

HTML5-Tutorium: JavaScript: Hello World 02

Wechseln zu: [Navigation](#), [Suche](#)

Dieser Artikel erfüllt die [GlossarWiki-Qualitätsanforderungen](#) **nur teilweise**:

Korrektheit: 3 (zu größeren Teilen überprüft)	Umfang: 4 (unwichtige Fakten fehlen)	Quellenangaben : 3 (wichtige Quellen vorhanden)	Quellenarten: 5 (ausgezeichnet)	Konformität: 3 (gut)
---	--	--	---	--------------------------------

Vorlesung WebProg

[Inhalt](#) | [Teil 1](#) | [Teil 2](#) | [Teil 3](#) | [Teil 4](#) | [Teil 5](#) | [Teil 6](#)

Musterlösung: [index.html](#) ([Git-Repository](#))

Inhaltsverzeichnis

- [1 Anwendungsfälle \(Use Cases\)](#)
- [2 Erstellen eines neuen Projektes](#)
- [3 Festlegen des Designs der Anwendung](#)
- [4 Optimierung des Designs für mobile Endgeräte](#)
- [5 Aufräumen des öffentlichen Web-Servers](#)
- [6 Fortsetzung des Tutoriums](#)
- [7 Quellen](#)

1 Anwendungsfälle (Use Cases)

Die Anwendungsfälle ändern sich gegenüber dem ersten Teil des Tutoriums nicht: Der einzige Anwendungsfall der Anwendung ist weiterhin die Begrüßung der ganzen Welt, sobald die Anwendung ausgeführt wird.

Allerdings soll die Begrüßung gefälliger gestaltet werden, indem das Layout der Begrüßungsseite verbessert wird. Zeichensatz, Schriftgröße, Zeilenabstand etc. werden geändert.

2 Erstellen eines neuen Projektes

Erstellen Sie im selben Ordner, in dem sich das Projekt `HelloWorld01` befindet, das Projekt `HelloWorld02` als (eigenständigen) [Fork](#) des Projektes `HelloWorld01`:

```
cd .. # Eltern-Ordner von HelloWorld01
git clone https://gitlab.multimedia.hs-augsburg.de/ACCOUNT/HelloWorld01
HelloWorld02

# Altes Projekt klonen

## Änderungen zur Projektidentifikation vornehmen (z.B. Titel anpassen)
git commit -m "HelloWorld01 fork created"

# Neues Repository in Gitlab anlegen:
git remote -v
git remote remove origin
git remote add origin https://gitlab.multimedia.hs-
augsburg.de/ACCOUNT/HelloWorld02
git remote -v
git push --set-upstream origin master
```

Überprüfen Sie, ob die Anwendung korrekt gestartet werden kann.

Falls dies der Fall ist, committen Sie. Anschließend können Sie das Projekt mittels eines Git-push-Befehls erneut auf den Gitlab-Server kopieren.

3 Festlegen des Designs der Anwendung

Nun wird das Design der Anwendung verbessert. Die Überschrift soll in einem serifenlosen Font dargestellt werden, der Text des nachfolgenden Absatzes in einem serifenhaltigen.

Die ist kein **CSS**-Tutorium, daher wird nicht darauf eingegangen, wie ein geeigneter Font gewählt werden sollte. Wir überlassen das einfach dem jeweiligen Browser-Hersteller, indem wir die Pseudofonts „**sans-serif**“ und „**serif**“ angeben. Für **Mac**-User wird die Überschrift vermutlich in **Helvetica** gesetzt und für **Windows**-Benutzer in **Arial**.

Erstellen Sie eine **CSS**-Datei:

Klick auf das Dokument-Plus-Icon neben „**HELLOWORD02**“ des Dateibaums
Angabe des Namens „**main.css**“

Es wird eine neue leere Datei namens „**main.css**“ angelegt.

Fügen Sie in diese Datei folgenden **CSS**-Code ein:

```
h1
{
  font-family: sans-serif;
}

p
{
  font-family: serif;
}
```

Achten Sie darauf, dass öffnende und zugehörige schließende Klammern stets übereinander oder in derselben Zeile stehen ([Allman-](#) oder [BSD-Stil](#)).

