# [http://www.fmi.uni-sofia.bg/logo.jpg](http://www.fmi.uni-sofia.bg/)

# Тема 22: CSS & JavaScript (JavaScript - част 1): Типове данни и променливи в JS

**Разработено от**: Мартин Маринов  
**Дата на защита:** 15.05.2013 (номер в група: 2)

**Ключови думи**: wwwTech,2013, реферат, software engineering, web, JS, data types, variables, examples, introduction

## Описание

Презентацията ще запознае накратко аудиторията със скриптовия език JavaScript, неговите особеноси и по-конкретно типовете данни, които езика поддържа и променливите, с които оперира.

## Анкета

**Групово:** Задайте един въпрос върху реферата и сложете ресултатите (въпроса ще го сложа в екселската таблица). Например: QЗанимавали ли сте се с HTML5? a) да – много, 2)да – малко 3) чувал съм, но не не съм писал; 4) нямам идея – **ще дам отделен гугъл-докс за гласуване (или в дудъл, ще видя) – и резултатите в таб3 (03\_VoteQuiz)**

## Съдържание

Кратко описание на JS, по-задълбочено разглеждане на типовете данни и променливите в скриптовия език, примери, заключение

## Теория

# Типове данни и променливи в JavaScript

## Кратко въведение

JavaScript е най-широко разпространеният език за програмиране в интернет, след HTML . Въпреки че го наричат "език за програмиране", с негова помощ не се пишат програми, а скриптове, които се вмъкват в HTML документа. В този смисъл JavaScript е език за писане на скриптове, докато JAVA  е език за програмиране. Освен съвпадението в част от името, двата езика нямат кой знае какви прилики, дори са разработени от различни корпорации ( JAVA е дело на SUN, a JavaScript е разработка на Netscape). JAVA е мощен език за програмиране не само на интернет приложения, но и на самостоятелни програми за различни платформи. Интернет приложенията на JAVA се наричат аплети. Те са файлове с разширение .class и се вмъкват в HTML документа между таговете <applet> и </applet>.

JavaScript кода се вмъква в HTML документа между двойката елементи <script> и </script>. Когато срещне тагът <script>, браузъра разбира че трябва да спре интерпретирането на HTML кода и да започне да обработва скрипта, намиращ се между <script> и </script>. Този скрипт не е задължително да бъде написан на JavaScript. Има и други езици за  писане на скриптове, например VBScript. Затова когато пишете отварящия таг за скрипт, трябва да укажете на браузъра на какъв език ще бъде скрипта. Ако този език е JavaScript трябва да напишете <script language=”javascript”> .Ето един пример на една уеб страница с вмъкнат JavaScript:

[Пример 1:](example1/Example1.html)

<html>  
<head>  
 <title>JavaScript</title>  
</head>  
<body>  
 <script lang = ”javascript”>  
   document.write("Здравей!");  
 </script>  
</body>  
</html>

Изпълнението на кода ще изведе на екрана съобщение: „Здравей!“

Важно уточнение!

JavaScript и Java са два напълно различни езика, когато става въпрос за концепции и дизайн. Java (създаден от Sun) е по-комплексен програмен език, подобен на C. JavaScript е създаден за 10 дена от Brendan Eich като проект на Netscape и е front-end скриптов език. Харатеризира се с лекота, може да бъде използван кат обектно-ориентиран език, ембднат в html страницата, интерпретира се от браузъра.

[Пример 2:](example2/Example2.html)

<html>

<body>

<script type=”text/javascript”>

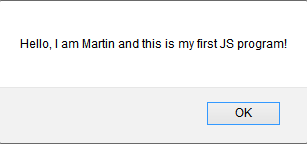
alert (‘Hello, I am Martin and this is my first JS program!’);

</script>

</body>

</html>

Ето какво ще видим след изпълнението на кода:



От Пример 1 и Пример 2 стана ясно, че командите document.write и alert са сходни, с тази разлика, че alert изкарва резултата в малко прозорче, а document.write просто отпечатва някакъв текст на екрана и пише директно в html документа.

