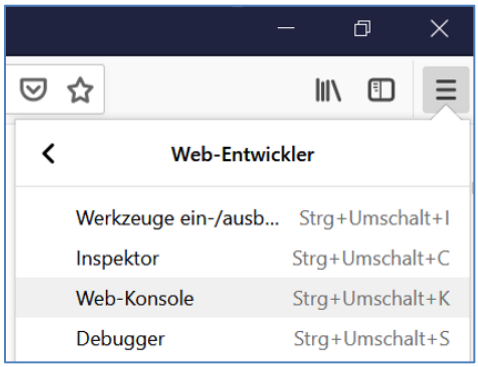
Entwicklertools

1)Fehlersuche mit einem Debugger

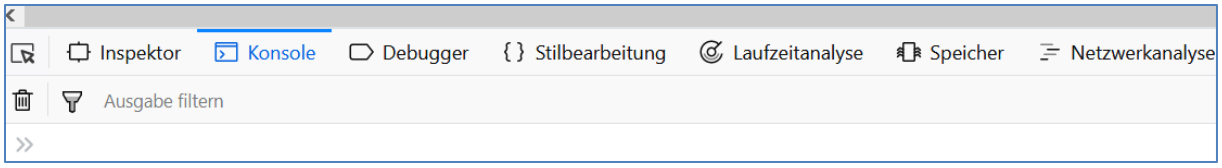
Ein spezielles Programm bzw. Tool zum Auffinden von Fehlern wird Debugger genannt. Debugger existieren für nahezu alle Programmiersprachen und für viele Skriptsprachen. Auch für JavaScript gibt es mittlerweile sehr gute Debugger.

Wer programmiert, macht Fehler. Aus diesem Grund ist die Fehlersuche fester Bestandteil der Arbeit eines Programmierers. Erfreulicherweise wird die Fehlersuche heutzutage durch eine Reihe von Werkzeugen, allen voran Debuggern, deutlich erleichtert.

Firefox: Web-Entwickler: dann Web-Konsole

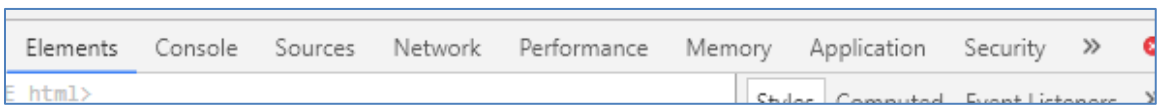


Standardmäßig erscheinen die Web-Entwicklertools im unteren Teil des Browserfensters.



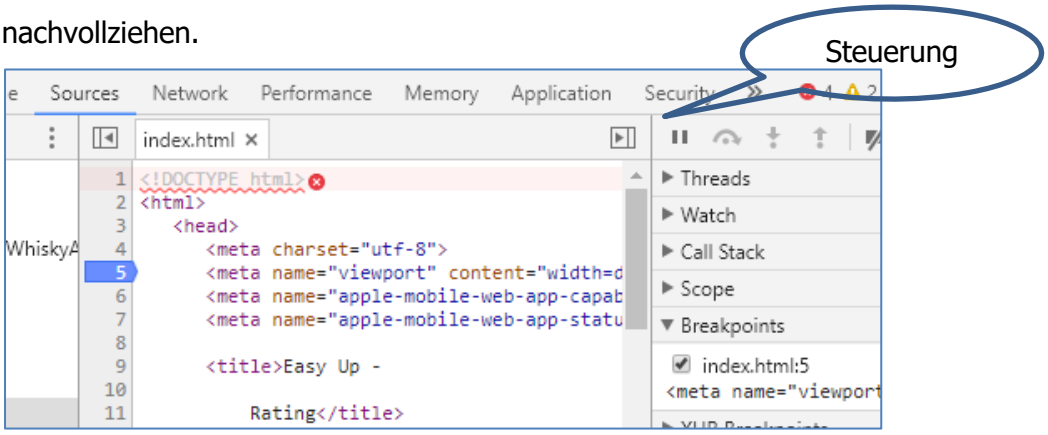
Die meisten JavaScript-Beispiele können leicht in der Konsole von Google Chrome oder Firefox ausprobiert werden. Die **Taste F12** öffnet die Entwickler-Tools und unter dem Reiter Konsole lassen sich direkt JavaScript-Befehle ausführen.