Wenn Sie andere Tutorien bearbeiten, Fachbücher lesen oder Voreinstellungen von Web-IDEs begutachten, werden Sie meist folgende Art der Formatierung entdecken („[OTBS](#)“, „[One True Brace Style](#)“, „[Einzig wahrer Klammerstil](#)“). Dieser Klammerstil ist sehr weit verbreitet, verstößt aber gegen die Regel, dass öffnende und zugehörige schließende Klammern stets übereinander oder in derselben Zeile stehen.

```
h1 {
  font-family: sans-serif;
}

p {
  font-family: serif;
}
```

Die Begründung lautet dann oft, dass dieser Code weniger Zeilen habe, als der obige Code. Das stimmt, lässt sich aber auch ohne Verletzung der Regel „Zusammengehörige Klammern befinden sich in derselben Spalte“ erreichen ([Horstmann-Stil](#)):

```
h1
{ font-family: sans-serif;
}

p
{ font-family: serif;
}
```

Anmerkung: Ein schöner Artikel von Cay Horstmann selbst zu diesem Thema: [The Horstmann Brace Style](#)

Im Falle von CSS-Code macht es keinen wirklichen Unterschied, welcher Konvention sie folgen. **Im Falle von JavaScript rate ich jedoch davon ab, öffnende und schließende Klammern nicht in derselben Spalte zu notieren.** Glauben Sie einem erfahrenen [LISP](#)-Programmierer (LISP ist die Sprache, bei der *jeder* nicht-atomare Programmcode durch Klammern umschlossen wird): Code ist

wesentlich besser zu lesen, wenn öffnende und schließende Klammern übereinander stehen. Insbesondere Strukturfehler sind bei dieser Konvention wesentlich einfacher zu finden. Ob Sie die öffnende Klammer jeweils in einer eigenen Zeile schreiben oder nicht, ist dabei weniger wichtig.

Unabhängig davon, welche Klammernotation Sie bevorzugen, einigen Sie sich im Team bitte auf EINE Variante. Halten Sie diese Konvention strikt ein.

Glücklicherweise highlighten die meisten Editoren (wie beispielsweise der VSC-Editor) die zugehörige schließende oder öffnende Klammer, wenn man den Cursor auf (bzw. vor oder nach) eine öffnende oder schließende setzt. So erkennt man Strukturfehler auch dann, wenn man bei der Einrückung Fehler gemacht hat. Derartige Fehler sollte man allerdings möglichst schnell beheben, wenn man sie einmal entdeckt hat. Sonst entsteht unlesbarer Code.

Wenn Sie Ihre CSS-Datei fertiggestellt haben und die Anwendung erneut testen (d. h. die Datei „`index.html`“ im Browser vom VSC-Server laden), werden Sie feststellen, dass sich nichts geändert hat (auch wenn Sie den Reload-Button des Browsers drücken).

Sie können die CSS-Datei direkt im Browser laden (ersetzen sie in dem Browser-URI einfach „`index.html`“ durch „`main.css`“ und drücken Sie den Reload-Button), aber das hilft auch nichts. Ihnen wird daraufhin einfach der Inhalt des CSS-Datei vom Browser angezeigt.

Der Grund ist, dass der Browser beim Laden der HTML-Datei nicht weiß, dass er den Inhalt mit Hilfe der CSS-Datei layouten soll. Damit er dies weiß, muss folgende Zeile in den `head`-Bereich der HTML-Datei eingefügt werden:

```
<link rel="stylesheet" href="main.css"/>
```

Wenn Sie jetzt noch einmal Ihre Web-Anwendung testen, wird die Überschrift (evtl. nach einem Klick auf den Reload-Button des Browsers) tatsächlich in einer serifenlosen Schriftart dargestellt.

