JavaScript

DOM
document oject model

PHP web development 2019/2020

Milena Tomova
Vratsa Software

https://vratsasoftware.com/

Table of Contents

Враца Софтуер Общество

browser objects

browser objects

window

the browser window

document

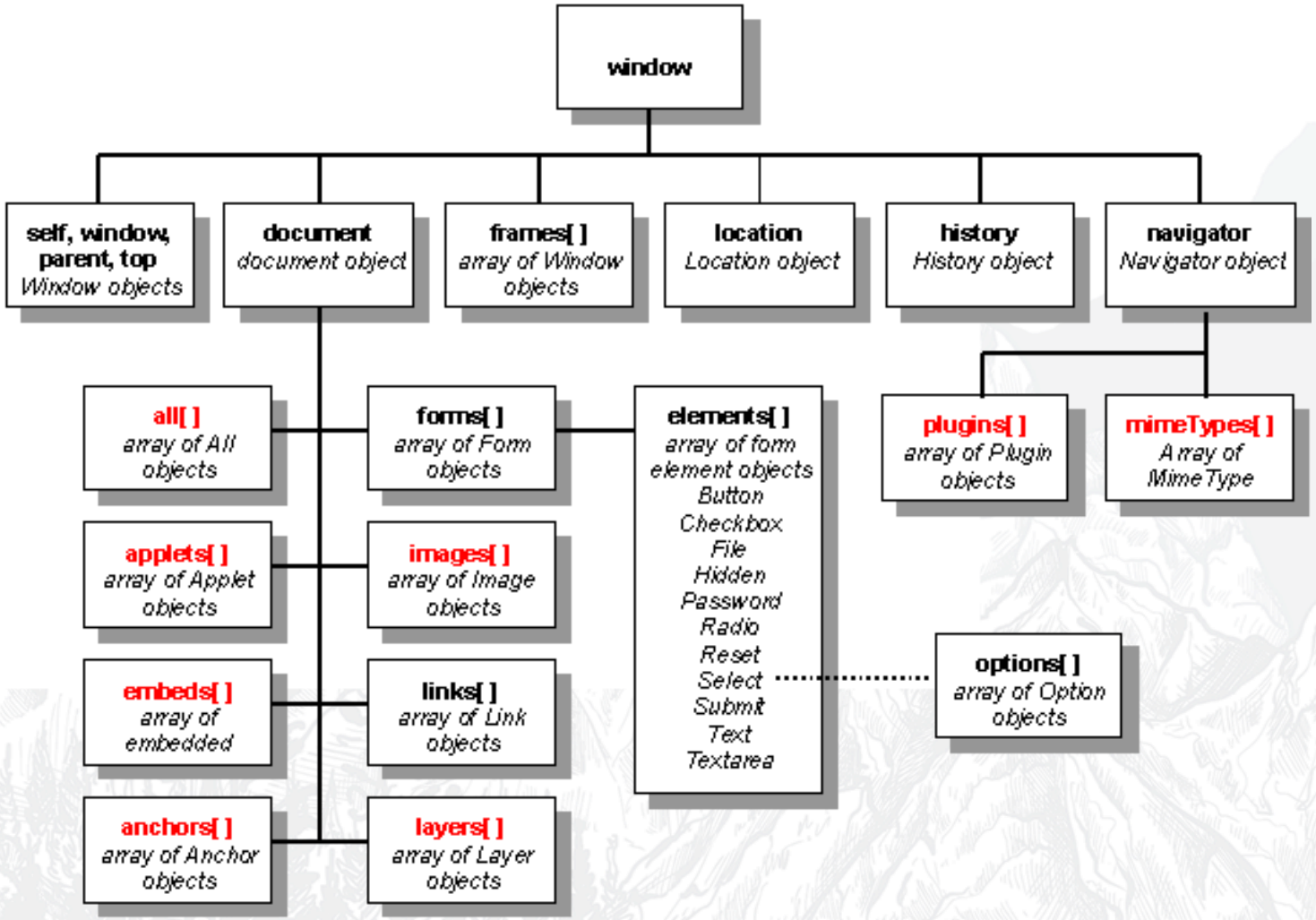
the currently open document in the browser

screen

the visual part of the browser

navigator

info about the currents browser



browser objects

Враца Софтуер Общество

document object methods

```
document.links
document.links[0].href = "yahoo.com";
document.write("This is some <b>bold text</b>");
document.location
```

