JavaScript 1.8.5

By Vijay Shivakumar
www.technicaltrainings.com

Before We Begin

Before We Begin

- What we should have
  - An IDE
  - AptanaStudio (recommended)
- What you should know
  - Basics of HTML, CSS, XML, DOM etc…

What we will learn

JavaScript Premier

Objects in JavaScript

Events

Build Web (Browser) based programs

Hands On Experience

About Me

Vijay Shivakumar

Designer | Developer | Trainer

Training on web and Adobe products from past 10+ years

About You ?

- Developer
- Designer
- Architect

History Of JavaScript

- Developed by Netscape
- Client side support for Sun's JAVA
- Concept to Creation in 10 days
- Shipped with Netscape ver. 2.0 (1995)
- Code Name ***MOCHA*** officially called ***LIVESCRIPT***
- Renamed to JavaScript as Microsoft had the patent on the name live.

Brendan Eich

History Of JavaScript

- Microsoft's implementation is called *JScript*
- ECMA Script embraced JavaScript for standardization after 1996
- Officially called as *ECMA Script* but popularly known as *JavaScript*
- Netscape was acquired by AOL in 1999
- AOL and Sun alliance for iPlanet after dissolution iPlanet and JavaScript is retained by Sun.
- Now Oracle owns the name JavaScript and Mozilla owns the source code

Who reads your program ?

Browser's JavaScript engines

- V8 engine in Chrome
- Chakra in IE
- Monkey in Firefox
- SquirrelFish in Safari
- Futhark in Opera

They do

- memory management
- just in time compilation
  (in olden days browsers used to interpret your JavaScript)

What JavaScript is NOT ...!

- JAVASCRIPT is not JAVA
- Can't create or edit files (cookies are an exception)
- Can't be used to talk to databases
- Doesn't need to be compiled
- Can't keep track of user's interaction (stateless)

NOTE : How ever there is a version of JavaScript derived from google's V8 JavaScript engine called node.js and rhino.js from mozilla which can do all of the above...

What is JavaScript ?

- A programming tool for HTML designers / developers
- Read, Modify and Create HTML elements
- React to events like click, swipe, drag, tap etc..
- Validate data
- Detect visitor's browser
- Create and read cookies

Why JavaScript ?

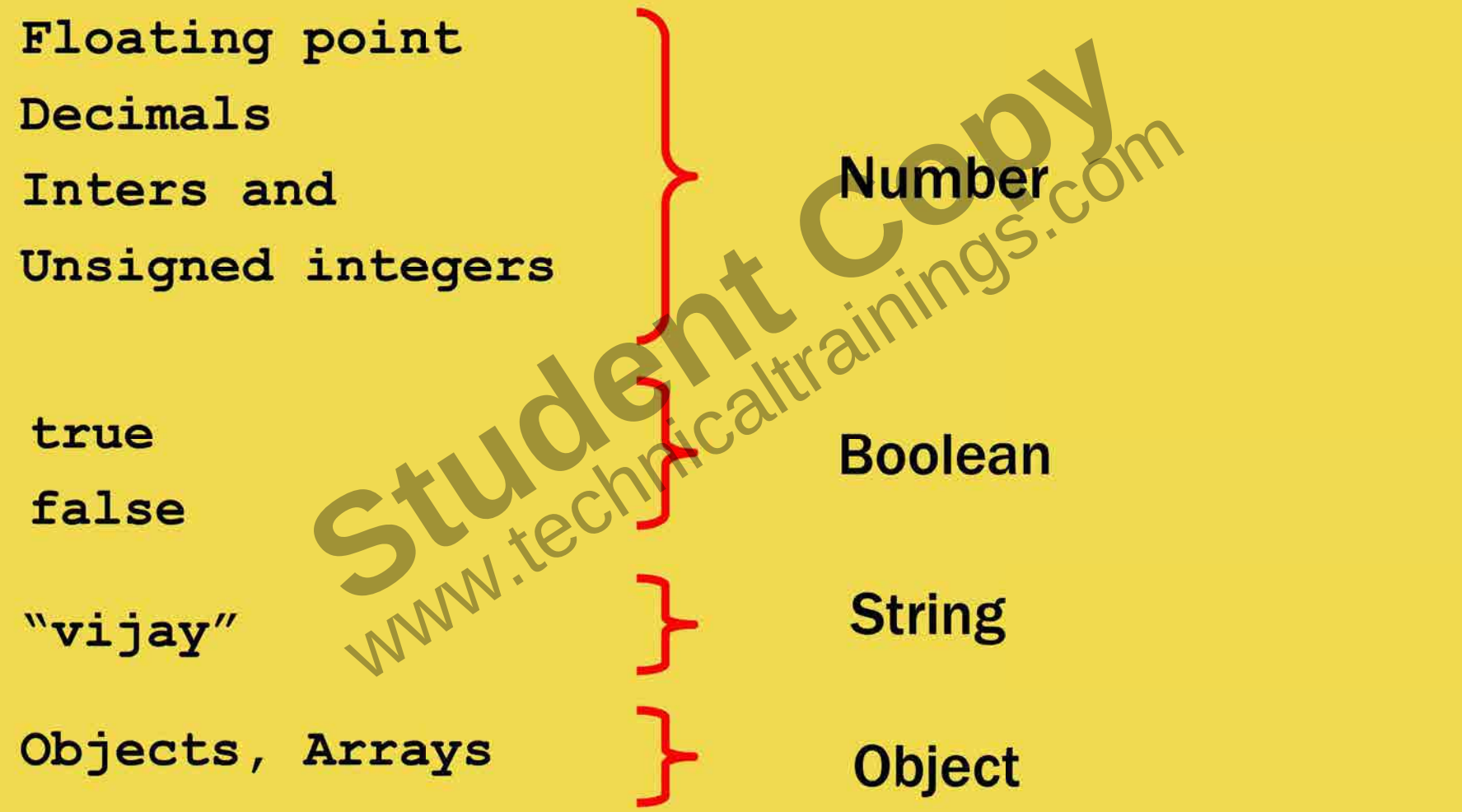
- Most used scripting language
- Great for UI-coding
- Flexible and powerful
- Everything is an object (including functions)
- AJAX makes it a must-know