Sie können jetzt noch weitere Verbesserungen vornehmen: Anpassung der Schriftgröße, Anpassung des Abstandes zwischen Überschrift und Absatz, Anpassung der Schriftfarbe, des Seitenhintergrundes etc. pp.

Zum Beispiel:

```

html
{ height: 100%;
  width: 100%;
  display: table;
}

body
{ display: table-cell;
  text-align: center;
  vertical-align: middle;
  background-color: #C5EFFC;
  font-size: 3vw !important;
  font-family: Verdana, Helvetica, sans-serif;
}

h1
{ padding: 0;
  margin: 0;
  font-size: 305.18%; /* 1.25^5 = 3.0518 */
}

p
{ font-family: "Times New Roman", Times, serif;
  font-size: 100%;
}

```

Spielen Sie mit den CSS-Attributen und testen Sie Ihre Anwendung.

Tipp: Mit der CSS-Regel

```
* { border: solid 1px red; }
```

können Sie sehr einfach den Effekt von Padding-Attribut-Werten überprüfen. Falls Sie Ränder um einige Ihrer Elemente zeichnen, sollten Sie

```
* { outline: solid 1px red; }
```

verwenden. Wenn Sie mit der Überprüfung fertig sind, sollten Sie diese Befehle allerdings wieder aus der CSS-Datei entfernen oder sie zumindest auskommentieren (`/* ... */`).

Wenn Sie schließlich mit dem Layout Ihrer Anwendung zufrieden sind, committen Sie sie (Commit Message: „Styling has been added.“). Sorgen Sie mittels eines Git-push-Befehls dafür, dass die aktuelle Version Ihrer Anwendung vorsorglich auf den Git-Server kopiert wird.

4 Optimierung des Designs für mobile Endgeräte

Öffnen Sie die Web-Anwendung mit einem Smartphone. Leider geht das derzeit noch nicht. Da der Zugriff auf den VSC-Servers auf Ihrem Entwicklungsrechner eingeschränkt ist, müssen Sie Ihre Anwendung auf einen echten Web-Server legen, um mit einem anderen Device darauf zugreifen zu können. Als Mitglied der Hochschule Augsburg können Sie dazu Ihre [HSA-Homepage](#) verwenden.

Damit nicht alle Projektdaten auf den öffentlichen Web-Server übertragen werden, ist es sinnvoll in VSC einen Ordner `web` anzulegen und in diesem alle Dateien des Web-Auftritts zu verschieben, die öffentlich zugänglich sein sollen:

Einen neuen Ordner `web` anlegen

`index.html` und `main.css` selektieren und in diesen Ordner verschieben.

Nun können Sie den Web-Ordner in Ihr HSA-Web-Verzeichnis kopieren (Achtung: Im Folgenden müssen Sie jeweils **ACCOUNT** durch Ihren RZ-Account ersetzen).

Öffnen Sie das Terminal in WebStorm.

`scp -r web ACCOUNT@login.rz.hs-augsburg.de://www/ACCOUNT/hello_world_02` (Der Befehl `scp` hat dieselbe Aufgabe wie `ftp`, überträgt die Daten aber verschlüsselt:

<https://www.computerhope.com/unix/scp.htm>)

Mit diesem Befehl haben Sie die Dateien `index.html` und `main.css` in den Ordner `HelloWorld02` Ihres Web-Verzeichnisses kopiert.

Loggen Sie sich auf dem HSA-Server ein, und sehen Sie sich das Ergebnis an:

```
ssh -lACCOUNT login.rz.hs-augsburg.de
cd /www/ACCOUNT # Wechseln Sie uns HSA-Web-Rootverzeichnis
ls -al
ls -al *
exit # Oder CTRL+D
```

Mit dem ersten Befehl haben Sie sich auf dem Benutzerserver der Hochschule Augsburg eingeloggt. Ab sofort werden alle Befehle, die Sie ins Terminal tippen, direkt von der Unix-Bash des Hochschulrechners ausgeführt. Mit dem Befehl `exit` kehren Sie wieder zu Ihrer lokalen Konsole zurück.

