



Saptamana 6

Partea 2

## Programare Front-End

# 1. JavaScript – Function types

# Funcție

- bloc care încapsulează instrucțiuni de cod ce performează un anumit *task*
- o funcție poate fi invocată în momentul în care este apelată la un anumit punct în cod, atunci când are loc un eveniment sau atunci când se autoapelează ( **event handler, function call, self-invoked** )

# Funcție – Sintaxa

```
function numeFuncție(parametruUnu, parametruDoi, parametruTrei) {  
    // blocul funcției - codul care se execută în momentul în care este apelată  
}
```

## 1.1 Named functions – Function declaration

# Functie – Declararea unei functii ( Named function )

```
function numeFunctie(parametruUnu, parametruDoi, parametruTrei) {  
    // blocul functiei - codul care se executa in momentul in care este apelata  
}
```

- Functiile declarate nu sunt executate imediat
- Acestea sunt "salvate" in memorie pentru a fi utilizate mai tarziu - prin invocare: *numeFunctie()*;

## 1.2 Function expressions

# Function expressions

- o functie in JavaScript poate fi definita si in alt mod: folosind o expresie
- o **function expression** poate fi identificata printr-o instructiune de forma:

```
var x = function (a, b) {return a * b};
```

- ulterior, variabila poate fi folosita ca si functie:

```
var z = x(4, 3);
```



## 1.3 Anonymous functions

# Anonymous functions

```
var x = function (a, b) {return a * b};
```

- Functia specificata in cadrul expresiei de mai sus poate fi deasemenea numita **functie anonima** - este o functie fara nume

# Immediately Invoked Function Expression

- **IIFE**, sau **Immediately Invoked Function Expression** este o functie care se autoapeleaza
- sintaxa specifica unei **IIFE** este urmatoarea:

```
(function () {  
    var x = "Hello!!"; // I will invoke myself  
})();
```

## 2. JavaScript Conditionals

# JavaScript – tipuri de conditii

- Folosim **if** pentru a defini un bloc de cod care trebuie executat daca o anumita conditie este **true**
- Folosim **else** pentru a defini un bloc de cod care trebuie executat daca aceeaasi conditie este **false**
- Folosim **else if** pentru a specifica o noua conditie de test daca prima conditie este falsa
- Folosim **switch** pentru a specifica mai multe blocuri de cod alternative pentru a fi executate in cazul unei conditii indeplinite

# if

```
if (conditie) {  
    // bloc de cod care se executa daca conditia returneaza true  
}
```

# if ... else

```
if (conditie) {  
    // bloc de cod care se executa daca conditia este true  
} else {  
    // bloc de cod care se executa daca conditia este false  
}
```

## if ... else if

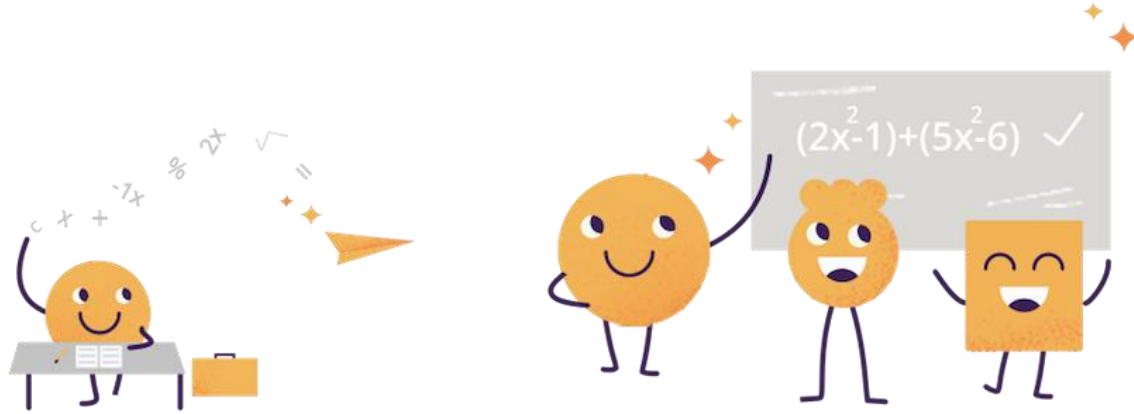
```
if (prima-conditie) {  
    // bloc de cod care se executa daca prima conditie este true  
} else if (a-doua-conditie) {  
    // bloc de cod care se executa daca prima conditie este false si a doua este true  
} else {  
    // bloc de cod care se executa daca ambele conditii anterioare sunt false  
}
```



