

```
1<!DOCTYPE html>
2<html lang="de">
3<head>
4  <meta charset="utf-8" />
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>Testseite - Byte GameMaker Magazin</title>
7  <link rel="stylesheet" type="text/css" href="style.css">
8  <script src="formular.js"></script>
9</head>
10<body>
11  <div id="navigation">
12    <a href="//www.bytegame.de/" target="_blank">Link zu Bytegame.de</a>
13  </div>
14  <h1>Willkommen!</h1>
15  <p>Du kannst mal auf den Button klicken!</p>
16  <form>
17    <button type="button" id="hello">Klick nicht!</button>
18  </form>
19  <div id="footer">
20    <p>© 2019 by Byte GameMaker Magazin</p>
21  </div>
22</body>
```

Webentwicklung Grundkurs Teil 2

Description

Die ersten Webseiten des Internets bestanden noch aus statischem HTML. Seitdem hat sich viel getan. Eine moderne Seite enthält eine Vielzahl von Dateien. Design und Inhalt werden strikt voneinander getrennt, die JavaScripte sind ausgelagert und an zahlreichen Stellen werden PHP-Skripte eingebunden. In diesem Teil des Kurses schauen wir uns an, warum das so ist und wie das funktioniert.

Der Grund allen Übels

Nehmen wir an, wir hätten eine Homepage mit nur drei Seiten. Jede Seite unterteilt sich in drei Bereiche. In der Kopfzeile findet sich das Logo und die Navigation, in der Mitte der eigentliche Inhalt und in der Fußzeile der Copyright-Hinweis und vielleicht weitere Links.

Wenn wir kurz darüber nachdenken, fällt uns gleich ein Problem auf: Kopf- und Fußbereich sind immer gleich. Bei einer statischen Seite müssen wir diese Bereiche kopieren, was noch kein Problem darstellt. Die Probleme beginnen, wenn wir Änderungen vornehmen und wachsen mit der Anzahl der Seiten. Jede neue Seite ein neuer Menüeintrag. Nach dem Jahreswechsel dürfen wir immer die Fußzeile jeder Seite anpassen. Und möchten wir das Design ändern, können wir gleich alles neu machen.

Ein moderner Aufbau

Eine halbwegs moderne Seite besteht heute aus mehreren Dateien, selbst wenn es sich nur um eine Seite handelt. Die Anzahl der Dateien (und hier rechne ich Grafiken nicht mit) beträgt fünf oder mehr Dateien. Unterteilt werden sie in:

- HTML Bereich
- PHP Dateien
- JS Dateien
- CSS Dateien

Da auch der HTML-Bereich mit PHP durchsetzt ist, haben auch diese Dateien oft die Endung .php. Wir halten es einfach und unterteilen unsere Beispielseite wie folgt:

- header.php
- navigation.php
- footer.php
- index.php
- style.css
- formular.js

Die index.php ist die erste Datei, die beim Aufruf einer URL gestartet wird. Sie kann übrigens auch index.htm oder index.html heißen. Von den anderen Dateien bekommt der Seitenbesucher nichts mit. [Hier ein Beispiel.](#)

Zugegeben, die Seite ist hässlich wie die Nacht. Aber dafür zeigt sehr anschaulich den Aufbau. Wenn man den Code kennt.

Die ganze Seite im Überblick

Was ich jetzt zeige, ist ein Mittelweg aus dem, was ich oben erklärt habe. Diesen Zwischenschritt mache ich, damit man das Prinzip besser verstehen kann:

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Testseite - Byte GameMaker Magazin</title>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script src="formular.js"></script>
  </head>
  <body>
    <div id="navigation">
      <a href="//www.bytegame.de/" target="_blank">Link zu Bytegame.de</a>
    </div>
    <h1>Willkommen!</h1>
    <p>Du kannst mal auf den Button klicken!</p>
    <form>
      <button type="button" id="hello">Klick mich!</button>
    </form>
    <footer>
      <br>
      <br>
      <p id="footer">© 2019 by Byte GameMaker Magazin</p>
```

