε συγκεκριμένο αριθμό δεκαδικών

Math Functions

Παράδειγμα #1

```
let a = Math.random();  
  
console.log(a);  
console.log(Math.round(a));  
console.log(Math.floor(a));  
console.log(Math.ceil(a));  
  
console.log(a.toFixed(2)); // Number.prototype
```

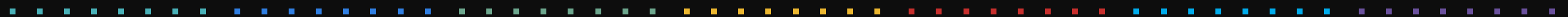
Math Functions

Παράδειγμα #2

```
for (let i = 0; i < 10; i++) {
  let dice_roll = Math.floor(Math.random() * 6) + 1;
  console.log(dice_roll);
}
```

```
// Math.random() ∈ [0, 1)
// Math.random() * 6 ∈ [0, 6)
// Math.random() * 6 + 1 ∈ [1, 7) <=> [1, 6]
```

STRING/ARRAY FUNCTIONS



String/Array Functions

Συναρτήσεις αναζήτησης

- **indexOf**
αναζητά και επιστρέφει τη πρώτη θέση ενός στοιχείου σε ένα κείμενο ή πίνακα
- **lastIndexOf**
αναζητά και επιστρέφει τη τελευταία θέση ενός κομμάτι κειμένου σε ένα κείμενο

String/Array Functions

Συναρτήσεις συνένωσης και διαχωρισμού

- **split**
Διαχωρίζει ένα αλφαριθμητικό σε περισσότερα, βάσει κάποιου διαχωριστή/οριοθέτη, και επιστρέφει τα αποτελέσματα σε πίνακα
- **join**
Συνενώνει τα στοιχεία ενός πίνακα, με τη χρήση κάποιου διαχωριστή/οριοθέτη, και επιστρέφει ένα αλφαριθμητικό

String/Array Functions

Παράδειγμα #1

```
let ipsum = 'December frosty jingle bells gift '  
ipsum     += 'december chestnuts gold december.';  
  
console.log(text.indexOf('December'));  
console.log(text.indexOf('december'));  
  
console.log(text.lastIndexOf('december'));
```

Παράδειγμα #2

String/Array Functions

```
const planets = [
  "Mercury",
  "Venus",
  "Earth",
  "Mars",
  "Jupiter",
  "Saturn",
  "Uranus",
  "Neptune"
];

let place = planets.indexOf('Earth') + 1;
place += place === 1 ? 'st' :
         place === 2 ? 'nd' :
         place === 3 ? 'rd' : 'th';

console.log(`Earth is in ${place} place.`);
```

String/Array Functions

Παράδειγμα #3

```
const input = prompt('Enter firstname & lastname:');  
const details = input.split(',');  
  
console.log(`Firstname: ${details[0]}`);  
console.log(`Lastname: ${details[1]}`);
```

String/Array Functions

Παράδειγμα #3

```
const grades = [18, 17, 19, 18, 20];

console.log(`Your grades are ${grades}.`);
console.log(`Your grades are ${grades.join(', ')}.`);
console.log(`Your grades are ${grades.join('-')}.`);
```

DOCUMENT FUNCTIONS

Document Functions

DOM συναρτήσεις

- **document.write**
«γράφει» ένα κομμάτι κειμένου σε μία ροή κειμένου (**stream**)
- **document.getElementById**
επιστρέφει ένα αντικείμενο βάσει ενός αναγνωριστικού (**id**)
- **document.querySelector**
επιστρέφει ένα αντικείμενο βάσει ενός επιλογέα (**selector**)
- **document.getElementsByName**
επιστρέφει μία συλλογή αντικειμένων βάσει του ονόματός τους
- **document.getElementsByTagName**
επιστρέφει μία συλλογή αντικειμένων για τη δεδομένη ετικέτα (**tag**)
- **document.getElementsByClassName**
επιστρέφει μία συλλογή αντικειμένων για τη δεδομένη κλάση (**class**)
- **document.querySelectorAll**
επιστρέφει μία λίστα αντικειμένων βάσει ενός επιλογέα (**selector**)

Document Functions

Παράδειγμα #1

```
document.write("<h1 id='header'>Hello!</h1>");
document.write("<p>It's me.</p>");

const el = document.getElementById('header');

el.textContent = 'Hello you!';
el.style.color = '#b00';
```

Document Functions

Παράδειγμα #2

```
const elements = document.getElementsByTagName('a');

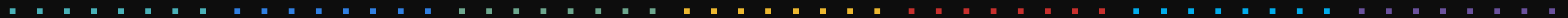
for (let el of elements) {
  el.target = `_blank`;
  console.log(el.text);
}
```

Document Functions

Παράδειγμα #3

```
const el = document.querySelector('input[type="submit"]');  
  
el.disabled = 'disabled';  
el.classList.add('gray');
```

CONSOLE FUNCTIONS

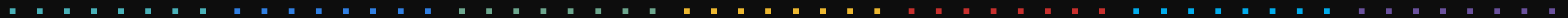


Console Functions

Συναρτήσεις εκσφαλμάτωσης

- **console.log**
Εμφανίζει ένα ή περισσότερα στοιχεία στην «κονσόλα» του browser
- **console.warn**
Εμφανίζει ένα ή περισσότερα στοιχεία στην «κονσόλα» του browser, ως προειδοποίηση
- **console.error**
Εμφανίζει ένα ή περισσότερα στοιχεία στην «κονσόλα» του browser, ως σφάλμα
- **console.time** & **console.timeEnd**
Ξεκινά και σταματά, αντίστοιχα, ένα χρονόμετρο και στο τέλος επιστρέφει το χρόνο που διανύθηκε σε ms

CLASSWORK



Classwork #1

Carousel

Φτιάξτε μία **HTML** σελίδα που θα εμφανίζει μία εικόνα η οποία θα αλλάζει αυτόματα κάθε μερικά δευτερόλεπτα.

