



НИ Е ВЪРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ

JavaScript

- Езиците за програмиране са изразно средство, близко до човешкия език, което използваме за да зададем някаква логика, отнасяща се към някакви определени събития, като искаме тази логика да се изпълни, когато въпросните събития се случат
- При динамичните езици като JavaScript, изпълнението на програмата зависи от външни фактори (например от user input), така че нашата логика обикновено се съобразява с тях

- Представете си че искаме да следим кога входната врата ще се отвори, но без да я виждаме.
- Едно от решенията би било да сложим звънче, така че при всяко отваряне то да звъни и ние да го чуваме
- (да, това ще бъде “събитие”)

- Нашата програма ще представлява следната логика:
 - Ако се възникне събитието (звънчето звъни), то
 - ние ще го чуем (регистрация на събитието)
 - Дори бихме могли да предположим, че някой влиза през вратата и да кажем “Добър ден”
 - (да, това ще е обработка на събитието)

- Това дали някой ще влезе е външен фактор и не зависи от нас
- Затова и програмата ни е динамична
- Логиката ни обикновено трябва да следва модела “Ако ..., значи трябва да ...”

- JavaScript е интерпретативен език, което означава, че командите които използваме за нашата програма, се интерпретират по определен начин от брауъра до получаването на конкретен резултат
- брауърът (т.е. JavaScript енджин-а) интерпретира кода ред по ред и така на всеки ред прави някакво изчисление или операция
- JavaScript конзолата е един интерпретатор, който ни позволява да изписваме ред по ред JS код, като за всеки ред получаваме междинните резултати

Коментари

- Както във всеки друг език, така и в JavaScript, можем да пишем произволен текст като коментар, който не се интерпретира от парсер-а
- Има два вида коментари:

- едноредови:

`alert('hi!');` // от тук до края на реда, всичко се игнорира

- многоредови:

`/* Този коментар би могъл да бъде поставен навсякъде
както и да се разпростира на няколко реда */`

Синтаксис

- Както обикновенните езици, така и тези за програмиране си имат синтаксис.
- Синтаксиса на JS ще взимаме в движение, но като за начало можем да кажем следните 2 основни неща:
 - всяка команда (ред) завършва с ‘;’
 - всяка “дума”, която не е дефинирана от нас, не е дума от езика и не е коментар, води до съобщение за грешка

Променливи

- Променливите са именувани контейнери за междинни резултати
- Както се подразбира от името, те могат да имат най-различни стойности
- Могат и в някакъв момент да сменят стойността си
- Синтаксиса за създаване на променлива е следният

```
var myVar = 3;
```

- Може и така:

```
var myVar;
```


Приетият стил за именуване на променливи в JS е да са максимално експлицитни!

Т.е. английски думи, като ако името на променливата ни се състои от няколко думи, пишем ги заедно и първата дума започва с малка буква, а всяка следваща с главна:

```
var myAwesomeVar = 1;
```

Основни типове данни

Undefined

- Когато нещо няма стойност
- `var myVar; // undefined`

Undefined vs Null

- Можем да зададем стойност null, което означава нищо
- Разликата е че null е обект а undefined е нещо което не е дефинирано
- И все пак:

```
null == undefined // true
```

```
typeof(null) // "object"
```

```
typeof(undefined) // "undefined"
```


Числа (Numbers)

- Цели числа: например '2' или '-10'
- Числа с плаваща запетая: '3.5'
- С числата можем да извършваме основните аритметични операции:
 - събиране: $2+3$
 - изваждане: $10-4$
 - умножение: $2*3$
 - деление: $10/5$

Числа (Numbers)

- Особености
 - операции със скоби - както в математиката
 - деление на нула: Infinity
 - при деление резултата понякога е число с плаваща запетая, ако искаме да го направим цяло, можем да използваме методът `round` на класа `Math`, ето така:

```
Math.round(8/3)  
=> 3
```

- Понякога искаме да разберем какъв е остатък при деление (модул), това става по следния начин:

```
8%3 // 2
```


Текстов низ (String)

- Всеки набор от символи, заграден в двойни или единични кавички е String

`"hello"`

`"1"`

`"Some very long text inputs are also strings"`

`'a'`

`""`

Основни операции със низове

- Съединяване (append):

```
"hello" + ", world!"
```

- Дължина:

```
"hello, world!".length // 13
```

- Символ на определена позиция:

```
"hello, world!".charAt(4) // 'o'
```

- Индекс (позицията) на определен символ:

```
"hello, world!".indexOf('l') // '2'
```

```
"hello, world!".indexOf('x') // '-1'
```


Списъци (Arrays)

- Наричат се също масиви
- Това е колекция от елементи, например числа и низове
- записва се, като елементите на колекцията се изреждат между квадратни скоби, разделени със запетайки:

```
['a', 1, 'w']  
// => ["a", 1, "w"]
```

- Използваме ги за да групираме най-често еднотипни елементи, за да можем да изпълняваме определени действия върху всички елементи от списъка

Списъци (Arrays)

- Основни действия и особености

- Дължина:

```
[1, 2, 3].length // 3
```

- Достъп до елемент

```
["hello", "world"][0] // "hello"
```

- Индекс (позицията) на определен елемент:

```
["hello", "world"].indexOf("world") // 1
```


Булеви стойности (Boolean)

- true и false
- Използват се за да кажат дали нещо е вярно или невярно
- Получават се при изпълнение на логически операции като:
 - Логически операции с числа: '<', '>', '==', '>=', '<=', '!='
 - Логически оператори и сравняване на низове:

```
if (userInput == "buy") alert("Out of stock")
```

Въпроси?



**KEEP
CALM
AND
LEARN
JAVASCRIPT**

Примери

<http://zenifytheweb.com/courses/lessons/lesson10/example/index.html>

Домашно

<https://github.com/zzeni/swift-academy-homeworks/tree/master/tasks/L10>