```
</footer>
</body>
</html>
```

Die Datei heißt zwar index.php, man könnte sie aber auch index.html nennen, da sich darin kein php-Code befindet.

Grob lässt sich der Code in vier Bereiche unterteilen:

<!DOCTYPE html> sagt dem Browser lediglich nur, dass es ein HTML-Dokument ist.

<html lang="de"> ist ein HTML-Block mit der Information, dass die Seite auf Deutsch ist.

<head> ist der Kopfbereich. Hier geht es um Informationen für den Browser, für Suchmaschinen und wir laden hier die **style.css** und **formular.js**. Das heißt, wir haben JavaScript und Style ausgelagert.

<body> ist der eigentliche Inhalt der Seite. Darin befindet sich auch die provisorische Navigation, das „Formular“ (ist ja nur ein Button) und der Footer.

Am Ende werden die Blöcke abgeschlossen. Auf die einzelnen Tags und wie das mit dem Style und dem JavaScript geht, möchte ich in diesem Tutorial noch nicht ausführen. Es geht schließlich nur um den groben Aufbau. Ich liefere jetzt noch den Code der beiden Dateien nach, damit ihr selbst damit experimentieren könnt.

style.css

```
body{
    margin: 2em;
    background-color: lightgreen;
    font-family: „Lucida Sans Unicode“, „Lucida Grande“, sans-serif;
}

h1 {
    display: block;
    font-size: 20px;
    letter-spacing: 1px;
    word-spacing: 0.4px;
    color: #000000;
    font-weight: 700;
    text-decoration: none;
    font-style: normal;
    font-variant: normal;
    text-transform: none;
}

p {
    font-size: 14px;
    letter-spacing: 0px;
    word-spacing: 0px;
    color: #000000;
    font-weight: 400;
    text-decoration: none;
}
```

```
font-style: normal;
font-variant: normal;
text-transform: none;
}

#navigation {
background-color: lightblue;
font-size: 16px;
letter-spacing: 0px;
word-spacing: 0px;
font-weight: bold;
color: #000000;
}

#footer {
background-color: lightblue;
font-size: 14px;
text-align: center;
letter-spacing: 0px;
word-spacing: 0px;
font-weight: 100;
color: #000000;
}
```

Ohne ins Detail gehen zu wollen, sieht man hier eine wichtige Sache: Der Style ist in verschiedene Blöcke unterteilt, den wir im HTML beliebig aufrufen können. Kurz: Wir definieren das Zeug nur EINMAL und können es beliebig oft verwenden. Letztlich geht es bei dem ganzen Zirkus nur darum.

formular.js

Ich weiß, es ist nicht gerade spektakulär. Aber es geht letztlich nur um das P

Aufteilung der HTML-Datei

Wir teilen nun alles wie oben beschrieben auf. Header, Navigation, Footer. Was **index.php** mit dem Inhalt.

header.php

[Wer aufmerksam hingeschaut hat, stellt fest, dass das JavaScript fehlt.](#)

footer.php

Und da ist das Skript auch schon. In diesem Beispiel musste es verschoben werden

```
<div id="navigation">  
<a href="//www.bytegame.de/" target="_blank">Link zu Bytegame.de</a>  
</div>
```

Hier kommt zurecht die Frage, warum man die Navigation auslagert, statt sie in

index.php

Tadaaaa! Der ganze Firlefanzen wird mit drei Zeilen eingeladen und dazwischen be

Ausblick

Im nächsten Teil haben wir noch einen trockenen Part. Wir schauen uns an, wie

An die Profis

Mir ist klar, dass in diesem Tutorial viele Themen, wenn überhaupt, nur gestre

Überblick Webdev-Serie

Webentwicklung Grundkurs Teil 1 – Einstieg

Webentwicklung Grundkurs Teil 2 – Aufbau von Webseiten

Webentwicklung Grundkurs Teil 3 – Datei- und Ordnerstrukturen

Webentwicklung Grundkurs Teil 4 – Einstieg in JavaScript

[Webentwicklung Grundkurs Teil 5 – Datenverarbeitung und Formulare mit JavaScript](#)

Überblick Interviews

[Interview mit Magnus Reiß – Webgamers](#)

[Interview mit Wolfgang Scheidle – Tischtennis Manager](#)

[Interview mit Warg – Drifting Souls II](#)

Date Created

15. November 2019

Author

sven