



Introduction to Javascript

Authored by :Snehal Pawar Presented by : Snehal Pawar

Agenda

- Introduction to Javascript
- Why Javascript???
- History
- Demos for JS
- Variable scopes
- Objects
- Functions
- Object oriented programming



What is Javascript

- Scripting language of browser
- Dynamic
- Object oriented and functional features
- Java syntax
- Small and simple

Why Javascript

- Easy to learn and edit
- Prototyping language
- Easy to debug
- Object oriented design

History

- Developed by Brendan Eich
- Netscape(2.0) 1995
- Microsoft Jscript
- ECMA (**European Computer Manufacturer's Association**)script.

Where to put javascript code

- Embedded in an HTML document between script tags
 <script language="javascript">
 JavaScript statements go here
 </script>
- In an external file which is loaded using
 <script src="program.js" ...></script>



JavaScript file

Writing javascripts

- DOM helps Javascript to interact with html.
- Javascript displays data in different ways
- Using **console.log()**.

Variables scope

- Global Scope
- Block vs Function Scope
- Variable hoisting

Strict mode

- Introduced in ECMA5
- Strict mode applies to *entire scripts* or to *individual functions*.
- Not applied to blocks
- Applied by “use strict”
- Catches coding bloopers and throws exceptions
- Prevents from unsafe actions.

objects

- Objects are collection of properties
- Objects can have single value or multiple values
- `var car="volvo";`
- `var car= {type:" Volvo", model:"100", color: "white"};`
- Objects with name value pair are called properties.

Functions

- First class citizen
- Can assign to a variable
- Can be added as an attribute to an object
- Can be passed as a parameter to a function
- Can be returned from a function.
- Can be passed as an expression

Closures

- JavaScript allows writing nested functions i.e. function within a function
- The inner functions can access variables defined outside the inner function and this concept is called a Closure.
- Closures are functions that refer to independent (free) variables.
- The function defined in the closure 'remembers' the environment in which it was created.
- A closure is a function defined within another scope that has access to all the variables within the outer scope.

Inheritance

- Each object has an internal link to another object called its **prototype**.
- JavaScript follows prototypal inheritance.
- One object can inherit properties of other object
- Every object can then be used as a prototype for another object.

IIFE(Immediately invoked function)

- IIFE is an design pattern
- Anonymous function created and then immediately invoked
- Can be used to avoid **variable_hoisting** from within blocks
- Primary reason to use an IIFE is to obtain data privacy
- Any variables declared within the IIFE cannot be accessed by the outside world.
- Syntax:-

```
(function() {  
    // the code here is executed once in its own scope  
})();
```

Feature Detection

<http://caniuse.com/#>:- use in javascript for feature detection

```
navigator.appName
```

```
"Netscape"
```

```
navigator.appVersion
```

```
"5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/48.0.2564.116 Safari/537.36"
```

```
document.addEventListeneraddEventListener()
```

```
{ [native code] }
```

```
document.attachEvent
```

```
Undefined
```

```
Object.createcreate()
```

```
{ [native code] }
```

What references to use?

- [Professional JavaScript for Web Developers](#) – Nicholas C. Zakas
- [JavaScript: The Good Parts](#) – Douglas Crockford
- [Learning JavaScript Design Patterns](#) - Addy Osmani
- [Mozilla Development Network](#)

A close-up photograph of two people in white business attire shaking hands. The person on the right is wearing a silver metal-link wristwatch. The background is blurred, showing what appears to be an office or industrial setting with orange and grey tones.

Thank You!