



# WEB разработка

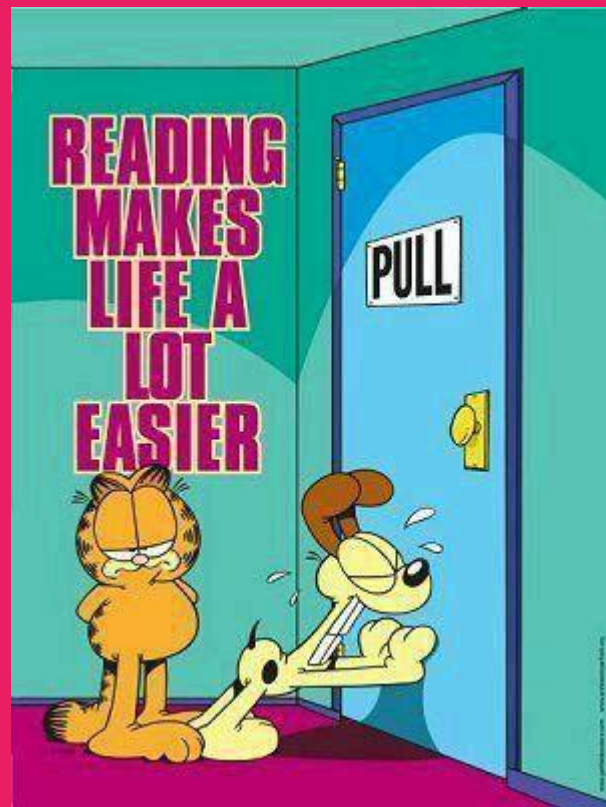
## JavaScript

### ВЪВЕДЕНИЕ

# Съдържание

- Ресурси
- Проекти ниво JavaScript
- JavaScript!?
- Инструменти
- Какво /не/може JS
- JS във времето
- Добавяне на JS към HTML
- JavaScript синтаксис
- Данни и променливи, NaN, undefined, Null
- Hoisting - **demo**
- Условни конструкции

# Ресурси



# Ресурси

[MDN](#)

Murach's JavaScript, 2nd Edition (Training & Reference) - 2015

Телерик, JavaScript курсове, VgCoder

Софтуни, JavaScript, Judge

# Проекты



# Проекти

Самостоятелна задача.

Решена с JavaScript.

Трябва да дава правилен отговор с различни типове данни - разгледани са и са изчерпани всички възможни `if ... else` условия.

**Работа с GitHub** в процеса на работа по проекта.

Don't try to understand  
<sup>JavaScript</sup>  
~~girl~~ completely. If you do,  
then either you'll go mad..  
or you'll start loving <sup>it</sup> ~~her~~.

- Unknown

# JavaScript

JavaScript is a scripting language that is run by the JavaScript engine of a web browser and controls the operation of the browser.

- Когато браузъра изпрати заявка към сървъра за HTML страница, съдържаща линк към JavaScript, и HTML кода и JavaScript се зареждат в браузъра.
- Тъй като JavaScript се изпълнява в средата на **клиента**, не на сървъра, кодът му не изисква “разходка” до сървъра обратно. Приложението работи по-бързо.



# Инструменти

# ИНСТРУМЕНТИ

Може да използвате всяко IDE, или текстов редактор, с който работите.

Browsers` console – Chrome

Sublime

[Конфигуриране на Sublime за работа с JS - плъгини и др. настройки](#)

# ИНСТРУМЕНТИ

[ESLint](#) – инструмент, който ще ви предпазва от грешки при писане на JS

Среди за разработка он-лайн –

[jsbin.com](http://jsbin.com)

[liveweave.com](http://liveweave.com)

[jsfiddle.com](http://jsfiddle.com)

JS performance tester –

[jsperf.com](http://jsperf.com)

Какво /не/може JS

# Какво /не/може JS

- ◆ Управлява събития /events/.
- ◆ Разчита и създава HTML елементи и модифицира DOM-дървото
- ◆ Валидира данни, постъпващи с формата /само на ниво клиент, не е достатъчно за сигурността на уеб-приложението.
- ◆ Достъпва/модифицира кукитата на браузъра.
- ◆ Може да прихване типа на браузъра ви или ОС.
- ◆ Може да изпълнява асинхронни заявки (AJAX)

Най-често JS -

При валидация на формите, слайдове,  
промяна на цвят/картинки ...