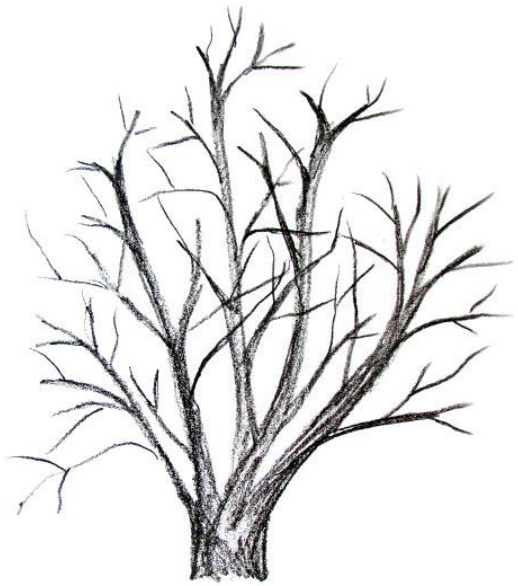
DOM

DOM

Document
Object
Model

When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.

The **HTML DOM**
model is constructed as a tree
of **Objects** -
of all the elements
in the document



DOM

The **HTML DOM** is an **Object Model** for **HTML**. It defines:

- HTML elements as **objects**
- **Properties** for all HTML elements
- **Methods** for all HTML elements
- **Events** for all HTML elements

The **HTML DOM** is an **API** (Programming Interface) for **JavaScript**:

- JavaScript can add/change/remove HTML elements
- JavaScript can add/change/remove HTML attributes
- JavaScript can add/change/remove CSS styles
- JavaScript can react to HTML events
- JavaScript can add/change/remove HTML events

Selecting DOM elements

Selecting ...

Selecting ...

HTML елементите могат да бъдат достъпени и **<u>запазени в променливи</u>** с помощта на DOM API

Selecting ...

target an html element by id or why id is to be unique

```
let email_2 = document.getElementById('email_address2');
                //returns one element
let span = document.querySelector('#email_form span');
                //returns the first element of that selector
```

Selecting ...

Враца Софтуер Общество

targeting a group of html elements

```
let inputs = document.getElementsByTagName('input');

let email = document.getElementsByName('email_address2');

let classGroup = document.getElementsByClassName('className');

let formInputs = document.querySelectorAll('#email_form input');

                                        //all of that selector
```

Selecting ...

Враца Софтуер Общество

targeting with methods for a type of elements

```
let body = document.body;

let links = document.links; //all the links elements in a
                            document

let forms = documents.forms; //all the form elements

let form = documents.forms[2] //third form in the forms
                              collection
```

Selecting ...

selecting nested elements

```
<div id="wrapper">
   <div>Divs in wrapper</div>
   <div>Divs in wrapper</div>
 </div>
```

```
var wrapper = document.getElementById('wrapper');
var divsInWrapper = wrapper.getElementsByTagName('div');
```

Selecting ...

selecting nested elements

```
<div id="wrapper">
   <div>Divs in wrapper</div>
   <div>Divs in wrapper</div>
 </div>
```

```
var divsInWrapper = wrapper.querySelectorAll('.wrapper div');
```

Selecting ...

