

Webanwendungen

- Plattform unabhängig
- On the fly & need deployment
- Mehrere Einstiegspunkte: keine fest definierten Einstiegspunkte
- Lose Struktur: Sammlung aus Web-Seiten, Programmen & Skripten

CSS

Inline:

```
<p style="font-style: italic">
Beispiel </p>
```

Im Head:

```
<head>
  <style type="text/css">
    p {font-style: italic}
  </style>
</head>
```

Extern:

```
<style type="text/css"> @import url(...);
... </style>
```

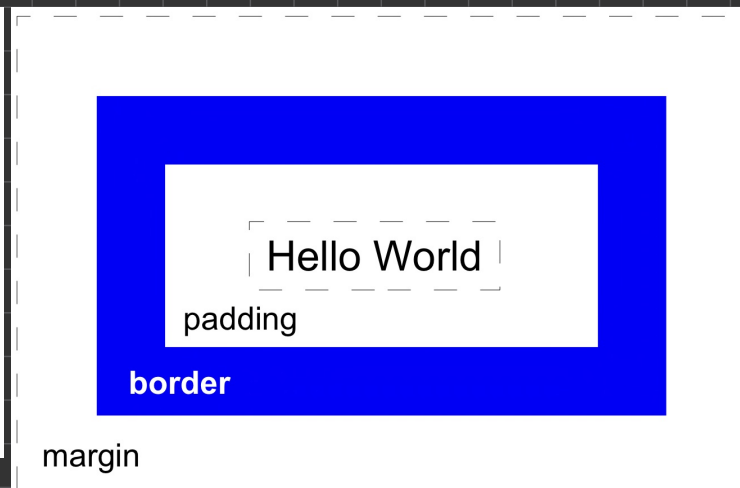
Referenz mittels ID:

```
#ueberschrift1 {text-align: center}
```

Angabe der ID mittels id="..."

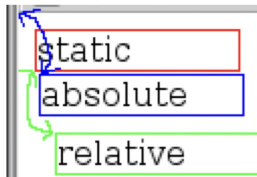
```
<h1 id="ueberschrift1"> ... </h1>
```

- ★ color: Farb-Namen oder Farb-Wert
- ★ background-color: Farb-Name oder Wert
- ★ background-image: url(...)
- ★ border: Dicke in px - Farbe - Linientyp
(solid, dashed, dotted)
Bsp: border: 1px green solid
- ★ padding: Abstand in px
- ★ margin: Abstand in px



Positionierung

```
<body>
<div style="position:static;">static</div>
<div style="position:relative;
left:10px; top:30px;">relative</div>
<div style="position:absolute;
left:10px; top:30px;">absolute</div>
```



CSS kommt nicht in der Klausur dran

JavaScript

```
<!DOCTYPE html>
<html lang="de">
<head>
<meta charset="utf-8"/>
<title>Das erste JavaScript</title>
</head>
<body>
<script>
console.log("Das erste JavaScript");
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="de">
<head>
<meta charset="utf-8"/>
<title>Das erste JavaScript</title>
</head>
<body>
<script src="script.js"></script>
</body>
</html>
```

```
script.js:
console.log("Das erste JavaScript");
```

Variablen

var x = 4

let x = 4

x = 4

(in Codeblock)
(Nur im Kontext)
(global)

```
function varTest() {
  var x = 31;
  if (true) {
    var x = 71; // gleiche Variable!
    console.log(x); // 71
  }
  console.log(x); // 71
}

function letTest() {
  let x = 31;
  if (true) {
    let x = 71; // andere variable
    console.log(x); // 71
  }
  console.log(x); // 31
}
```

Konstruktor:

```
class Animal {
  constructor() {}
}
```

- x
↑
private

Node.js

lokalen Webserver hosten

vorher: `npm install express`

```
const express = require('express');
```

```
let app = express();
```

```
app.get('/', function (req, res) {  
  res.sendFile('tmp/index.html');  
});
```

(<port> , <callback>)

```
app.listen(3000, function () {  
  console.log(  
    'Listening on port 3000!');  
});
```

Callback on success

```
app.post('/', function (req, res, next) {  
  res.set('Content-Type', 'text/html');  
  res.send('<!DOCTYPE html>...' +  
    '<p>Your input was ' +  
    req.body.text +  
    '</p>...');  
});
```

Name of
HTML-element

`npm install mysql`

```
const mysql = require('mysql');
```

Configure database access

```
let con = mysql.createConnection({  
  host: 'localhost',  
  database: 'schema',  
  user: 'username',  
  password: 'password'  
});
```

```
con.connect(function(err) {  
  if (err) throw err;  
  
  // functionality after connecting  
  // to database  
});
```

Throw error if connection failed.

Rest

6 Methoden:

- Get
 - Post
 - Put
 - Delete
 - Head
 - Options
- Wichtigsten

HTTP Methoden	URI	Beschreibung
GET	/book	Bücher auflisten
POST*	/book	Buch aufnehmen
GET	/book/42	Buch holen
PUT	/book/42	Buchdaten ändern
DELETE	/book/42	Buch löschen

Ajax

- Bibliothek: jQuery

Asynchron JavaScript and XML

Qualitätssicherung

Softwarequalität

- Funktionalität
- Zuverlässigkeit
- Benutzbarkeit
- Effizienz
- Änderbarkeit
- Übertragbarkeit

QUnit

```
<title>QUnit</title>
<link rel="stylesheet" href="qunit.css"/>
```

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

```
// Hier werden die Tests spezifiziert
```

Zu testende Klasse
oder Funktionen

```
QUnit.test('Animal', function (assert) {
  let a = new Animal('Hydra', 4);
  assert.equal(a.name, 'Hydra', 'getter name');
});
```

Methoden

`equal(actual, expected, [message])`

`notEqual()`

`deepEqual()`

`strictEqual()`

Auf Exceptions prüfen

```
throws(block [, expected [, message ]])
```

Beispiel

```
QUnit.test('Animal with negative age.',
  function (assert) {
    assert.throws(function () {
      new Animal('Kitty', -5);
    }, Error, 'thrown exception');
  }
);
```

Wenn hier keine Exception
geworfen wird, wird ein
Fehler gemeldet!

Typescript