Bad things about JavaScript

- Global Scope
- + for adding and concatenation
- No need for a semicolon to terminate a line

JavaScript Fundamentals

Types in JavaScript



stores any data type above or arrays & objects

Data Types in JavaScript

Number | 4.5 Any number not inside quote marks

Boolean | true or false A logical operator

String | "Vijay" A series of characters inside quote marks

Object | A virtual thing defined by its properties and methods (in javascript most of them are objects)

Undefined | Returns when a non existent value is called.

undefined when you have not assigned any thing yet

Null | Usually assigned by developers when we initialize a variable but don't want to assign anything yet.

null is assigned by developers as place holders

Where to write JavaScript

Inline JavaScript

```
<a href="javascript:callfun()">click me</a>
<a onclick="callfun()" href="#">click me</a>
```

Infile (Embedded) JavaScript

```
<script type="text/javascript">
   callfun()
</script>
```

External JavaScript (best recommended)

```
yourscript.js
```

(do not use spaces for file names)

Programming in JavaScript

variables

operators

strings

arrays

functions

conditions

loops

OOP in JavaScript

What is Object Oriented Programming ?

A paradigm that uses objects to create your program.

Any thing that is usually self contained and re-usable…

An object has the resources to work on its own to achieve the objective or can inherit properties and methods from other objects.

Why OOP ?

Makes code easy to re-use |No Re- write

Makes code easy to update |Less Bugs

Code easily accessible through APIs|Minimize Mistakes

(Hides what is not required by other objects, Provides access to only what is required)

Objects

Objects contain properties and methods

Objects are made up of key value pairs

Key : value

If more than one property they are separated by comma " , "

Keys can not be reserved key words

eg do, while, class, for etc

If so you can use quotes to overcome them eg., "class"

Values can be of any data type.

If values are functions we call them methods.

OOP Concepts

Creation

- creating Instances a piece of code via classes, functions or duplication

Inheritance

- Extending the behavior of other classes

Encapsulation

- Protect the internal functionalities from being accessed or modified

Polymorphism

- Modify properties and methods of the parent class to achieve a customized performance

Creating Objects

Objects Creation

Objects can be created using

```
var obj = new Object();
var obj = {};
var obj = Object.create(null);
```

Object.defineProperty

```
Object.defineProperty(obj, prop, descriptor)
obj   : The object on which to define the
        property.
prop  : The name of the property to be
        defined or modified.
descriptor : The descriptor for the property
        being defined or modified.
```

Descriptor Object

configurable: true if the descriptor itself can be changed, defaults to false.

enumerable: true if this property shows up only while enumeration defaults to false.

value : The value for the property.

writable : true if the value can be changed. default is false

get : A function which serves as a getter for the property defaults to undefined.

set : A function which serves as a setter for the property, defaults to undefined.

Scope in JavaScript

Closure in JavaScript

Design Pattern

- JavaScript Design Patterns
  - Constructor Pattern
  - Module Pattern
  - Revealing Module Pattern
  - Singleton Pattern
  - Observer Pattern
  - Mediator Pattern
  - Prototype Pattern
  - Command Pattern
  - Facade Pattern
  - Factory Pattern
  - Mixin Pattern
  - Decorator Pattern
  - Flyweight Pattern
- JavaScript MV* Patterns
  - MVC Pattern
  - MVP Pattern
  - MVVM Pattern
- Modern Modular JavaScript Design Patterns
  - AMD
  - CommonJS
  - ES Harmony

vijay.shivu@gmail.com