Return a **single element**

getElementById()

querySelector(‘selector’)

Return a **collection of elements**

getElementsByTagName('tag name');

getElementsByName('name');

getElementsByClassName('classNam e');

querySelectorAll(‘selector’)

Node, Nodes, Nodes List

Node, Nodes, Nodes List

Node,
Nodes,
Nodes List

NodeList - elements/collection returned by the DOM API method -

- getElementsByTagName(tagName)
- getElementsByName(name)
- getElementsByClassName(className)
- querySelectorAll(selector)

Node, Nodes, Nodes List

Node,
Nodes,
Nodes List

NodeList is a JavaScript Object

has
- length property
- index for every node in the list

Node, Nodes, Nodes List

Node,
Nodes,
Nodes List

There are -

live node lists

- returned by the **getElementsBy** methods

static node lists

- returned by the **querySellector** methods

Node, Nodes, Nodes List

Node,
Nodes,
Nodes List

live node list

watches for changes in its nodes and reflects them

static node list

doesn`t change the data for its nodes if they have been changed

How to add JS to your projects?

How to add JS to ….

Враца Софтуер Общество

```
<body>
  <script type="text/javascript">
     alert('Hello JavaScript!');
  </script>
</body>
```

Saving uploaded file

Врача Софтуер Общество

Script in the head element

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

External files linked in the head element

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

Saving uploaded file

Врача Софтуер Общество

Script in the body element

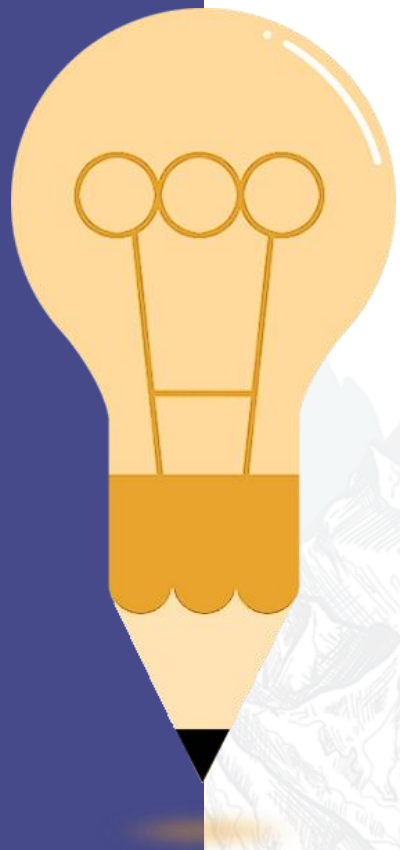
```
<script type="text/javascript">
    //js script goes here;
</script>
```

External files linked before the closing body tag

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

JS syntax

Remember

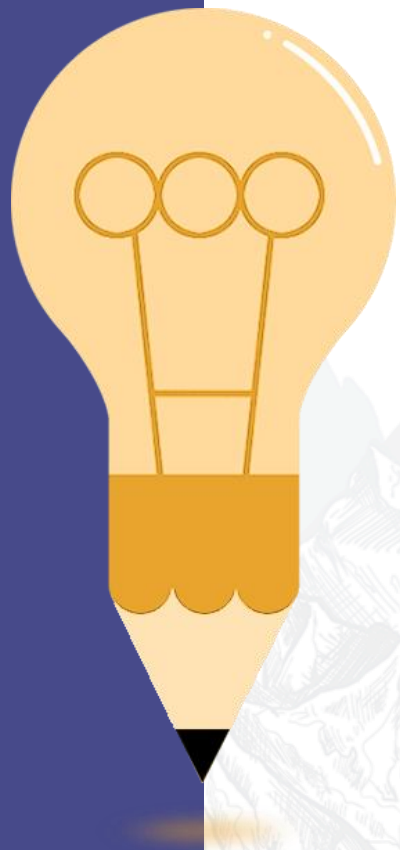


1. Variables and function names are Case-sensitive
2. Every declaration should end with ;
3. Whitespace/tab is ignored
4. JS code can be carried over to a new line for better readability under certain conditions. / JS will try to add ; if he considers that they are missing☺ /

