

# HTML5-Tutorium: JavaScript: Entwicklungsumgebung: WebStorm

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

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

<b>Korrektheit:</b> 3 (zu größeren Teilen überprüft)	<b>Umfang:</b> 2 (wichtige Fakten fehlen)	<b>Quellenangaben</b> : 2 (wichtige Quellen fehlen)	<b>Quellenarten:</b> 3 (gut)	<b>Konformität:</b> 5 (ausgezeichnet)
------------------------------------------------------------	-------------------------------------------------	--------------------------------------------------------------	---------------------------------	------------------------------------------

## Vorlesung WebProg

[Inhalt](#) | [Visual Studio Code](#) | [Node.js](#)

Zur Erstellung von Web-Anwendungen kann [WebStorm](#) eingesetzt werden. Zunächst müssen Sie diese IDE installieren und konfigurieren. Es handelt sich um eine kommerzielle Software, aber als Student erhalten Sie eine kostenlose Lizenz.

## Inhaltsverzeichnis

### [1 Installation von WebStorm](#)

- [1.1 settings.jar](#)
- [1.2 Toolbar](#)
- [1.3 IDE-Konfiguration](#)
  - [1.3.1 JavaScript](#)
  - [1.3.2 Code-Style](#)
- [1.4 IDE-Konfiguration](#)
  - [1.4.1 TODO-Erinnerung](#)
  - [1.4.2 UTF-8-Zeichen in JavaScript-Bezeichnern erlauben](#)
- [2 Quellen](#)

## 1 Installation von WebStorm

Laden Sie sich zunächst WebStorm herunter:

<https://www.jetbrains.com/webstorm/>

Sie können diese Software zunächst 30 Tage lang ohne Lizenz benutzen. Als Student der Hochschule Augsburg erhalten Sie mit Ihrer Hochschul-E-Mail-Adresse eine kostenlose Studentenlizenz. Füllen Sie dazu das Formular auf der Antragsseite aus:

<https://www.jetbrains.com/estore/students/>

**Anmerkung:** Auf Rechnern der Hochschule ist WebStorm bereits vorinstalliert.

Installieren Sie das Softwarepaket. Dabei werden Ihnen ein paar Fragen gestellt.

Wählen Sie eines der beiden angebotenen UI Themes (Dracula oder IntelliJ) aus.

**Next: Featured Plugins**

Klicken Sie bei Markdown auf **Install**

**Start using WebStorm**

Wenn Sie schon eine Lizenz erhalten haben, geben Sie die Zugangsdaten entsprechend ein.

Anderenfalls selektieren Sie **Evaluate for free** (und geben die Lizenzdaten nachträglich ein, sobald Sie sie erhalten haben).

**Activate**

## 1.1 settings.jar

---

Laden Sie die Datei [settings.jar](#) herunter und speicher Sie sie lokal auf Ihrem Rechner. Importieren Sie diese Datei in Ihre WebStorm-Umgebung:

Wenn Sie sich auf der WebStorm-Startseite befinden:

**Configure** → **Import Settings** → **settings.jar** selektieren → **OK** → **OK**

Wenn Sie schon ein Projekt bearbeiten, gehen Sie bitte folgendermaßen vor:

**File** → **Import Settings** → **settings.jar** selektieren → **OK** → **OK**

**Die nachfolgenden Anweisungen brauchen Sie nun nicht mehr von Hand auszuführen, da mit der vorangegangenen Anweisung bereits alle dort aufgeführten Änderungen ausgeführt wurden. Sie sollten sich aber den Text dennoch durchlesen, damit Sie die Begründungen für die einzelnen Änderungen an den Defaulteinstellungen kennenlernen.**

## 1.2 Toolbar

---

Aktivieren Sie die Toolbar:

**Create New Project** → **Empty Project** (= Vorauswahl) → **Create** (Dieses Projekt wird eigentlich nicht gebraucht, es dient nur dazu, die Toolbar aktivierenden zu können; siehe nächsten Punkt)

**View** → Häkchen bei **Toolbar**

Hier finden Sie insbesondere den Reload-Button, mit dem Sie den Dateibaum in WebStorm neu laden können, wenn dieser außerhalb von WebStorm verändert wurde.

