

# Multimedia-Programmierung: Best Practices

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

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

<b>Korrektheit:</b> 4 (größtenteils überprüft)	<b>Umfang:</b> 2 (wichtige Fakten fehlen)	<b>Quellenangaben</b> : 2 (wichtige Quellen fehlen)	<b>Quellenarten:</b> 3 (gut)	<b>Konformität:</b> 3 (gut)
---	--	---	---------------------------------	--------------------------------

## Inhaltsverzeichnis

- [1 CSS](#)
- [2 JavaScript](#)
- [3 Grundsätzliches](#)
- [4 Modularisierung](#)
- [5 Minimierung](#)
- [6 Quellen](#)

## 1 CSS

CSS-Code wird niemals explizit in den HTML-Code eingefügt (weder in **HTML-Tags** noch mittels des `style`-Elements). Er wird immer in CSS-Dateien bereitgestellt, die eindeutig benannt sind. Die Namen ändern sich im Laufe der Zeit nicht.

Im HTML-Header-Bereich wird **eine** zentrale CSS-Datei eingebunden. Diese ist klein und sorgt im Wesentlichen dafür, dass der Bereich „above the fold“<sup>[1]</sup>, also der Bereich, der im größten denkbaren Monitor zu Beginn zu sehen ist, korrekt dargestellt wird. (Mittels **webpack** oder eines vergleichbaren Tools kann man dafür sorgen, dass deren Inhalt *automatisch* in ein `style`-Element eingefügt wird.)

Die restlichen CSS-Dateien werden bei Bedarf dynamisch mittels JavaScript eingebunden.

Verschiedene HTML-Seiten teilen sich möglichst viele CSS-Dateien. Eine CSS-Datei, die unter einem eindeutigen Namen vom Browser einmal geladen wurde, wird üblicherweise im Browser-**Cache** gehalten und muss daher kein zweites Mal geladen werden, wenn sie nochmals verwendet wird.

CSS-Dateien werden in komprimierter Form ausgeliefert (d. h. überflüssige Kommentare, Leerzeichen und Leerzeichen werden entfernt). Die Komprimierung erfolgt automatisch mittels des besten verfügbaren CSS-Komprimierungstools.

## 2 JavaScript

JavaScript-Code wird niemals explizit in den HTML-Code eingefügt (weder in **HTML-Tags** noch mittels des `script`-Elements). Er wird immer in JavaScript-Dateien bereitgestellt, die eindeutig benannt sind. Die Namen ändern sich im Laufe der Zeit nicht.

Die `script`-Tags zum Einbinden von JavaScriptdateien stehen am Ende der Datei vor dem schließenden `body`-Tag.

Es ist auch üblich, bestimmte JavaScript-Dateien erst bei Bedarf dynamisch mittels JavaScript nachzuladen.

JavaScript-Dateien werden in komprimierter Form ausgeliefert (d. h. überflüssige Kommentare, Leerzeichen und Leerzeilen werden entfernt; Variablennamen, Funktionsnamen, Parameternamen werden durch kürzere Namen ersetzt, soweit dies möglich ist). Die Komprimierung erfolgt automatisch mittels des besten verfügbaren JavaScript-Komprimierungstools.

## 3 Grundsätzliches

---

**Don't repeat yourself!** (DRY): Vermeide grundsätzlich die Verdopplung von Code. Mehrfach vorhandener Code muss bei notwendigen Änderungen auch mehrfach geändert werden.

**Keep it simple, stupid!** (KISS): Schreibe nur Code, der auch benötigt wird. Code, der auf Vorrat angelegt wird, wird häufig nicht benötigt, muss aber trotzdem gewartet werden.

## 4 Modularisierung

---

Definiere so wenig globale Größen (Variablen und Funktionen) wie möglich, am Besten gar keine. Greife auf so wenig globale Größen wie möglich zu, am besten auf gar keine.

Verwende keine Konstanten im Code, es sein denn, diese können sich garantiert **nie** ändern.

Teile eine Anwendung in Module ein:

- Jedes Modul steht in einer eigenen Datei.
- Jedes Modul erfüllt nach Möglichkeit eine einzige, wohldefinierte Aufgabe.
- Jedes Modul greift auf möglichst wenige andere Module zu.

Halte den Code-Umfang eines jeden Moduls möglichst gering, damit der Code schnell geladen werden kann und bei mobilen Endgeräten kein unnötigen Kosten verursacht (vgl. KISS).

## 5 Minimierung

---

Füge in JavaScript-Code stets alle notwendigen Strichpunkte ein (auch wenn JavaScript fehlende Strichpunkte oft automatisch ergänzt), da anderenfalls der Code nach einer Komprimierung, bei der überflüssige Leerzeichen und Leerzeilen entfernt werden, evtl. nicht mehr funktioniert.

## 6 Quellen

---

[https://de.onpage.org/wiki/Above\\_the\\_fold](https://de.onpage.org/wiki/Above_the_fold)

<https://wiki.selfhtml.org/wiki/Webstandards>

[https://wiki.selfhtml.org/wiki/CSS/Anwendung\\_und\\_Praxis/Guter\\_CSS-Stil](https://wiki.selfhtml.org/wiki/CSS/Anwendung_und_Praxis/Guter_CSS-Stil)

**Google (Web)**: Make the Web Faster; Organisation: [Google Developers](#);

<https://developers.google.com/speed/>; Quellengüte: 3 (Web)

[http://www.w3schools.com/js/js\\_best\\_practices.asp](http://www.w3schools.com/js/js_best_practices.asp)

[https://www.w3.org/wiki/JavaScript\\_best\\_practices](https://www.w3.org/wiki/JavaScript_best_practices)

**Kowarschick (MMProg)**: Wolfgang Kowarschick; Vorlesung „Multimedia-Programmierung“;

Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); 2018; Quellengüte: 3 (Vorlesung)

---

Kategorie:

Kapitel:Multimedia-Programmierung:Beispiele

Diese Seite wurde zuletzt am 31. Oktober 2018 um 17:43 Uhr bearbeitet.

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