Интересен факт:

JavaScript в Windows 8 – Microsoft използва JavaScript за разработка на Windows 8 приложения. Може да се каже, че JavaScript определено е бъдещето за Internet и Windows.

Силно препоръчително е, JavaScript кода да е поставен в изцяло отделен файл с разширение .js и така да бъде линкнат (свързан) с html файла. Ето как става това:

[Пример 3:](example3/Example3.html)

<html>

<head>

<script src=”Example3.js” type=”text/javascript">

<!-- Кодът поставен тук няма да бъде изпълнен! -->

</script>

<head>

<body>

<button onclick=”sample()” value=”Call JS function from Example3.js” />

</body>

</html>

Чрез атрибута src, задаваме пътят/името на нашия JavaScript файл и така при отваряне на html файла, ще се изпълни и script.js. Важно е да се спомене, че <script> тага винаги е празен и съответно всички команди пишем в .js файла (подобно на CSS).

## 2. Променливи

Променливите в JavaScript са букви(последователност от букви), в които се съдържа определена информация. Може да се каже, че променливите са „контейнери“ за съхранение на някаква информация. Тази информация може да се извика при нужда и да се обработва допълнително.

Променливите в JavaScript имат:

* Име
* Тип ( на съдържаната информация)
* Стойност

Пример 4:

var counter = 5;

Променливите в JavaScript се декларират с ключовата дума var. В този случай, името на променливата е counter, типа е integer, стойността е 5.

Когато декларираме дадена променлива ние задаваме име, което се нарича идентификатор. При формирането на индетификаторите трябва да се спазват някои правила. Името може да съдържа големи и малки латински букви, както и знака за подчертаване \_ . Всички останали символи са забранени. Имената на променливите трябва да започват винаги с буква или знак за подчертаване, но никога с цифра. Не трябва да съдържа ключова дума.

Правилно: greeting\_card ; greet2 ; \_my\_greeting ;

Грешно: 123 ; var.2 ; greeting 3 ;

Добра практика е да декларираме всички променливи, които ще ни трябват на едно място в началото на кода!

Също така е препоръчително имената на идентификаторите да са смислени. Не трябва да бъдат прекалено дълги или прекалено къси. **Важно** е да се уточни, че имената са case sensitive, тоест има разлика дали ще започват с главна или малка буква.

Задаването на стойност на променливите става с помощта на знака равно „=“. Отляво на оператора „=“ трябва да има индентификатор, а отдясно стойност на променливата. Възможно е да се използва при каскадното повикване (cascade calling), където възлагането става от дясно на ляво(препоръчително е да се избягва). Също така е възможно да се декларират няколко променливи на един ред просто като ги разделим със запетаи:

Пример 5:

var firstname=”Martin”, lastname=”Marinov”, age=20 ;

Променливите в JavaScript могат и да се редекларират:

Пример 6:

var firstname=”Martin”;

var firstname;

В този случай сме задали стойност Martin на променливата firstname. На следващият ред декларираме отново същата променлива, но не и задаваме стойност. В крайна сметка, стойността на firstname няма да се промени и ще си остане Martin.

## 3. Типове данни

Компютрите са машини, които обработват информация. Тази информация се съхранява в компютърната памет като променливи. Променливите имат име, тип и стойност.

Типовете данни са поле от стойности със сходни характеристики. Чрез тях се дефинира вида на информацията съхранена на компютърната памет (като променилива)

Типовете данни в JavaScript са:

Примитивни

* Integer
* Floating-Point
* Boolean
* String

Комбинирани

* Обекти (оbjects)
* Масиви (arrays)

**Integer**  – представляват всички числа. Имат диапазон на стойностите, в зависимост от размера на използваната памет. Поддържа числа от -9007199254740992 до 9007199254740992. Могат да се представят в осмична, десетична и шеснайсетична бройна система. Най-често в JavaScript се използва десетичен запис на числата.