# switch

```
switch(expresie) {  
    case x: // o posibila valoare returnata de expresie  
        // bloc de cod  
        break;  
    case y:  
        // bloc de cod  
        break;  
    default:  
        // bloc de cod  
}
```

# PRACTICE: Functions and conditionals



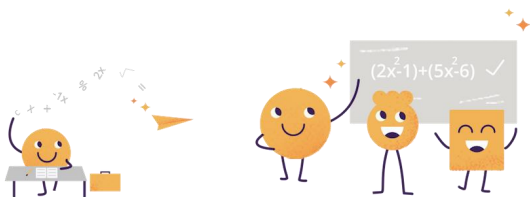
# PRACTICE: Functions and conditionals

## Cerinta:

1. Scrieti o functie de tip **named function** denumita **greatestNrBetween** care:

- Primeste ca argumente doua numere
- Returneaza numarul care este mai mare

Apelati functia de doua ori si afisati pe ecran textul: "Numarul mai mare dintre 5 si 10 este 10"



# PRACTICE: Functions and conditionals

## Cerinta:

2. Scrieti o functie de tip **function expression** asociata cu numele **helloWorldInLang** care:

- Primeste ca argument codul unei limbi: "en", "es", "de"
- Returneaza "Hello World" pentru limba corespunzatoare ( cel putin 3 limbi )
- Daca limba nu este specificata, trebuie sa returneze textul pentru limba engleza

Apelati functia pentru fiecare caz in parte.

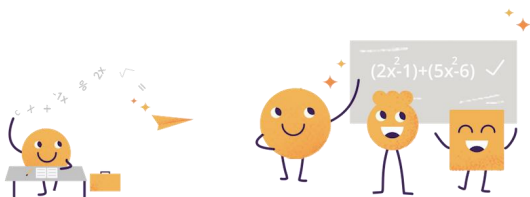


# PRACTICE: Functions and conditionals

## Cerinta:

3. Scrieti o functie numita **pluralizeWord** care:

- accepta doua argumente: un substantiv in engleza si un numar
- returneaza numarul si forma de plural corespunzatoare substantivului, precum: "5 cats", "1 dog"
- adaugati cateva cazuri particulare pentru cuvinte precum "sheep" sau "geese"



# PRACTICE: Functions and conditionals

## Cerinta:

4. Scrieti o functie care accepta ca argument venitul anual al unui angajat in dolari (ex: \$30,000) si calculeaza taxa corespunzatoare considerand urmatoarele criterii:
- daca angajatul castiga anual o suma mai mica de \$30,000, taxa este 0%
  - daca venitul anual se situeaza intre \$50,000 - \$99,999, taxa calculata trebuie sa fie 35%
  - pentru venit mai mare de \$100,000, taxa se calculeaza ca procentaj de 40%

Returnati un text care sa aiba un format asemanator: "Pentru un venit anual de x, aveti de platit taxe in valoare de

y"

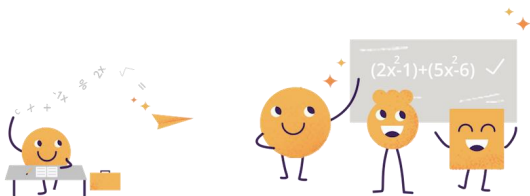


# PRACTICE: Functions and conditionals

## Cerinta:

5. Scrieti in doua variante o functie care accepta ca argument o valoare numerica pentru un parametru numit **points** si afiseaza:

- AA if points is greater than 90
- AB if points is greater than 80 and less than or equal to 90
- BB if points is greater than 70 and less than or equal to 80
- BC if points is greater than 60 and less than or equal to 70
- CC if points is greater than 50 and less than or equal to 60
- CD if points is greater than 40 and less than or equal to 50
- DD if points is greater than 30 and less than or equal to 40
- FF if points is less than or equal to 30



# PRACTICE: Functions and conditionals

## Cerinta:

6. Scrieti o functie care testeaza daca o anumita data introdusa este weekend:

Test Data :

```
console.log(is_weekend('Nov 15, 2014'));  
console.log(is_weekend('Nov 16, 2014'));  
console.log(is_weekend('Nov 17, 2014'));
```

Output :

```
"weekend"  
"weekend"  
undefined
```

