

WEBLAB Notizen

Javascript

Jahr	JavaScript	JScript	ECMAScript
1996	1.0	1.0	
	1.1	2.0	
1997 / 1998	1.3	3.0	Edition 1, Edition 2
2000	1.5	5.5	Edition 3
2010	1.8.5	9.0	Edition 5.1
2015			Edition 6
2016			Edition 7
2017			Edition 8

JS

Strict Mode

- `"use strict";` zuoberst im JS-File.
- Verlangt dass man,
 - Variablen deklariert (`var`)
 - Objekte deklariert
 - und vieles mehr ...

Objekte, Object Literals

- Deklaration:

```
var bachelorModule = {  
  title: "WebApplication Development", instructor: "Thomas Koller"  
};
```

- Properties hinzufügen

```
bachelorModule.credits = 3;
```

- Properties löschen

```
delete bachelorModule.credits;
```

- Getters / Setters ES5

```
var otherModule = {  
  course: "WebApp",  
  semester: "F15",  
  get title() {return this.course + this.semester},  
  set title(value) {}  
};
```

- Objekt mit `new` erzeugen

```
var masterModule = new Object();
```

- Objekt mit `create` erzeugen

```
var doctoralModule = Object.create(Object.prototype);
```

- Objekt mit anderem Objekt als Prototype

```
var doctoralModule2 = Object.create(bachelorModule);
```

Arrays

- Sparse Array

```
let a = [];  
a[1000] = "thousand";
```

- Array Reduce Beispiel

```
let sum = a.reduce((x, y) => { return x + y; });
```

Functions

- Function Literal (anonym)

```
let add = (a, b) => { return a + b; };
```

- Function Literal (mit Name)

```
let subtract = function subtract(a, b) { return a - b; };
```

- Function Declaration Statement

```
function mult(a, b) { return a * b; };
```

Hoisting

Variablen haben Block Scope und nach oben auf die ersten Linien in ihrem Scope gehostet.

```
function log () {  
  console.log(a); // funktioniert  
  let a;  
  console.log(b); // funktioniert nicht  
}  
  
function declare () {  
  let b = 5;  
}
```

Immediate Function Invocation

```
let fiveToThePowerOfTwo = function (x) { return x * x; }(5);
```

Constructors

```
function Name(vorname, nachname) {  
  this.vorname = vorname;  
  this.nachname = nachname;  
}  
  
let name = new Name("John", "Doe");
```

Prototype

- (Jedes) Objekt enthält (implizit) ein `prototype` Objekt
- Wird ein Property nicht im Objekt selber gefunden, so wird es (rekursiv) im `prototype` Objekt gesucht
- Das letzte Objekt (i.e. Object) in der Kette hat den Prototyp `null`
- Mit Hilfe der Methode `Object.getPrototypeOf()` (ES5) kann explizit auf den Prototype zugegriffen

werden

- Einem Circle-Prototype eine neue Methode zuweisen

```
function Circle() {  
  this.radius = 1;  
  this.center = {x:0, y:0}  
}  
  
Circle.prototype.draw = function () {  
  console.log("Radius: " + this.radius);  
};  
  
let circle = new Circle();  
c.draw();
```

The Awful Parts of JS

- Global Variables
- Scope
- Semicolon Insertion
- typeof
- Eval
- Falsy values

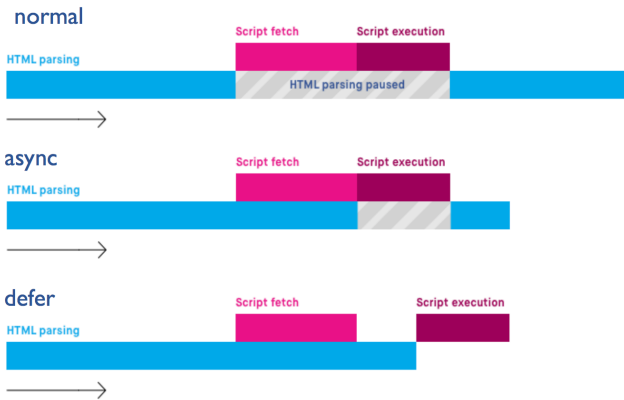
Embedding JS

- Inline `<script></script>`
- External File `<script src="something.js"/>`
- HTML event handler
- URL mit javascript: protocol