Пример 7:

var age = 20;

var maxInt = 9007199254740992;

var minInt = -9007199254740992;

**Floating-Point –** репрезентира всички реални числа. Има диапазон на стойностите и прецизност и може да се държи [необичайно\*](#abnormally) при някои сметки. JavaScript представя числата използвайки IEEE 754 стандарта за цифрена изобразяване. Това означава, че може да пишем големи числа като 1,79797x10308 и малки като 5x10-324. Размера на Floating-Point типа зависи от платформата – браузера и операционната система. 32-битовите операционни системи и браузъри имат 32 бита за число, докато 64-битовите имат 64 бита. Добра практика е да използваме до 32-битови числа, което ще ни гарантира, че ще фунционира на всички платнофми.

**\*- Понякога при използването на floating-point типа, могат да се наблюдават някои анормални последици. Например, сравняването на числа от floating-point тип не може да стане директно чрез „==“ оператора:**

**Пример 8:**

**var a = 1.0;**

**var b = 0.33;**

**var sum = 1.33;**

**var equal = (a+b == sum) // False!**

**console.log(“a+b = ”+ (a+b) + “, sum = “+ sum + “, sum == a+b? is “ + equal);**

**Като допълнение е нужно да се спомене, че в JavaScript се поддържат числа със специална стойност:**

* **NaN (not a number) – Използва се, когато някоя математическа операция е изпълнена с неподходяща информация, като например стрингове.**
* **Positive infinity (+ безкрайност) – Използва се, когато някое положително число е прекалено голямо, за да се репрезентира от JavaScript.**
* **Negative infinity (- безкрайност) – Използва се, когато някое отрицателно число е прекалено голямо, за да се репрезентира от JavaScript.**
* **Positive and Negative 0. JavaScript прави разлиа между положителна и отрицателна 0.**

**Boolean – може да има само две стойности – true (вярно) и false (грешно). Булевата стойност е истинно-стойностна, тоест определя кога поставеното условие е вярно. Много полезна при логическите изрази и условните оператори:**

**Пример 9**

**if (x == 2000) {**

**z = z + 1;**

**}**

**else {**

**x = x + 1;**

**}**

**В случай, че стойността на x e равна на 2000, тоест булевата стойност е true, оператора изпълнява това, което е в първите скоби (z = z + 1). Ако е false, се изпълнява x = x + 1. В** [Пример 10](example10/Example10.html) **отново е показано на какъв принцип действат булевите типове данни с помощта на условния оператор if.**

**String – представлява поредица от символи (букви, цифри, пунктуационни знаци и др.) Използва се за да се пресъздаде текст в JavaScript. Добавят се в скрипта като се ограждат с единични или двойни кавички. Стринговете могат да бъдат конкатенирани, чрез „+“ оператора. В следващият пример е показано как може удобно да се използват конкатенирани стрингове:**

[Пример 11](example11/Example11.html)

**var firstname = “Martin”;**

**var lastname = “Marinov”**

**var fullname = firstname + “ “ + lastname;**

**alert (“My full name is ” + fullname);**

## Примери – съдържание

## Тестови въпроси

*(изтрийте този параграф -> напишете от 3 до 10 тестови въпроса един верен от много върху реферата-> всеки въпрос да е с поне 4 избора) –* ***отделен гугъл докс за въпроси* резултатите в таб4 (04\_TestQuestions)**

## Използвани ресурси

### 7.1. Ресурси

http://www4u.search.bg/javascript/javascript\_intro.phtml , http://www.w3schools.com/js/default.asp

### 7.2. Ресурси, зададени по условие

гр. София Разработил: . . . . . . . . . . . . . . . . . .

Дата: 13.4.2013 г. /Мартин Маринов/

Приел: . . . . . . . . . . . . . . . . . . /доц. д-р Милен Петров/

**За изтриване:**

**Забележка 1**: Виж инструкции за оценяване, публикувани на адрес:

<http://moodle.le.tsdoit.org/mod/forum/discuss.php?d=58#p139> (wwwTech\_Evaluation\_Instructions\_v3.pdf от 2013-04-01)

**Забележка 2:** За повече информация – на лекции!