wantsome
The friendly IT Academy

Saptamana 7

Partea 1

Programare Front-End

1. JavaScript - Loops

JavaScript - tipuri de “bucle” - loops

- **for** - itereaza un bloc de cod de n ori
- **for...in** - itereaza proprietatile unui obiect
- **while** - itereaza un bloc de cod atat timp cat o anumita conditie este indeplinita
- **do...while** - itereaza un bloc de cod atat timp cat o anumita conditie este indeplinita

for

```
for (statement 1; statement 2; statement 3) {
  // bloc de cod care trebuie executat
}
```

```
for (i = 0; i < 39; i++) {
  text += "Contorul este:" + i + "<br>";
}
```

```
for (i = 0, len = users.length, text = ""; i < len; i++) {
  text += users[i] + "<br>";
}
```

for...in

```
var userDetails = { name:"Ovidiu", surname:"Grig", varsta: 18 };

var userInfoText = "Detaliile utilizatorului sunt: ";

for (var key in userDetails) {
  userInfoText += userDetails[key] + “, “;
}
```

while

```
while (conditie) {
  // bloc de cod care trebuie executat
}

while (i < 39) {
  text += "Contorul este: " + i + "<br>";
  i++;
}
```

do...while

```
do {
  // bloc de cod care trebuie executat
}
while (conditie);
```

```
do {
  text += "Contorul este: " + i;
  i++;
}
while (i < 39);
```

do...while

```
do {
  // bloc de cod care trebuie executat
}
while (conditie);
```

```
do {
  text += "Contorul este: " + i;
  i++;
}
while (i < 39);
```

2. JavaScript - Loops - Stop/Continue iterations

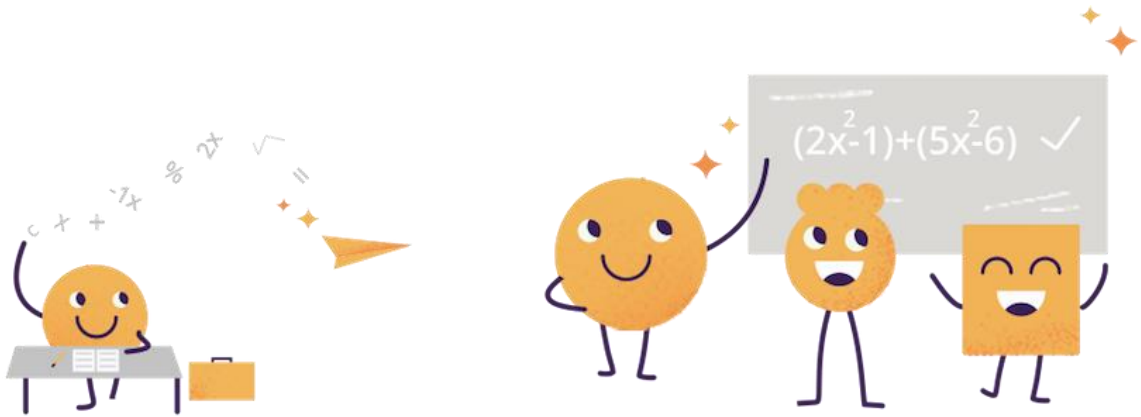
break

```
var i = 99;
while (true)
{
    console.log(i + " oameni mai sunt in asteptare");
    i -= 1;
    if (i == 0)
    {
        break;
    }
}
```

continue

```
for (var i = 0; i < 100; i++)
{
    // verificam daca valoarea numerica pentru i din cadrul iteratiei curente este para
    if (i % 2 == 0)
    {
         continue;
    }
    // daca executia ajunge aici, inseamna ca i este impar
    console.log(i + " este numar impar.");
}
```

PRACTICE: **JavaScript Loops**



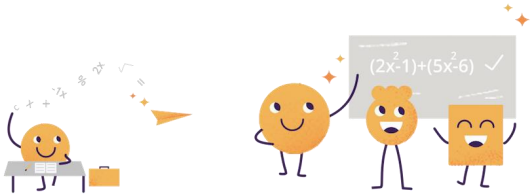
PRACTICE: **Loops**

Cerinte:

1. *Scrieti o functie care accepta ca **argument o valoare numerica** si afiseaza de atatea ori pe ecran textul: “Sunt la iteratia numarul [index]” - Scrieti codul in doua variante*

1. *Scrieti o functie care accepta ca **argument un array** si afiseaza in consola elementele acestuia prin parcurgerea sa - Scrieti codul in doua variante*

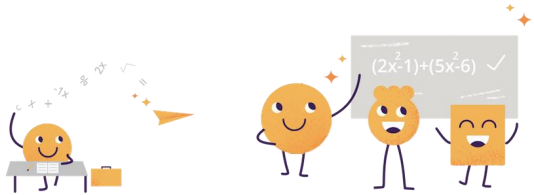
1. *Scrieti o functie care accepta ca **argument un obiect** si afiseaza in consola elementele acestuia prin parcurgerea sa - scrieti codul in doua variante*



PRACTICE: **Loops**

Cerinte:

1. `0 100 200 300 400 500 600 700 800 900 1000`
2. `1 2 4 8 16 32 64 128`
3. `0 2 4 6 8 10`
4. `3 6 9 12 15`
5. `9 8 7 6 5 4 3 2 1 0`
6. `1 1 1 2 2 2 3 3 3 4 4 4`
7. `0 1 2 3 4 0 1 2 3 4 0 1 2 3 4`



PRACTICE: **Loops**

Cerinte:

Implementati functia `stream`. Trebuie sa:

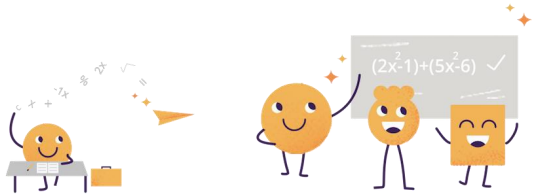
- accepte ca argumente 2 functii: `conditionalFn`, `actionFn`.
- apeleaza `actionFn` pana cand `conditionalFn` returneaza fals
- nu returneaza nimic

Exemplu 1:

```
conditionalFn = function() { return false; };
actionFn = function() { console.log("Vreau sa ies la
tabla!"); };
stream(conditionalFn, actionFn); // Nu afiseaza nimic
```

Exemplu2:

```
counter = 10;
conditionalFn = function() {
  counter--;
  return counter >= 0;
};
stream(conditionalFn, actionFn); // Afiseaza “Vreau sa
ies la tabla” de 10 ori
```



PRACTICE: **Loops**

Cerinta:

Implementati functia `computeSumOfArrayElements`.

Trebuie sa:

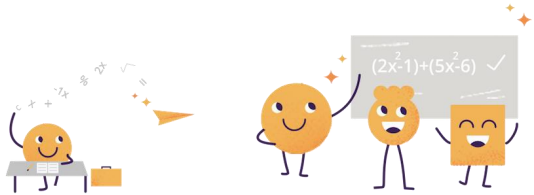
- **accepte ca argument un array format din numere**
- **returneze suma numerelor din array**
- **foloseasca instructiunea `while`**

Exemplu 1:

```
computeSumOfArrayElements([]); // returneaza 0
```

Exemplu 2:

```
computeSumOfArrayElements([1,2,3,4]); // returneaza 10
```



wantsome
The friendly IT Academy

https://js.checkio.org/

https://codewars.com/