



Saptamana 6

Partea 1

Programare Front-End

JavaScript – Arrays

Array - este folosit sa stocam mai multe valori intr-o singura variabila

```
var fruits = ["Apple", "Banana", "Orange"]
```

- La fel si la array-uri, primul element are indexul 0

Methods and Properties:

length - returneaza numarul de elemente dintr-un array

```
var myArray = ["Madonna", "Eminem", "F.S."]  
var result = myArray.length;  
console.log(result) // 3
```

concat() - Concateneaza 2 arrayuri

```
var myArray = ["Madonna", "Eminem", "F.S."]  
var mySecondArray = ["Nightwish", "Haggard"]  
var result = myArray.concat(mySecondArray);  
console.log(result) // ["Madonna", "Eminem", "F.S.", "Nightwish", "Haggard"]
```

JavaScript – Arrays

isArray() - verifica daca un array este ...aray.

```
var myArray = [20, 30, 40, 50];  
var result = myArray.isArray();  
console.log(result) // true
```

join() - Transforma un array in **string**

```
var myArray = ["Unu", "Doi", "Trei", "Cincishpe"];  
var result = myArray.join();  
console.log(result) // "Unu,Doi,Trei,Cincishpe"
```

push() - Aduaga un element la sfarsitul array-ului.

```
var myArray = ["Unu", "Doi", "Trei", "Cincishpe"];  
myArray.push("cinci");  
console.log(myArray) // ["Unu", "Doi", "Trei", "Cincishpe", "cinci"]
```

JavaScript – Arrays

pop() - Scoate ultimul element din array

```
var myArray = ["Unu", "Doi", "Trei", "Cincishpe"];  
myArray.pop();  
console.log(myArray) //[ "Unu", "Doi", "Trei"]
```

shift() - Scoate primul element din array

```
var myArray = ["Unu", "Doi", "Trei", "Cincishpe"];  
myArray.shift();  
console.log(myArray) //[ "Doi", "Trei", "Cincishpe"]
```

unshift() - Adauga un element la inceputul array-ului

```
var myArray = ["Unu", "Doi", "Trei", "Cincishpe"];  
myArray.unshift("Number, zero");  
console.log(myArray) //[ "Number, zero", "Unu", "Doi", "Trei", "Cincishpe"]
```

JavaScript – Arrays

reverse() - inverseaza ordinea elementelor din array

```
var myArray = ["Unu", "Doi", "Trei", "Cincishpe"];  
myArray.reverse();  
console.log(myArray) //[ "Cincishpe", "Trei", "Doi", "Unu"]
```

sort() - sorteaza elementele din array

```
var myArray = ["Unu", "Doi", "Trei", "Cincishpe"];  
var numbersArray = [1, 2, 3, 5, 12, 13]  
myArray.sort();  
numbersArray.sort();  
console.log(myArray) //[ "Cincishpe", "Doi", "Trei", "Unu"]  
console.log(numbersArray) //[1, 12, 13, 2, 3, 5]
```

JavaScript – Arrays

Cum sortam numerele dintr-un array?

Funcția `sort()`, aranjează valorile ca și `string`-uri, din acest motiv atunci când avem `string`-uri le va lua după litere iar dacă avem numere vor fi transformate în `string`uri("1" "2" "13" etc) și "2" va fi după "13" pentru că "2" este mai mare ca "1".

Pentru a sorta un array de numere ne folosim de o **funcție** de comparație

```
var numbers = [40, 100, 1, 5, 25, 10];  
numbers.sort(function(a, b){  
    return a - b;  
});
```

Când `sort` compară 2 valori, trimite la sortat ceea ce rezultă din operația matematică(negativ, zero, pozitiv).

Dacă rezultatul este negativ **a** va fi înainte de **b**. Dacă e pozitiv **b** va fi înainte de **a**.

Putem folosi această funcție și pentru a sorta descrescător doar că vom face (return b - a).