Remember

Враца Софтуер Общество

Basic rules for naming variables



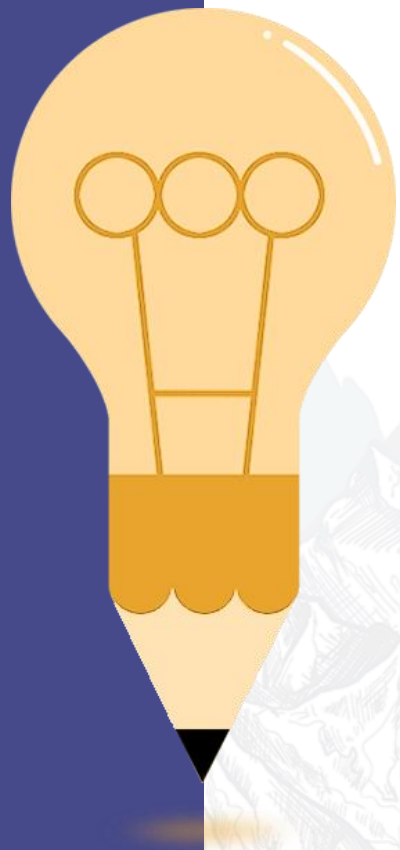
1. Can contain letters, numbers, _, $ / don't confuse it with $ in PHP /
2. Cannot start with a number
3. Variable names are case-sensitive.
4. There is no limit to the number of characters

   but still DO NOT name them this way - theManOfTheGameThatPlayedTheDayBeforeYesterday! :))

1. There are special words in JS that cannot be used as variable names - new, this, etc.

Remember

Basic rules for naming variables

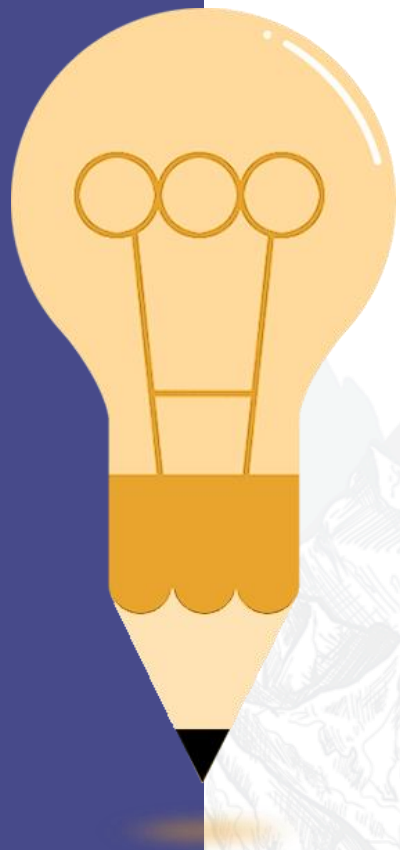


6. More popular for JS is the camel casing (taxRate)
   if you use underscore for word separation -
   be consistent in your code
   do not mixed both

Remember

Враца Софтуер Общество

Basic rules - comments

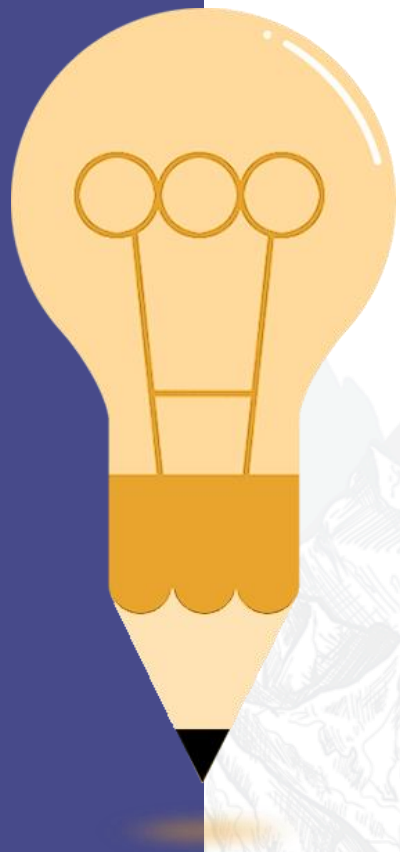


/* Block comments */

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

Remember

Everything in JavaScript is a predefined object



Every string is a JS object
Every number is a JS object
Every array is a JS object
Every object is a JS object

Every JS object comes with a predefined behaviour - a set of methods/functions and properties that can be used to write JS programs.