Sehr hilfreich sind die Funktionen, die in den Entwicklungs-Tools zu finden sind. Die Darstellung kann je nach Browser variieren:



- Elements: In diesem Bereich wird der dynamische DOM angezeigt. Dies bedeutet, dass dort das HTML analysiert werden kann.

- Ressource: In diesem Bereich findet man die Datenbanken, Cookies usw.
- Network: Hier kann analysiert werden, was wann vom Netz geladen wurde und wie lange dies brauchte.
- Sources: Dies ist der JavaScript-Debugger. Mit seiner Hilfe kann man schnell Fehler in der Programmlogik finden. Dazu setzt man in einer geöffneten Datei durch Klick auf eine Zeile zunächst einen Breakpoint. Wenn der Browser das nächste Mal an diesem Punkt vorbeikommt, wird die Ausführung angehalten und man kann die Variableninhalte analysieren und den Ablauf des Codes in Einzelschritten nachvollziehen.



- Console: Dies ist die JavaScript- und Fehler-Konsole. In dieser kann man per „console.log()“ Meldungen ausgeben oder andere Fehlermeldungen analysieren.

Im Chrome-Browser sind die Tools im Menü Anzeigen, Entwickler, Entwickler-Tools zu finden.

1a)Addon bei Firefox:

Für die konkrete Programmierung mit JavaScript respektive die Webseitenerstellung im Allgemeinen empfehlen sich folgende Addons des Firefox, die aber natürlich nicht zwingend sind:

Add-on	Beschreibung
DOM Inspector	Ein Add-on für die Analyse des DOM-Baums, dessen Darstellung in dem Add-on sehr übersichtlich ist.
Firebug	Das Schweizer Taschenmesser für Webentwickler. Das Add-on umfasst einen Debugger, ein Analyse-Tool für CSS, JavaScript, HTML, einen integrierten Sniffer und vieles mehr. Auf dieses Add-on wird im Buch mehrfach zurückgegriffen, denn selbst die schon sehr guten integrierten Entwicklertools von Firefox bieten nicht alle Möglichkeiten, die Firebug bereitstellt.

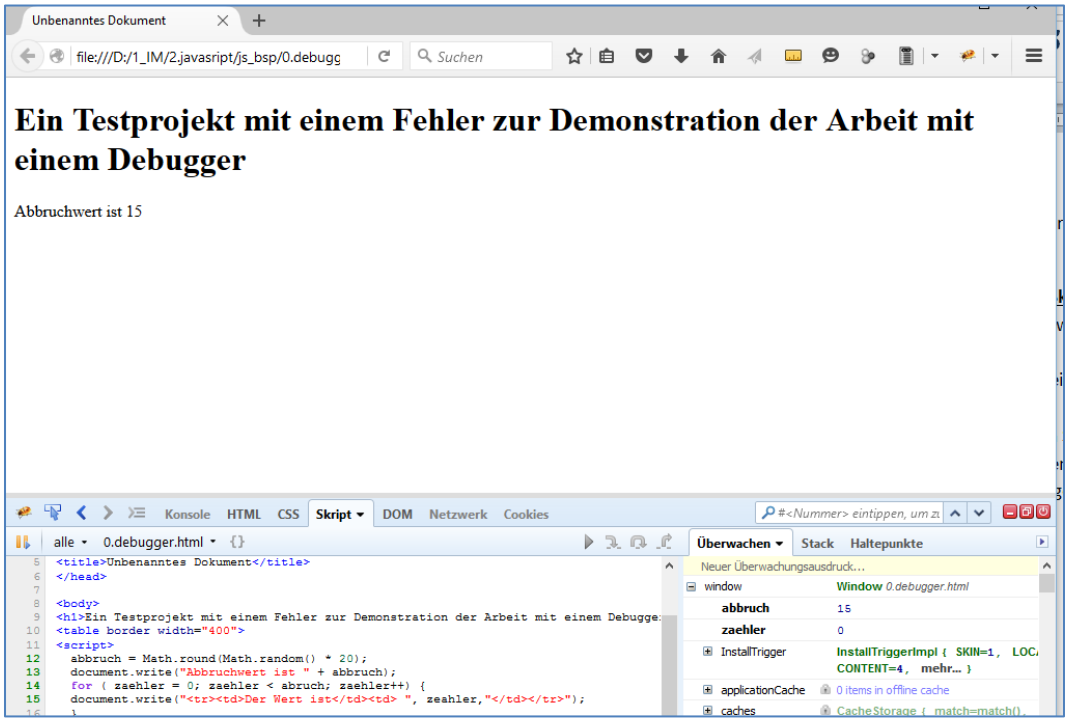
Tool für Prüfung von JavaScript-Code und HTML Seiten…

Als Add-ons nachinstallieren: suche in Add-ons oder „plugins“ nach „firebug“.