Öffnen Sie nun eine der folgenden URLs in einem Browser Ihrer Wahl:

https://www.hs-augsburg.de/homes/ACCOUNT/hello_world_02/

https://www.hs-augsburg.de/homes/ACCOUNT/hello_world_02/index.html

Dann sollten Ihnen Ihre zuvor erstellte HTML-Seite präsentiert werden.

Wenn Sie nun eine dieser URLs im Smartphone öffnen, werden Sie feststellen, dass die Schrift viel zu klein ist und der Begrüßungstext daher nur schwer gelesen werden kann. Der Grund ist, dass die mobilen Browser in der Regel genauso rendern wie stationäre Browser. Da der Bildschirm aber viel kleiner ist, erkennt man nicht mehr viel.

Unter **Responsive Webdesign** versteht man ein gestalterisches Paradigma, welches zum Ziel hat, eine Web-Site auf jedem möglichen Endgerät möglichst optimal les- und benutzbar darzustellen.

Technische Basis ist hier vor allem [CSS](#), aber natürlich auch [HTML5](#) und [JavaScript](#).

In unserem Fall reicht es, ein weiteres Meta-Tag in den HTML-Header einzufügen, das den Browser anweist, für die Darstellung der HTML-Body-Elemente eine Bühne zu verwenden, die genauso breit ist, wie das Fenster des Devices (PC, Tablet im Querformat, Smartphone im Hochformat etc.) .

```
<meta name    = "viewport"  
      content = "width=device-width, initial-scale=1.0, user-  
scalable=yes"  
>
```

Darüber hinaus wird dem Browser mitgeteilt, das der Benutzer die Größe ändern kann (per Touchgeste). Initial beträgt die Größe 100% (= 1,0).

Testen Sie die Anwendung erneut:

```
scp -r web/* ACCOUNT@login.rz.hs-augsburg.de://www/ACCOUNT/hello_world_02
```

Achtung: Da auf dem HSA-Server der Ordner `/www/ACCOUNT/hello_world_02` bereits existiert, müssen Sie diesmal (und auch zukünftig) beim SCP-Befehl `web/*` an Stelle von `web` schreiben. Damit kopieren Sie den **Inhalt des Ordners** `web` in den Ordner `/www/ACCOUNT/hello_world_02`. Beim ersten Mal hatten Sie den gesamten Ordner `web` (samt Inhalt) auf den HSA-Server kopiert und dort unter dem neuen Namen `/www/ACCOUNT/hello_world_02` abgelegt. Die Option `-r` steht für "recursive". Das heißt, der Ordner wird rekursiv durchlaufen: Nicht nur `web`, sondern auch alle seinen Kind-, Enkel-, Urenkel-, ... Dateien und Ordner werden kopiert.

Wenn alles funktioniert, committen und pushen Sie sie (Commit Message: „Support for mobile devices added.“).

5 Aufräumen des öffentlichen Web-Servers

Wenn Sie mit der Optimierung des Designs für mobile Endgeräte fertig sind, sollten Sie die Testdaten auf dem öffentlichen Web-Server löschen (Vergessen Sie nicht `ACCOUNT` jeweils durch Ihren RZ-Account zu ersetzen):

```
ssh ACCOUNT@login.rz.hs-augsburg.de  
cd /www/ACCOUNT  
ls -al  
rm -rf hello_word_02  
exit
```

6 Fortsetzung des Tutoriums

Sie sollten nun [Teil 3 des Tutoriums](#) bearbeiten.

7 Quellen

Kowarschick (MMProg): [Wolfgang Kowarschick](#); Vorlesung „Multimedia-Programmierung“;
Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); 2018; [Quellengüte](#): 3 (Vorlesung)

Kategorien:

[Multimedia-Programmierung/Tutorium](#)

[Praktikum:MMProg](#)

[HTML5-Tutorium: JavaScript: Hello World](#)

[HTML5-Beispiel](#)

[Kapitel:Multimedia-Programmierung:Beispiele](#)

Diese Seite wurde zuletzt am 5. April 2022 um 11:21 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-SA 4.0](#).