Remember

Everything in JavaScript is a predefined object

Often will get an error that says *… is not a function …*

If you`ve written the function correctly /no syntax mistake/ that will definitely mean that you are trying to use a function that do not belong to the type of object you have - a function that belongs to a string object and not to a number object for example!

Syntax

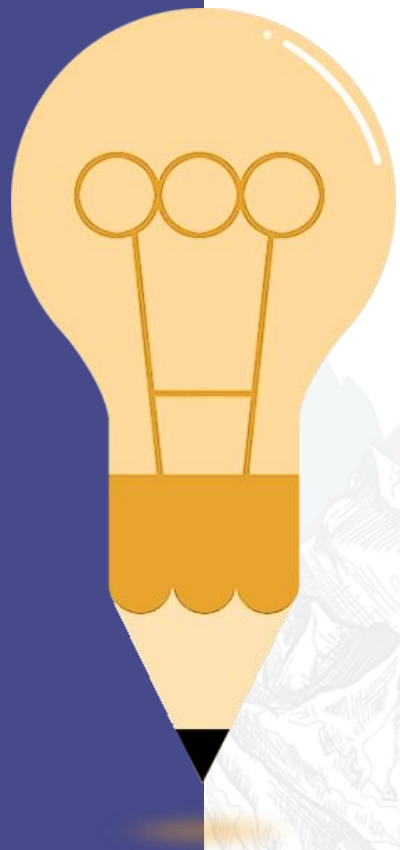
Window is a global JavaScript object, that refers to browser window
alert is window`s method

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

Notice You can omit window, because it is a global object, when calling its alert method

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

Remember



How to call JS object method

window.alert()

How to use JS object property

alert(window.location)

will print the URL of the current open page

location is a window property

JS variables

JS variables

```
var subtotal; // declares one variable
var investment, interestRate, years; // declares three variables
```

Declaring a variable and setting a value

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

After decarning the variable, its subsequent use is without var!

JS variables

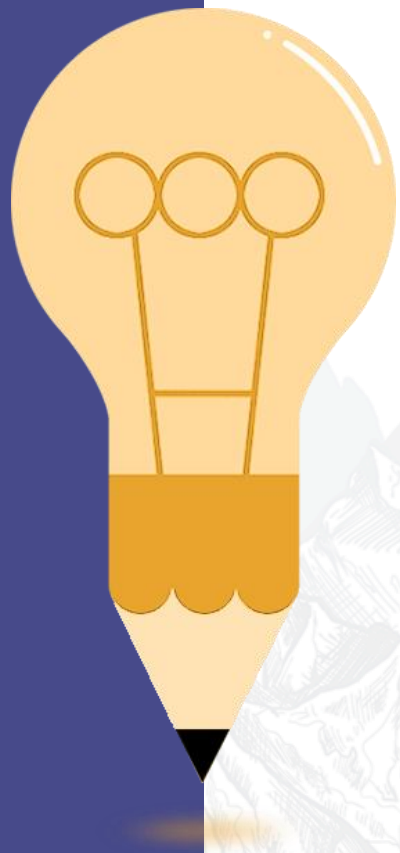
Set a value with an expression

```
var subtotal = 74.95; //subtotal = 74.95
subtotal += 20.00; // subtotal = 94.95
```

JS datatypes

Remember

Datatypes in JS



Like in PHP, we do not explicitly declare the type of data that the variable will store.

