

Introduction to JavaScript

IoT & the Raspberry Pi

Ulrich Hauser-Ehninger

Introduction to JavaScript

- Originally a scripting language
- Server side and client side
- Event-Driven, functional and imperative
- ECMAScript standard
- Interpreted language, no compiler
- Created by Brendan Eich at Netscape (1995)
 - Mocha -> LifeScript -> JavaScript
 - Aim: get rid of java applets for dynamic web pages



ECMAScript



- Standard
- Managed by ...
 - European Computer Manufacturers Association (ECMA)
 - Since 1997
- Implementations
 - NodeJS
 - JScript
 - ActionScript
 - JavaScript
 - ...
- Newest version is ECMA Script 8
 - <https://www.ecma-international.org/ecma-262/8.0/index.html>

Applications

- Desktop application
- Website
- Browser Plugin
- Web-Application
- Native Mobile Application



- Server
- Skripting



JS in Web pages

- Embedded or external script
- Script element can be placed anywhere in the document
 - Normal: head or body
 - Best practice: directly in front of `</body>` tag

```
<script>  
    alert('Hello World');  
</script>
```

```
<script src="script.js"></script>
```

Statements

```
alert("Thank you.");
```

Comments

```
// This is a single-line comment.
```

```
/* This is a multi-line comment.
```

```
Anything between these sets of characters will  
be completely ignored when the script is  
executed. This form of comment needs to be  
closed. */
```

Variables

- First symbol needs to be a letter, \$ or _
- No JavaScript keyword as name
- No whitespace
- «JavaScript is case-sensitive»

```
var foo = 5;
```


Primitive data types

Undefined

- Declare a variable
- `var foo;`

Null

- Define a variable, but do not assign an inherent value
- `var foo = null;`

Number

- `var foo = 5;`

String

- `var foo = "5";`

Boolean

- `var foo = true;`

Arrays

```
var foo = [5, "five", "5", '5'];
```

```
alert(foo[0]);
```

```
alert(foo[1]);
```

```
alert(foo[2]);
```

Operators

Operator	Description
==	equal
!=	Not equal
===	Identical (value and type)
!==	Not identical
>	larger
>=	Larger or equal
<	smaller
<=	Smaller or equal

Operators

Operator	Beschreibung
+=	Add value to a variable
++	Increment by 1
--	Decrement by 1

Conditions

```
var test = "testing";  
if( test == "testing" )  
{  
    alert( "You haven't changed anything.");  
} else if( test == "test" )  
{  
    alert( "You've changed variable to test");  
} else  
{  
    alert( "You've changed something!");  
}
```

Loops

```
for( var i = 0; i < 2; i++ )  
{  
    alert(i);  
}
```

Loops

```
var i = 0;  
while (i < 2)  
{  
    alert(i);  
    i++;  
}
```

Loops

```
var i = 0;  
do  
{  
    alert(i);  
    i++;  
} while (i < 2);
```


Scope

«A variable declared with var inside a function is valid inside the function. If declare outside a function it is global.»

```
var foo = "bar"; // global scope
console.log(window.foo); // works
```

```
function doSomething()
{
    var foo = "bar"; // function block scope
}
```

```
function doSomething()
{
    for(var i = 0; i<10;i++)
    {
        // i is visible (function block scope)
    }
    // i is visible (function block scope)
}
```

Scope

«A variable declared with `let` inside a function block is valid inside block.
If declare outside a function it is global.»

```
let foo = "bar"; // global scope
console.log(window.foo); // undefined
```

```
function doSomething()
{
    let foo = "bar"; // function block is enclosing block
}
```

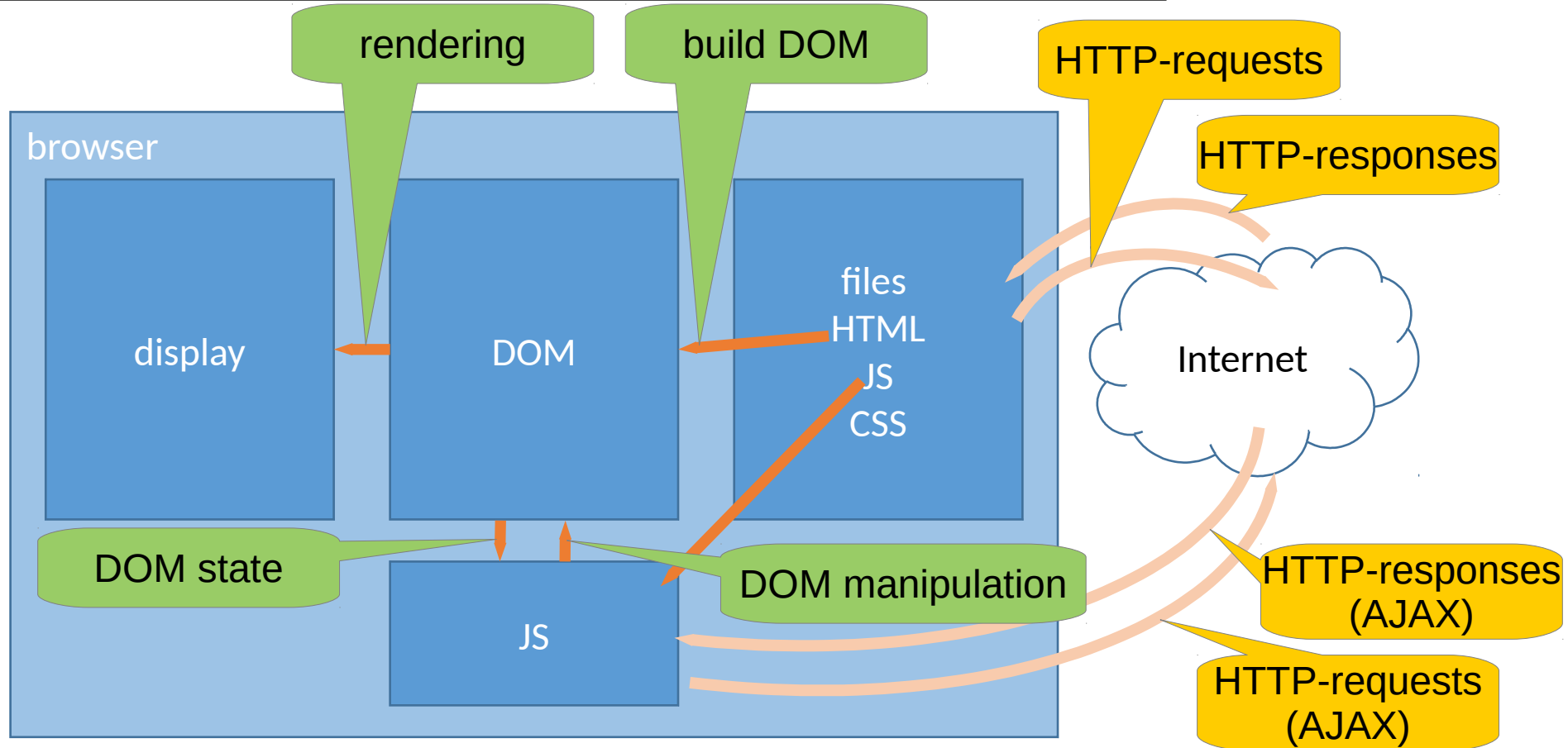
```
function doSomething()
{
    for(let i = 0; i<10;i++)
    {
        // i is visible (enclosing block scope)
    }
    // i is not visible
}
```