*Исключете скрипта на Google и вижте какво се случва ...*

**JS във времето**

# JS във времето

Версия ...

1 June 1997.

2 June 1998.

3 December 1999.

4 Изоставена, не е публикувана никога

5 December 2009.

5.1 June 2011.

6 средата 2015.

7 в процес на разработка ...

# ECMAScript 6

*Все още не се поддържа изцяло във всички браузъри.*

*[ECMAScript 6 спецификация](#)*

*Vanilla JavaScript vs. JavaScript библиотеки ...*



Добавяне на JS към HTML

# Добавяне на JS към HTML

Време е за първия ни JS код:))

.....

```
<body>
```

```
  <script type="text/javascript">
```

```
    alert('Hello JavaScript!');
```

```
  </script>
```

```
</body>
```

.....

# Добавяне на JS към HTML

Като скрипт в head

```
<script type="text/javascript">  
  //js script goes here;  
</script>
```

Като външен/ни файл/фалове в head

```
<script src="path/to/your/js/file/here" type="text/javascript"></script>
```

Като скрипт/ове в body

```
<script type="text/javascript">  
  //js script goes here;  
</script>
```

Като външен/ни файл/фалове преди затварящия body таг.

```
<script src="path/to/your/js/file/here" type="text/javascript"></script>
```

# JavaScript синтаксис

# JavaScript синтаксис

## Основни правила ...

Case-sensitive

Всяка декларация завършва с ;

Игнорира whitespace/tab

За прегледност на кода - може да бъде пренасян,  
при определени условия. /JS ще се опита да добави ; ако прецени,  
че липсват 😊/

# JavaScript синтакс - 2

## Основни правила - имената на променливите

- Съдържат букви, цифри, \_, \$ /не го бъркайте с \$ в PHP/
- **Не могат** да започват с цифра
- case-sensitive.
- Няма ограничение за броя на символите,

но все пак **HE - theManOfTheGameThatPlayedTheDayBeforeYesterday !:))**

- Не може да използвате запазените думи в JS /Виж Murach`s JS за списък на запазените думи
- Добра практика - придържайте се в кода си или само към **camel casing (taxRate)**  
или **underscores (tax\_rate)**
- Ако използвате underscore, използвайте само малки букви.

# JavaScript синтакс - 3

## Основни правила - коментарии

```
/* Block  
   comments */
```

```
//Single-line comments
```

```
//Single-line comments
```

```
//Single-line comments
```

# JavaScript синтаксис - 4

**Обект** в JS - съвкупност от методи и свойства/property.

**Методът** изпълнява функция/действия

**Свойства** - данни, свързани с обекта

**Основни правила - извикване на методи на обекти**  
**objectName.methodName()**

**/Window** е глобален обект за JavaScript, и JavaScript ви позволява да изпускате името му и точката като оператор за извикване на негов метод/

```
window.alert( "This is a test of the alert method " );
```

Можем да извикаме prompt() като изпуснем името на обекта.

```
var userEntry = prompt("This is a test of the prompt method ", 100);
```



# JavaScript синтаксис - 5

Основни правила - свойство на обект

`objectName.propertyName`

Декларация, с която отпечатваме location свойството на window обекта

`alert(window.location);` //отпечатва URL на текущата страница

# JavaScript синтаксис - 5

## Document

Обектът, който ни позволява да работим с т.нар.

**Document Object Model (DOM),**

който представлява **всичките HTML елементи** на една уеб-страница.

var

var

var

Данни и променливи в JS

var

var

var

# Данни и променливи в JS

Деклариране на променлива без да задаваме ѝ стойност.

```
var subtotal; // declares one variable
```

```
var investment, interestRate, years; // declares three variables
```

Деклариране на променлива и задаване на стойност

```
var subtotal = 74.00;
```

```
var salesTax = subtotal *.1; //subtotal = 74.00, salesTax = 7.4
```

След декларирането на променливата, последващото ѝ използване е без var!

