IT-Academy.BG

Цикли в JavaScript

Димитър Митев

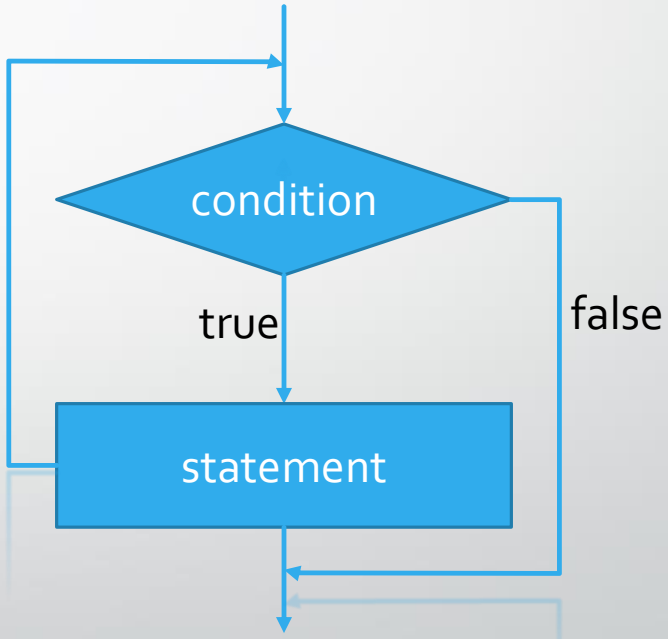
Цикли (loops)

- Позволяват *повторението* на *дадено* парче код *определен брой* пъти или докато дадено *условие* е *изпълнено*
- Съществуват цикли, които се изпълзняват постояно. Те се наричат *безкрайни цикли (infinite loops)*

while цикъл

- Най-лесния и може би най-използван цикъл
- Условието (*condition*) е стандартен булев израз или променлива, като може да бъде и *false-like* или *true-like*
- Ако условието не е истина, то тялото на цикъла няма да се изпълни нито веднъж

```
while (condition) {
    statements;
}
```



break оператор

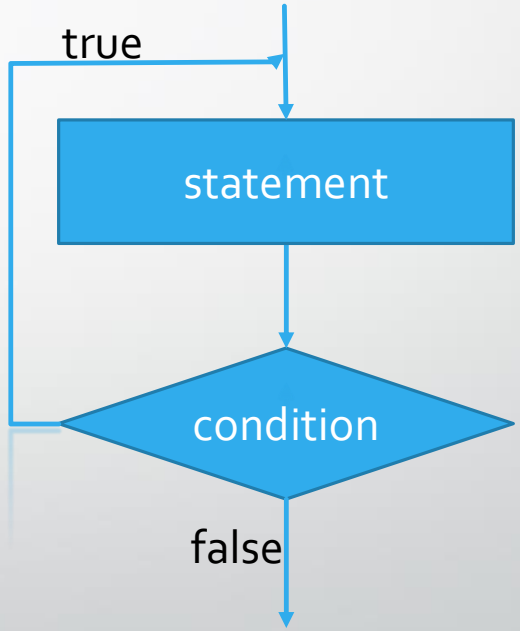
- Може да се използва, за да се прекрати изпълнението на даден цикъл
- *break* операторът прекратява изпълнението на най-близкия /най-вътрешния/ цикъл

```
var n = 0;
while (1) {
      n++;
      if ( n == 10 ) {
            break;
      }
}
```

do-while цикъл

- Условието (*condition*) е стандартен булев израз или променлива, като може да бъде и *false-like* или *true-like*
- Тялото на цикъла ще се изпълни поне веднъж, без значение дали улсловието е *true* или *false*

```
do {
    statements;
} while(condition);
```



for-цикъл

- Най-често използвания цикъл
- Състои се от
  - Инициализация (initialization)
  - Условие, което се оценява до булев израз (test expression)
  - Обновнление (update statement)
  - Тяло на цикъла (loop body)

```
for (initialization; test; update) {
    // loop body
}
```

for-цикъл. Инициализация

- Инициализацията се извършва преди да стартира изпълнението на цикъла
- Използва се за деклариране на променливи преди да е започнало изпълнението на цикъла
- Подходящо място за инициализацията на броячи

```
for (var i = 0; i < 10; i++) {
    // loop body
}
```

for-цикъл. Проверка

- Оценява се преди всяка итерация на цикъла
  - Ако условието е изпълнено, изпълнява се тялото на цикъла
  - Ако условието **не е** изпълнено, **не се** изпълнява тялото на цикъла

```
for (var i = 0; i < 10; i++) {
    // loop body
}
```

for-цикъл. Обновление

- Изпълнява се всеки път след като се е изпълнило тялото на цикъла
- Изпълнява се преди да се направи оценка на условието
- Подходящо е да се обновява стойността на брояча

```
for (var i = 0; i < 10; i++) {
    // loop body
}
```

Вложени цикли

- Цикли, които се намират в други цикли
- Изпълнението им започва от най-външния
- Най-вътрешния се изпълнява, докато приключи
- След като най-вътрешния приключи своето изпълнение, се преминва към следващия отвътре навън

```
for (initialization; test; update) {
    for (initialization; test; update) {
        // loop body
    }
    ……
}
```

for-in цикъл

- Работата му е сходна с тази на *for* цикъла
- Използва се за итериране по пропъртита на даден обект
- Трябва да се внимава дали се итерират собствени пропъртита или вградени

```
for (var key in object) {
    // loop body
}
```

for-of цикъл

- Използва се за итериране на масиви и масивоподобни обекти
- Дава имплементация на т.нар. Iterator шаблон за дизайн
- Навлиза с EcmaScript 2015
- Поддържа се само от последните версии на браузърите

```
for (var key of object) {
    // loop body
}
```


НЯКОЙ ИМА ЛИ
ВЪПРОСИ?

IT-Academy.BG

Домашна работа

1. Напишете JS код, който по зададен интервал, да намира всички числа, които се делят на 3.
2. Напишете JS код, който да намира всички „щастливите числа". Ако A+B = C + D, то числото ABCD е щастливо. Например: 1322
3. Напишете JS код, който намира най-голямото и най-малкото просто число в даден интервал
4. Напишете JS код, който обхожда и принтира в конзолата всички пропъртитета на *window* обекта