Verarbeitung von JS

Phase 1

1. Dokument wird geladen
2. Code `<script>` wird ausgeführt (synchron, asynchron oder deferred)



Phase 2

1. Asynchrone Ausführung durch Event Handler

Event Handlers

```
window.addEventListener("load", function () => { ... }, false);
```

Server Requests

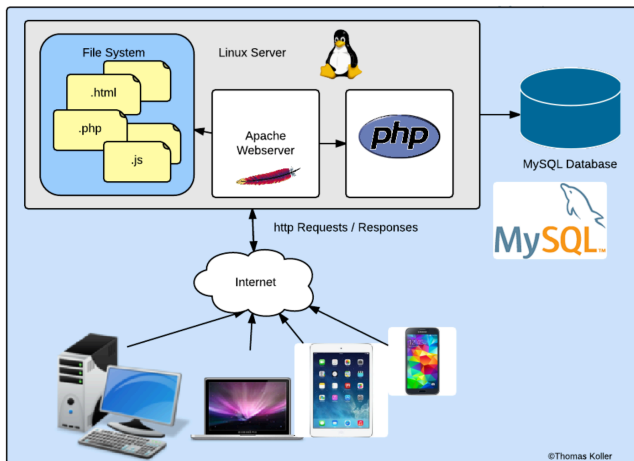
Example

```
var request = new XMLHttpRequest();
request.open("GET", "data.json" /* url */);
request.setRequestHeader("Content-Type", "text/plain");
request.send(null /* body */);
request.onreadystatechange = function() {
    if (request.readyState === 4 && request.status === 200) {
        var type = request.getResponseHeader("Content-Type");
        if (type.match(/^text/)) {
            // do something with request.responseText()
        }
    }
}
```

XMLHttpRequest readyState

Constant	Value	Meaning
UNSENT	0	<code>open()</code> has not been called yet
OPENED	1	<code>open()</code> has been called
HEADERS_RECEIVED	2	Headers have been received
LOADING	3	The response body is being received
DONE	4	The response is complete

LAMP



Wie funktioniert PHP?

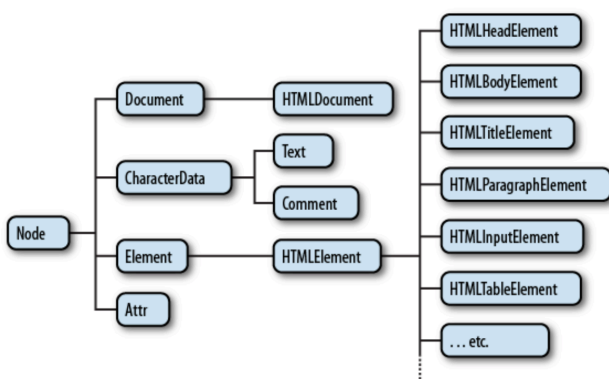
- Apache (Webserver) leitet http-Requests für PHP Files an PHP Interpreter weiter
- Seite wird nach Verarbeitung durch Interpreter wieder an Apache geschickt (als HTML Seite)
- Browser sieht PHP nie!

MySQLi Object Interface

```
$db = new mysqli(host, user, password, databaseName);  
$sql = "SELECT * FROM ... WHERE ..";  
$result = $db->query($sql);  
$row = $result->fetch_assoc();  
echo $row["id"];
```

HTML

DOM-Klassenhierarchie



CSS

Embedding CSS

- Inline Style
- Inline Style Sheets
- External Style Sheets