Задаване на стойност с израз

```
var subtotal = 74.95; //subtotal = 74.95
```

```
subtotal += 20.00; // subtotal = 94.95
```

# Данни и променливи в JS

## *Типове данни в JS*

Подобно на данните в PHP - не декларираме типа на данните, които ще съхранява променливата. Стойността, която даваме на променливата може да бъде /както и в PHP/

- Цяло число - integer
- Дробно число - float
- String
- Масив
- Boolean
- NULL

Специфични за JS са undefined и NaN

# Данни и променливи в JS - 2

Работим с математически оператори и оператори за сравнение, както и в PHP.  
Употребата на скоби е аналогична на PHP.  
Приоритетът на операторите, също.

Променяме стойността с 1 по три начина

```
var counter = 1; // counter = 1  
counter = counter + 1; // counter now = 2  
counter += 1; // counter now = 3  
counter ++; // counter now = 4
```

Внимавайте с изчисленията с floating-point

```
var subtotal = 74.95; // subtotal = 74.95  
var salesTax = subtotal * .1; //salesTax = 7.4950000000000001
```

# Данни и променливи в JS - 3

**НО**, за разлика от PHP

Знака **+** в изрази, ако

- и двете стойности са числа - JS ще ги събере
- и двете стойности са стрингове - JavaScript ще ги конкатенира
- Едната е число, другата стринг - JavaScript конвертира числото към стринг и конкатенира!!!!

# Данни и променливи в JS - 4

Придаване на стойност на променлива с израз, съдържащ данни от смесен тип

```
var months=120;  
var message = "Months: ";  
message+ = months; // message is "Months: 120"
```

Ескейпване на стринговете - както и в PHP

```
var message = "A valid variable name cannot start with a number.";  
var message = "This isn't the right way to do this.";
```



# Данни и променливи в JS - 4

Деклариране на променлива и придаване на Boolean стойност

```
var isValid= false;
```

# Данни и променливи в JS - NaN

## parseInt() & parseFloat()

parseInt ( string) Конвертира подадения стринг към integer и връща получената стойност. Ако не може да я превърне в цяло число, връща NaN.

parseFloat(string) Конвертира подадения стринг към float и връща получената стойност. Ако не може да я превърне в float, връща NaN.

/NaN е стойност, означаваща "Not a Number". /

Има по-бързи начини да свършим това, което правят parseInt()and parseFloat()

# Данни и променливи в JS - NaN

Превръщане от дробно в цяло число –

```
var PI = Math.PI //3,14.....
```

```
PI = parseInt(PI)//3
```

Или

```
PI = PI | 0
```

# Данни и променливи в JS - typeof()

Как да проверим типа на данните, които държи променливата - **typeof()**

```
var x = 5;  
console.log(typeof(x)); // number  
console.log(x); // 5  
x = new Number(5);  
console.log(typeof(x)); // object  
console.log(x); // Number {}  
x = null;  
console.log(typeof(x)); // object  
x = undefined;  
console.log(typeof(x)); // undefined
```

# Данни и променливи в JS - local/global variables

Локални променливи/Local variables декларираме с ключовата дума **VAR**

Глобални променливи / Global variables - декларираме **без VAR**

**Лоша практика - никога не го правете!**

50

```
var a = 5; // a is local in the current scope  
a = 'alabala'; // the same a is referenced here
```

```
b = undefined; /също недобра практика/  
b = 5; // the same as window.b = 5;
```

# Hoisting demo

# Условни конструкции

# Условни конструкции - if/else

```
if (condition) {
```

```
    Statements1
```

```
    } else {
```

```
    Statements2
```

```
    }
```

```
if (x > 5) {
```

```
    } else if (x > 50) {
```

```
    } else {
```

```
    }
```

*\*Също както и в PHP може да влагаме if/else конструкциите*



# Условни конструкции - switch

[Switch в MDN](#)

DEMO ...