The value we give to the variable can be of type

Integer

Fractional number - float

String

Array

Boolean

NULL

Remember

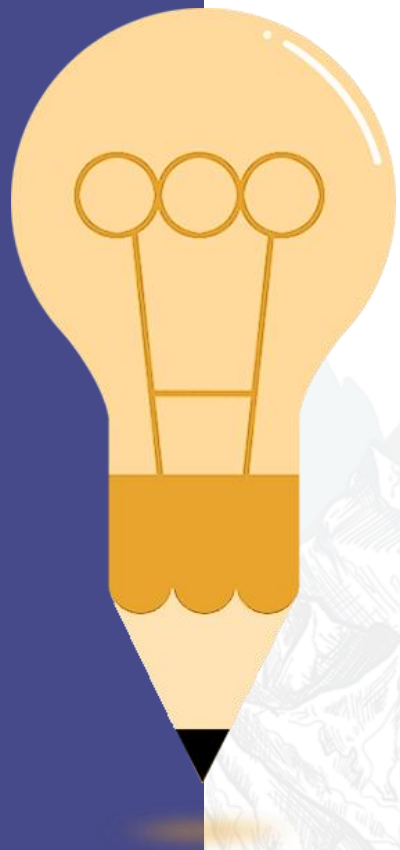
Datatypes in JS

Specific for JS are undefined and NaN

Remember

Операторс Софтуер Общество

Operators in JS



The only difference from PHP is the behaviour

of + operator

JS syntax - operators

Враца Софтуер Общество

```
console.log( 1 + 1 ) //results is 2

console.log( ‘1’ + ‘1’ ) //results is 11

console.log( ‘1’ + 1 ) //results is 11
```

JS syntax - operators

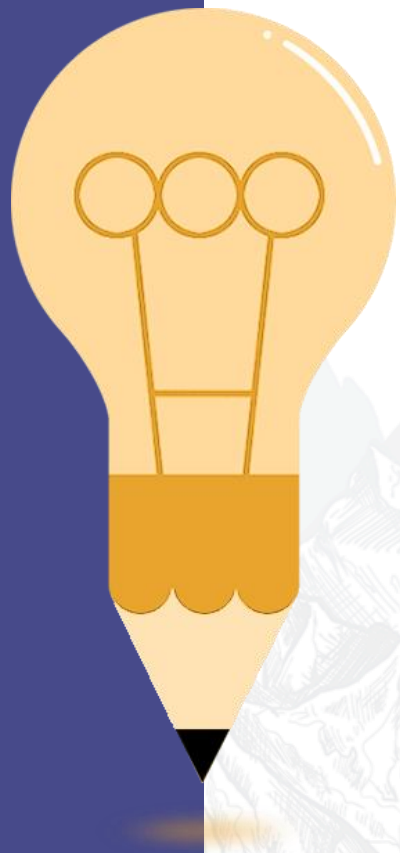
Be careful when using JavaScript for calculations with a floating-point numbers

```
var subtotal = 74.95, salesTax; // subtotal = 74.95
    salesTax = subtotal * . 1; //salesTax = 7.495000000000001
```

Remember

Враца Софтуер Общество

parseInt () & parseFloat ()



parseInt (string)

Converts a string to an integer and returns the resulting value.

If the string is not convertable to a number, returns NaN.

parseFloat (string)

Converts a string to a float and returns the resulting value.

If the string is not convertable to a float, returns NaN.

NaN = not a number

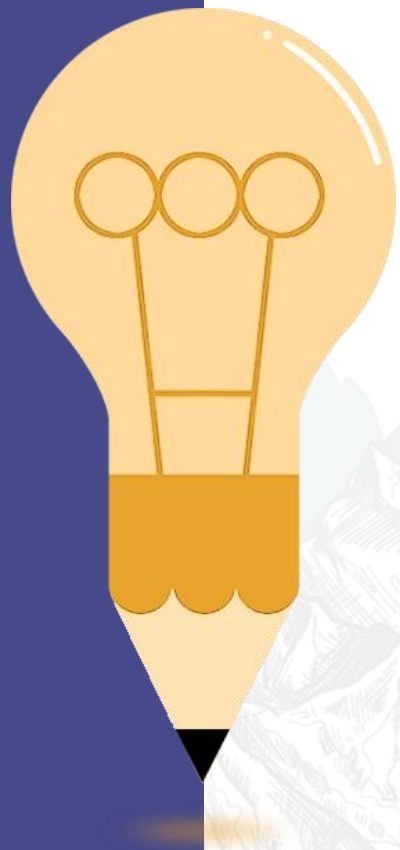
JS syntax - operators

Fastest way to convert a string to a number/to cast to a number is to use +

```
var subtotal = ‘74.95’; // string
    subtotal = +subtotal + 10; //84.95
```