Μπορείτε να βρείτε τυχαίες εικόνες από εδώ: <https://picsum.photos/>

Classwork #2

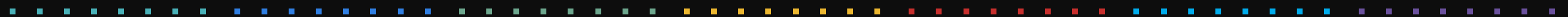
Ζάρια

Φτιάξτε μία **HTML** σελίδα που θα προσομοιώνει τη ρίψη δύο ζαριών, κάθε φορά που φορτώνει/ανανεώνεται η σελίδα.

Μπορείτε να βρείτε εικόνες για τα ζάρια από εδώ:

https://commons.wikimedia.org/wiki/Category:Dice_faces

HOMework



Homework

Homework #1

Γράψτε μία συνάρτηση με το όνομα **longestWord** η οποία να δέχεται έναν πίνακα με αλφαριθμητικά και να επιστρέφει το μεγαλύτερο από αυτά.

Για να την ελέγξετε, μπορείτε να χρησιμοποιήσετε τον κώδικα του *παραδείγματος 4* και συγκεκριμένα τα κομμάτι εκείνο που ζητά επαναληπτικά τιμές από το χρήστη και τις αποθηκεύει σε πίνακα.

Homework

Homework #2 (1/2)

Φτιάξτε ένα **script** που να δέχεται ένα λεκτικό από το χρήστη και θα ελέγχει αν το λεκτικό αυτό περιέχει μη επιτρεπόμενους χαρακτήρες. Ο έλεγχος αυτός πρέπει να γίνεται με τη βοήθεια μιας συνάρτησης, όπως αυτή φαίνεται παρακάτω:

```
// Should return true/false
function containsNotAllowedChars(input) {
  const notAllowedChars = ['<', '>', '@', '#'];
  ...
}
```

Η συνάρτηση μπορεί να παίρνει έναν-έναν τους χαρακτήρες του λεκτικού και να ελέγχει αν υπάρχουν στον πίνακα ή το ανάποδο. Αν βρεθεί μη επιτρεπτός χαρακτήρας θα πρέπει να επιστρέφεται η τιμή **true**, αλλιώς η τιμή **false**.

Η όλη διαδικασία επαναλαμβάνεται μέχρι να δοθεί έγκυρο λεκτικό από το χρήστη.

Homework

Homework #2 (2/2)

Για να ελέγξετε τη συνάρτηση που φτιάξατε μπορείτε να εκτελέσετε το παρακάτω:

```
if (!containsNotAllowedChars('This is a test') &&
    containsNotAllowedChars('This is a <test>') &&
    containsNotAllowedChars('This is @ test') &&
    containsNotAllowedChars('T#is is a test'))
{
    alert('All good!');
} else {
    alert('Something is not right');
}
```

Χρήσιμα links (1/2)

 Functions


<https://javascript.info/function-basics>

 JavaScript Functions

https://www.w3schools.com/js/js_functions.asp

 Function expressions

<https://javascript.info/function-expressions>

 JavaScript Anonymous Functions

<https://www.javascripttutorial.net/javascript-anonym...>

 ES5 vs. ES6: Functions. ES5 vs. E6: Functions

https://medium.com/@zac_heisey/es5-vs-es6-functio...

 String - JavaScript | MDN

<https://developer.mozilla.org/en-US/docs/Web/JavaS...>

 Array - JavaScript | MDN


<https://developer.mozilla.org/en-US/docs/Web/JavaS...>

Χρήσιμα links (2/2)

 Standard built-in objects - JavaScript
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects

 JavaScript Global Reference
https://www.w3schools.com/jsref/jsref_obj_global.asp

 Numbers
<https://javascript.info/number>

 Interaction: alert, prompt, confirm
<https://javascript.info/alert-prompt-confirm>

 JavaScript Popup Boxes
https://www.w3schools.com/js/js_popup.asp

 Searching: getElement*, querySelector*
<https://javascript.info/searching-elements-dom>

 Document.getElementById() - Web APIs
<https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementById>

 HTML DOM Document Objects
https://www.w3schools.com/jsref/dom_obj_document

Extra info

🦋 Arrow functions, the basics

<https://javascript.info/arrow-functions-basics>

🔗 Arrow Functions: Fat and Concise Syntax in JavaScript

<https://www.sitepoint.com/es6-arrow-functions-new-f...>

👉 javascript - parseInt vs unary plus, when to use which? - Stack Overflow

<https://stackoverflow.com/questions/17106681/parsei...>

THANK YOU!