## 1.3 IDE-Konfiguration

---

WebStorm ist auf die Entwicklung mit HTML/CSS/JavaScript ausgelegt, wodurch sich die Konfiguration sehr einfach gestaltet.

Anmerkungen: Die Default-Konfigurationen (Default-Settings) finden auf der WebStorm-Startseite unter dem Button:

**Configure**

Da Sie schon ein Projekt bearbeiten, finden Sie das Konfigurationsmenü unter

File → Settings for New Projects...

Im Folgenden gehe ich davon aus, dass Sie dieses Menü bereits geöffnet haben.

## 1.3.1 JavaScript

Verifizieren Sie, dass unter Languages & Frameworks → JavaScript die JavaScript language version auf ECMAScript 6 eingestellt ist.

Language & Frameworks → JavaScript language version: ECMAScript 6

## 1.3.2 Code-Style

Standardmäßig sind WebStorm in 4 Leerzeichen als Einrückung für Code-Formatierungen eingestellt. Eine allgemeingültige Regel für einen sinnvollen Wert gibt es nicht, aber ich empfehle, lediglich 2 Leerzeichen zu verwenden. Damit ist es leichter möglich, die maximale Anzahl auf 80 Zeichen pro Zeile zu begrenzen, was einen deutlich lesbareren Code zur Folge hat als überbreite Zeilen.

Grundsätzlich gilt: **Verwenden Sie bei Programmieren niemals Tabulatorzeichen zum Einrücken von Text, sondern immer nur Leerzeichen.** In den meisten Texteditoren kann man einstellen, dass beim Drücken der Tabulatortaste Leerzeichen an Stelle von Tabulatorzeichen erzeugt werden sollen. Der Grund ist, dass ein Programmcode, der Tabulatorzeichen enthält, mit ziemlicher Sicherheit vollkommen anders aussieht, wenn man ihn in einem anderen Editor oder in einem HTML-Browser öffnet.

Passen Sie die nachfolgenden Einstellungen an:

Editor → Code Style → öffnen (Klick auf Dreieck vor den Namen)

Im rechten Fenster: Visual guides:: 80, 100, 120

Klick auf JavaScript im linkem Baummenü

- Achten Sie darauf, dass Use tab character deaktiviert ist.
- Tragen Sie bei Tab size, Indent und Continuation Indent die Zahl 2 ein.
- Klick auf Reiter Wrapping and Braces. **Öffnende Klammern sollten sich aus Gründen der Lesbarkeit immer an derselben Position innerhalb der Zeile befinden wie die zugehörige schließende Klammer.** (Üblicherweise wird dies heute nicht mehr so gemacht, um eine Zeile Platz zu sparen. Sie können natürlich alle nachfolgenden Braces-placement-Modifikationen überspringen, wenn Sie die öffnende Klammer lieber am Ende der zugehörigen Anweisung platzieren möchten. Glauben Sie aber einen erfahrenem **Lisp**-Programmierer. Die andere Variante ist besser.)
  - Braces placement → In class declaration → Next line
  - Braces placement → In function declaration → Next line
  - Braces placement → In function expression → Next line
  - Braces placement → Other → Next line

Klick auf HTML im linkem Baummenü

- Tragen Sie bei Tab size, Indent und Continuation Indent die Zahl 2 ein.
- Klick auf Other
  - Do not indent children of: → Entfernen alle Einträge bis auf html.

Klick auf JSON im linkem Baummenü

- Tragen Sie bei Tab size, Indent und Continuation Indent die Zahl 2 ein.

Klick auf CSS im linkem Baummenü

- Tragen Sie bei Tab size, Indent und Continuation Indent die Zahl 2 ein.
- Klick auf Other
  - Braces placement → Next line

- Align values → On value
- Häkchen bei Keep single-line blocks

Klick auf Sass im linkem Baummenü

- Tragen Sie bei Tab size, Indent und Continuation Indent die Zahl 2 ein.

