

Променливи

- Променливите са именувани *указатели*, които използваме, за да съхраняваме в тях данни
- Както се подразбира от името, те могат да имат най-различни стойности, които не са постоянни
- Представяйте си ги като имена или етикети на дадени стойности/обекти
- Синтаксиса за създаване на променлива е следният:

```
var pi = 3.14;    // стойността на pi е 3.14
```

- Може и така (без задаване на начална стойност):

```
var myVar;        // стойността на myVar е undefined
```

Именуване

Приета е следната конвенция за именуване на променливи в JavaScript:

1. само латински букви + цифри + "_"
2. имената да са максимално експлицитни (описателни)
3. използват се думи от английския език
4. когато името на променливата се състои от няколко думи, то ги изписваме "слято", като първата дума започва с малка буква, а всяка следваща с главна (CamelCase):

```
var myAwesomeVar = 1;
```

Undefined

- Когато нещо няма стойност:

```
var myVar;      // undefined  
[1, 2, 3][4];  // undefined
```

- Дефинирани се водят променливите, на които сме задали конкретна стойност

- Грешката:

ReferenceError: variable is not defined

се получава, когато променливата *variable* не е била създадена със специалната дума: **var**

Логически и аритметични операции

- Аритметичните операции са тези, които се извършват върху числови данни и резултатът от тях е пак число (+, -, *, /, % и т.н.)
- Логическите операции се използват за сравнение на данни или за формиране на условие и те могат да се използват за всякакви видове данни. При тях резултата е **true** или **false**
- Логически оператори: <, >, ==, !=, ===, !==, !, &&, || и т.н.
- Логически изрази:
 - `(2 > 1) && (3 < 5) // true`
 - `(age > 18) || (parent !== undefined)`

Логически сравнения и оператори

> , >=	"по-голямо", "по-голямо или равно" (за числа)
< , <=	"по-малко", "по-малко или равно" (за числа)
&&	логическо "и"
	логическо "или"
!	логическо отрицание

Логически сравнения и оператори

==	сравнение за "еднаквост" по стойност
===	сравнение за "еднаквост" по стойност и по тип
!=	сравнение за "различност" по стойност
!==	сравнение за "различност" по стойност и по тип

http://www.w3schools.com/js/js_comparisons.asp

f(js)

функции в JavaScript

Функции

- Това е едно от най-фундаменталните неща в JavaScript (и в повечето езици за програмиране)
- Използват се за съхраняване на код, който ще се изпълнява при всяко извикване на функцията
- Функциите могат да приемат входящи стойности (аргументи)
- Функциите прилагат кода, който съхраняват върху получените при извикването си аргументи
- Функциите могат да връщат стойност (резултата от прилагането на кода в тях върху аргументите)

СИНТАКСИС

```
function myFunctionName(myParam1, myParam2, ...) {  
    var result = "set initial result value";  
    // do some calculations  
    return result;  
}  
  
// calling (executing) the function with arguments:  
console.log(myFunctionName(42, 'param 2 value'));
```

The return statement

- За да върнем стойност с нашата функция, използваме запазената дума `return`
- Когато изпълним функцията, резултатът от нея ще бъде стойността върната с `return`
- Ако функцията не изпълни `return`, то резултатът от нея ще бъде `undefined`
- Във функцията можем да имаме повече от един `return`, като изпълнението ѝ ще приключи при достигането до първият от тях или при достигането до края на тялото на функцията (което се случи по-напред)

Вградени функции

- JavaScript предлага известен брой 'вградени' (built-in) функции, които са включени в езика по подразбиране и могат да се ползват наготово
- Примери за built-in функции са `Array.indexOf()`, `Math.round()` и `Date.now()`, където `Array`, `Math` и `Date` са глобални обекти (ще стигнем и до там)
- http://www.tutorialspoint.com/javascript/javascript_builtin_functions.htm

Извикване на функции

- След като сме си запазили някаква последователност от изчисления във функция, бихме могли да я изпълним във всеки момент, в който си поискаме
- действието, което кара функцията да се изпълни, се нарича "извикване"
- извикването на функцията става, като изпишем името ѝ, последвано от кръгли скоби: `fn()`
- извикването може да бъде със или без аргументи: `fn(x)`

Въпроси?



**KEEP
CALM
AND
LEARN
JAVASCRIPT**

Полезни връзки

- Codecademy:
<https://www.codecademy.com/learn/javascript>
- Codeschool:
<https://www.codeschool.com/courses/javascript-road-trip-part-1>
- Интерактивен онлайн интерпретатор:
<https://repl.it/languages/javascript>
- Загубалка на време:
<https://blockly-demo.appspot.com/static/demos/interpreter/index.html>