Remember

typeof(var) check the variable type



JS syntax - typeof(var)

Враца Софтуер Общество

```
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
```

hoisting

hoisting

hoisting

Hoisting is JavaScript's default behavior of moving declarations to the top.

JavaScript Declarations are Hoisted

In JavaScript, a variable can be declared after it has been used.

In other words; a variable can be used before it has been declared.

JS syntax - hoisting

Враца Софтуер Общество

```
x = 5; // Assign 5 to x
console.log(x) //5
var x; // Declare x
```

hoisting

hoisting

JavaScript Initializations are Not Hoisted

JavaScript only hoists declarations, not initializations.

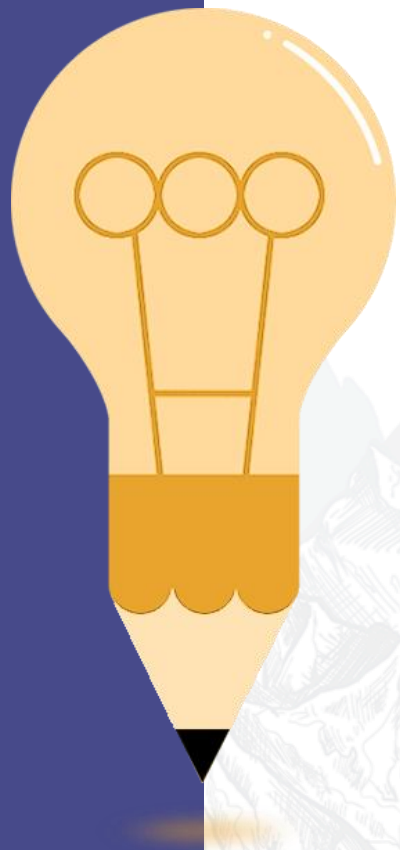
JS syntax - hoisting

Враца Софтуер Общество

```
var x = 5; // Initialize x
var y = 7; // Initialize y
console.log(x);//5
console.log(y);//7
```

```
var x = 5; // Initialize x
console.log(x);//5
console.log(y);//undefined
var y = 7; // Initialize y
```

Remember



Declare Your Variables At the Top !

Hoisting is (to many developers) an unknown or overlooked behavior of JavaScript.

If a developer doesn't understand hoisting, programs may contain bugs (errors).

To avoid bugs, always declare all variables at the beginning of every scope.

Since this is how JavaScript interprets the code, it is always a good rule.

JS syntax - hoisting

Враца Софтуер Общество

```
var x = 5, y = 7; // Initialize x, y
console.log(x);//5
console.log(y);//7
```

```
var x, y; // declare x, y

x = 5; // assign value to x
y = 7; // assign value to x

console.log(x);//5
console.log(y);//7
```

Questions?


Враца Софтуер Общество
Гнездото Coworking
Цялостен курс по програми ране
Дизайн курс
Курс по дигит. маркети нг
MindHub
CodeWeek Враца
HACK VRATSA
#кодаправиш

Partners

Враца Софтуер Общество

AMERICA FOR BULGARIA FOUNDATION

ВРАЦА

Telerik Academy

НАЦИОНАЛНА МРЕЖА ЗА ДЕЦАТА

MindHub

ПРОМЯНАТА

Trainings @ Vratsa Software

Враца Софтуер Общество

- Vratsa Software – High-Quality Education, Profession and Jobs
  - www.vratsasoftware.com
- The Nest Coworking
  - www.nest.bg
- Vratsa Software @ Facebook
  - www.fb.com/VratsaSoftware
- Slack Channel
  - www.vso.slack.com