Klick auf SCSS im linkem Baummenü

- Tragen Sie bei Tab size, Indent und Continuation Indent die Zahl 2 ein.

Klick auf Other File Types im linkem Baummenü

- Tragen Sie bei Tab size und Indent die Zahl 2 ein.

OK

**Nochmals: Vermeiden Sie es, Tab-Zeichen an Stelle von Leerzeichen zu verwenden, da die Tabulatorbreite nicht genormt ist und je nach Editor anders interpretiert wird.**

## 1.4 IDE-Konfiguration

---

Weitere Setups können Sie unter

File → Settings (Windows) / Preferences (Mac OS X)

vornehmen. Im Folgenden gehe ich davon aus, dass Sie dieses Menü bereits geöffnet haben.

### 1.4.1 TODO-Erinnerung

---

Bei einem **Repository**-Commit überprüft WebStorm vor dem eigentlichen Sichern des Projekt den Code. Wenn dabei irgendwelche Fehler entdeckt werden, fragt WebStorm sicherheitshalber noch einmal nach, ob man die Fehler wirklich nicht vor dem Sichern beheben will. Das ist eine sehr sinnvolle Funktionalität. Allerdings fügt mindestens ein **webpack**-Modul den Kommentar **TODO** in den generierten Code ein. Wenn WebStorm diesen Kommentar liest, wird ebenfalls nachgefragt, ob man diese Änderung nicht noch erledigen will, bevor man den Code speichert. Dies ist sehr lästig, da man den Code ja gar nicht selbst geschrieben hat und auch gar nicht ändern kann, da spätestens beim erneuten Generieren des Codes dieser Kommentar wieder enthalten ist. Als Workaround ändere ich den Todo-String, nach dem WebStorm sucht, von **TODO** in **TBD** (to be done) ab. Dann können Sie in Ihren Code **TBD** einfügen und werden von WebStorm bei einem Commit daran erinnert, bleiben aber gleichzeitig von Meldungen verschont, an denen Sie sowieso nichts ändern können:

Editor → TODO → Doppelklick auf `\btodo\b.*` → Ändern des Pattern in `\btbd\b.*`

### 1.4.2 UTF-8-Zeichen in JavaScript-Bezeichnern erlauben

---

In JavaScript ist es möglich, in Variablen-, Funktionsbezeichnern etc. auch Umlaute und Buchstaben anderer Sprachen (wie z. B. Griechisch) zu verwenden. Defaultmäßig ist dies in WebStorm nicht erlaubt. Sie können aber WebStorm so konfigurieren, dass Sie keine Warnmeldungen mehr erhalten, wenn Sie Nicht-SCII-Zeichen in Bezeichnern verwenden:

Editor → Inspections → im Menü rechts daneben: **Internationalization** → **Non-ASCII characers**: Häkchen entfernen

Editor → Inspections → im Menü rechts daneben: **JavaScript** → **Naming conventions** → **Identifiers with non-ASCII characers**: Häkchen entfernen

## 2 Quellen

---

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

**JetBrains (WebStorm)**: WebStorm – The smartest JavaScript IDE; Organisation: [JetBrains s.r.o.](#); <https://www.jetbrains.com/webstorm/>; Quellengüte: 3 (Web)

**JetBrains (WebStorm 2017)**: Meet WebStorm – Welcome to WebStorm help!; Organisation: [JetBrains s.r.o.](#); <https://www.jetbrains.com/help/webstorm/meet-webstorm.html>; 2017; Quellengüte: 3 (Web)

Kategorien:

[Multimedia-Programmierung/Tutorium](#)

[HTML5-Tutorium: JavaScript: Entwicklungsumgebung](#)

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

[Praktikum:MMProg](#)

[Web-Programmierung-HowTo](#)

Diese Seite wurde zuletzt am 15. Oktober 2018 um 17:14 Uhr bearbeitet.

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