Firebug ***http://getfirebug.com***

Mit dem Firebug kann man **Webseiten analysieren**. So kann man durch die Analyse des Quellcodes, den ein Browser tatsächlich verarbeitet, eine ganze Reihe von Fehlern erkennen, etwa ob eine externe JavaScript- oder CSS-Datei überhaupt korrekt eingebunden wurde.

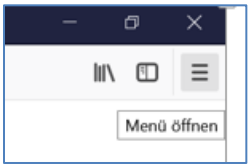
Lade die Webseite also in den Firefox-Browser und öffne Firebug. Dort wähle das Register Skript. Unter Umständen muss man ein Neuladen durchführen, damit man debuggen kann.



2)Teile des Quellcodes auskommentieren

Eine andere Technik zur Fehlerlokalisierung beruht darauf, eine größere Menge an Anweisungen, in denen man die Wanze vermutet, auszukommentieren. Hat ein Skript einen Fehler und läuft es nach der Auskommentierung (natürlich ohne den auskommentierten Part), muss sich der Fehler in den auskommentierten Anweisungen befinden. Nun verkleinert man Schritt für Schritt (sinnvoll) den auskommentierten Bereich und testet nach jeder Verkleinerung das Skript. Nach dem Schritt, nachdem das Skript nicht mehr läuft, hat man die fehlerhafte Anweisung mit hoher Wahrscheinlichkeit lokalisiert.

3)Fehlerkonsole



Die Anzeige einer Fehlerkonsole ist allerdings je nach Browsertyp und -version etwas unterschiedlich. In Firefox ist sie rechts oben in „Menü öffnen“ zu finden, wo sie mittlerweile Teil der Web-Konsole ist (Web-Entwickler → Web-Konsole), aber das hat sich in den letzten Versionen immer wieder geändert. Dennoch ist in der Regel sowohl die Art des Fehlers als auch vor allen Dingen die Zeilennummer zu erkennen, wo sich ein Fehler auswirkt.
Konsolen können die Meldungen des Codes aufzeichnen. Um eine Nachricht in das Konsolenprotokoll zu schreiben, verwendet man die Funktion „console.log“. Ihr übergibt man den String, der in der Konsole ausgegeben werden soll. Daher ist die Konsole ein Werkzeug, um Fehler im Code zu finden.

Beispiel für Debugging

Wir wollen nun einige elementare Arbeitsschritte mit einem Debugger an einem konkreten Beispiel durchspielen. Als erste Beispieldatei zum Umgang mit dem Debugger verwenden wir folgendes Listing, das bewusst einige Fehler enthält:

Eine Webseite mit Fehlern im Skript ...

```
<body>
<h1>Ein Testprojekt mit einem Fehler zur Demonstration der Arbeit mit einem Debugger</h1>
<table border width="400">
<script>
        abbruch = Math.round(Math.random() * 20);
        document.write("Abbruchwert ist " + abbruch);
        for ( zaehler = 0; zaehler < abruch; zaehler++) {
        document.write("<tr><td>Der Wert ist</td><td> ", zeahler,"</td></tr>");
        }
</script>
</body>
```

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Unbenanntes Dokument</title>
</head>

<body>
<h1>Ein Testprojekt mit einem Fehler zur Demonstration der Arbeit mit einem Debugger</h1>
<table border width="400">
<script>
    abbruch = Math.round(Math.random() * 20);
    document.write("Abbruchwert ist " + abbruch);
    for ( zaehler = 0; zaehler < abruch; zaehler++) {
    document.write("<tr><td>Der Wert ist</td><td> ", zeahler,"</td></tr>");
    }
</script>
</body>
</html>
```

Die Datei enthält im Skriptbereich einen Fehler. Wenn man die Datei in einen Browser lädt, wird nur ein Teil des zu erwartenden Resultats (eine Überschrift und eine Tabelle mit dem aktuellen Wert der Zählvariablen als Inhalt jeder Zeile) zu sehen sein.

Wenn man die Fehlerkonsole des Browsers öffnet, sieht man eine Fehlermeldung:

Quelle:

Ralph Steyer, in JavaScript, die universelle Sprache zur Web-Programmierung, 2014 Carl Hanser Verlag München, S. 132-138