Further reading

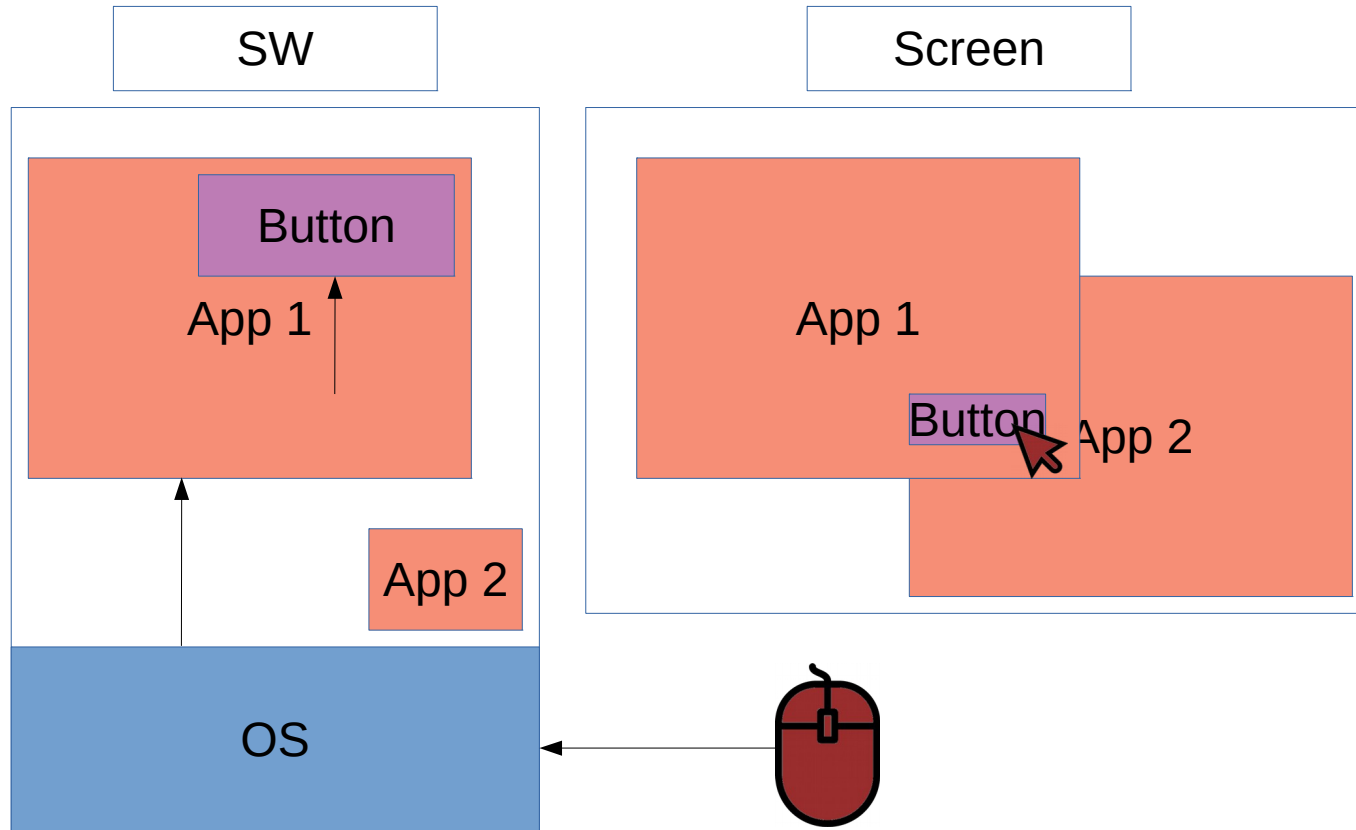
book

- Eloquent JavaScript
 - Introduction
 - Values, Types, and Operators
 - Program Structure
 - Functions

The browser



Hollywood (Event-Driven)



JavaScript-engine in the browser