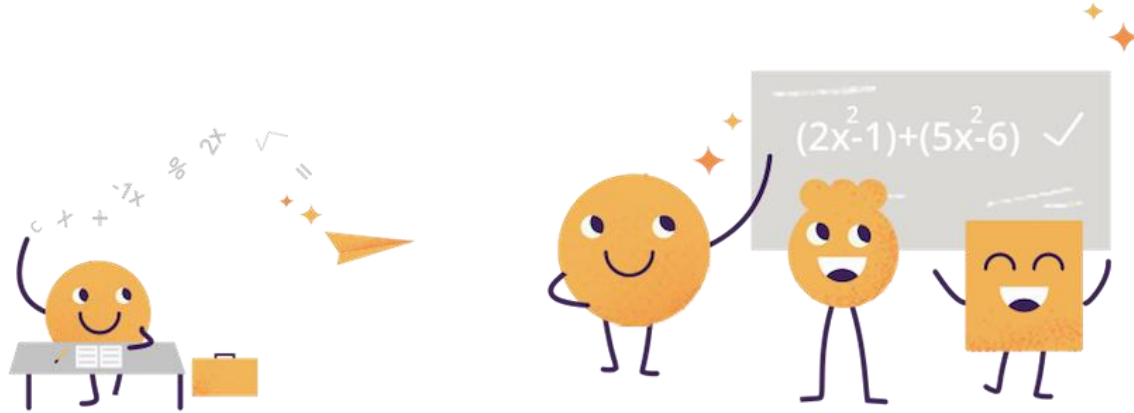
JavaScript – Arrays

splice() - Adauga/scoate elemente din array // `array.splice(index, howmany, item1,, itemX)`

```
var myArray = ["Unu", "Doi", "Trei", "Cincishpe"];  
myArray.splice(2,0,'patru');  
console.log(myArray) //["Unu", "Doi", "patru", "Trei", "Cincishpe"]
```

PRACTICE: Array methods and properties

<http://bit.do/ExArray1>



JavaScript OBJECTS

JavaScript – Objects

Data Types - String Number Boolean Null Undefined Symbol **OBJECT**

Obiectele - sunt folosite sa 'tina' o colectie de date si alte entitati complexe(de exemplu functii).

Obiectele - pot fi create cu acolade(brackets){..} ce contin o lista de proprietati.

O proprietate - este sub forma de **cheie:valoare** (**key:value**) unde cheia(key) este un string si mai este numita "property name" iar valoarea poate fi orice (number string boolean etc).



```
var user = new Object(); // 'object constructor'
```

```
var u
```



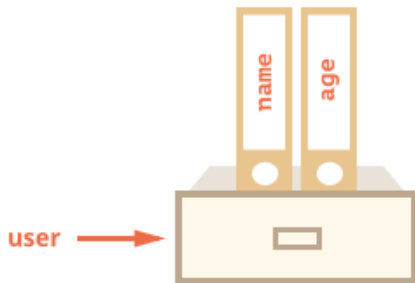
JavaScript – Objects

De obicei obiectele le vom crea cu sintaxa {..}

```
let user = {      // object
  name: "Ovidiu", // by key "name" store value "Ovidiu"
  age: 29         // by key "age" store value 29
};
```

În obiectul de mai sus avem 2 **proprietati**:

- Prima proprietate are numele **'name'** și valoarea **'Ovidiu'**.
- A doua proprietate are numele **'age'** și valoarea **'30'**.



JavaScript – Objects

Putem adauga proprietati noi intr-un obiect deja creat.

```
var user = {  
  name: "Ovidiu",  
  age: 29,  
};  
user.isAdmin = true;  
user["isTrainer"] = true;  
console.log(user); // {name: "Ovidiu", age: 29, isAdmin: true, isTrainer: true}
```

Sa stergem o proprietate putem folosi "delete"

Ex: `delete user.age;`

JavaScript – Objects

Cum accesam proprietatile unui obiect?

```
var user = {  
  name: "Ovidiu",  
  age: 29,  
  isAdmin: false,  
  isTrainer: true,  
};  
console.log(user.name); // "Ovidiu"  
console.log(user['age']); // 29
```

JavaScript – Objects

Putem pune si metode pe obiecte.

```
var user = {  
  name: "Ovidiu",  
  age: 29,  
  saySomething: function () {console.log('hello ppl')}  
};  
user.saySomething(); //"hello ppl"
```

JavaScript – Objects

```
var user = {  
  name: "Ovidiu",  
  age: 29,  
  saySomethingAboutUser: function () {  
    console.log(this.name + ' is ' + this.age + ' years old')  
  }  
};  
user.saySomethingAboutUser(); // "Ovidiu is 29 years old"
```

În această funcție **this** se referă la cine detine funcția respectiv **obiectul user** detine funcția *saySomethingAboutUser*.

Cu alte cuvinte **this.name** reprezintă proprietatea **name** a acestui obiect.

JavaScript – Objects

METHODS

Object.keys() and Object.values();

```
var user = {  
  name: "Ovidiu",  
  age: 29,  
  job: "Frontend dev"  
};  
var userKeys = Object.keys(user);  
console.log(userKeys); // ["name", "age", "job"]  
  
var userValues = Object.values(user);  
console.log(userValues); // ["Ovidiu", 29, "Frontend dev"]
```


JavaScript – Objects

Constructor function.

O alta metoda de a crea un obiect este prin intermediul unei functii constructor.

```
function Car(make, model, year) {  
  this.make = make;  
  this.model = model;  
  this.year = year;  
}
```

```
var mycar = new Car('Eagle', 'Talon TSi', 1993);
```

PRACTICE: Objects and arrays

<http://bit.do/ExObject1>

