

Teil 1: Unsere erste Webseite

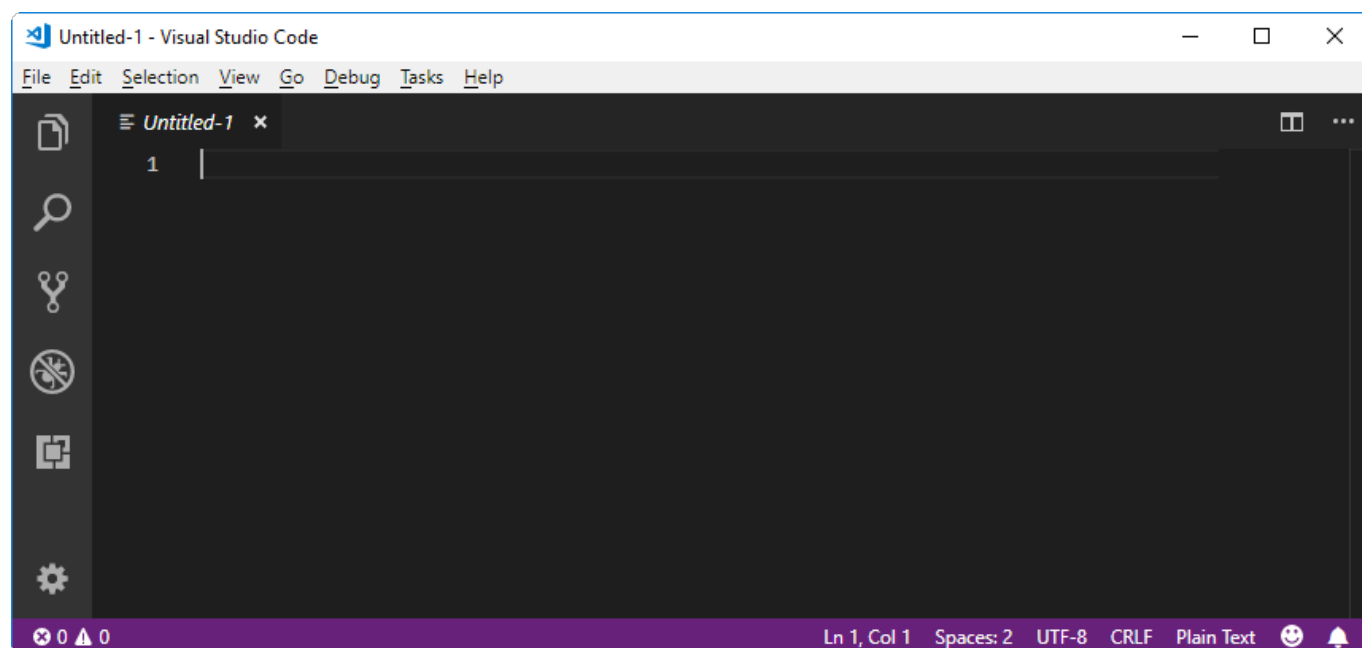
Wir legen gleich los und erstellen unsere erste Webseite.

Unsere Tools

Für die Webentwicklung brauchen wir zwei Programme: einen **Editor**, um die Dateien für die Webseite zu erstellen und einen **Browser** (z.B. Edge, Firefox, Safari oder Chrome), um unsere Webseite anzuzeigen und zu testen.

Editor

Zum Erstellen von Webseiten reicht eigentlich ein ganz normaler Texteditor. Es gibt aber Editoren, die uns die Arbeit beim Programmieren stark vereinfachen. Deshalb empfehle ich, dass Sie gleich mit einem modernen Code-Editor beginnen.



Mein aktueller Favorit unter den zahlreichen Editoren ist [Visual Studio Code](#) (kostenlos). Ich werde zwischendurch auf die Bedienung von Visual Studio Code (VS Code) eingehen, was nicht heisst, dass Sie nicht auch einen anderen Editor verwenden können. Gute Alternativen sind zum Beispiel [Atom](#) oder [Brackets](#).



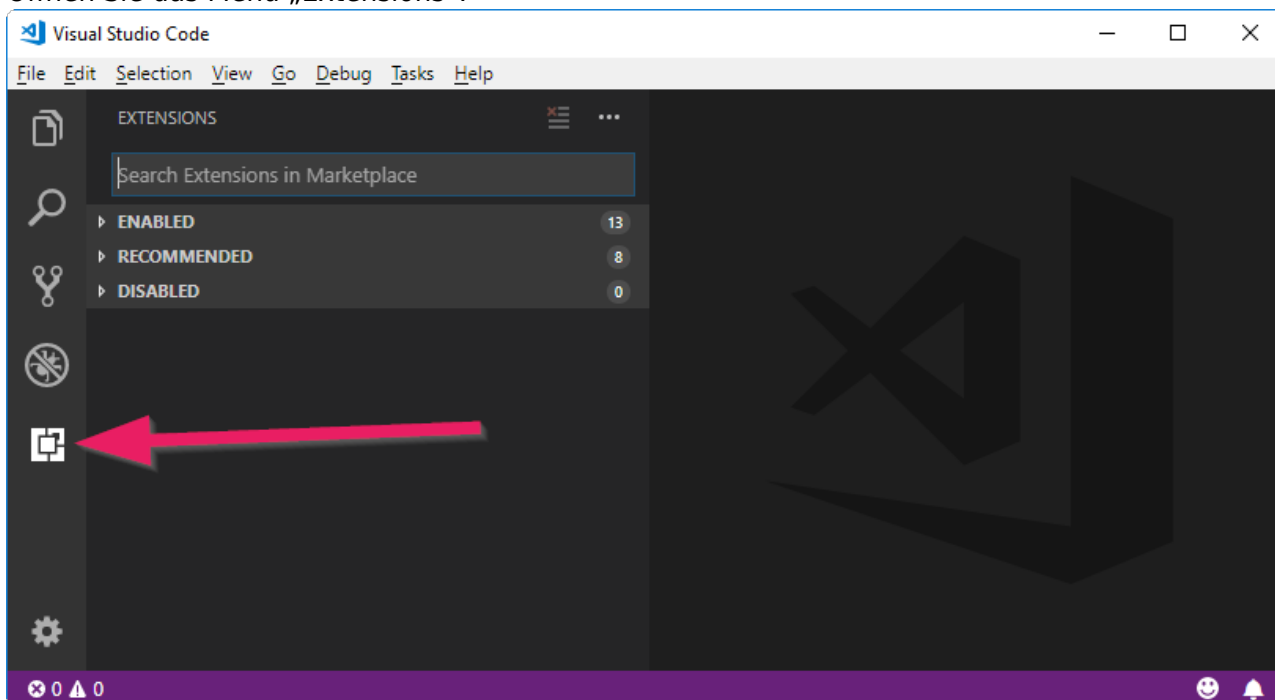
Visual Studio Code

Also, legen Sie los und installieren Sie einen Editor. Wenn Sie nicht wissen welchen, nehmen Sie [VS Code](#).

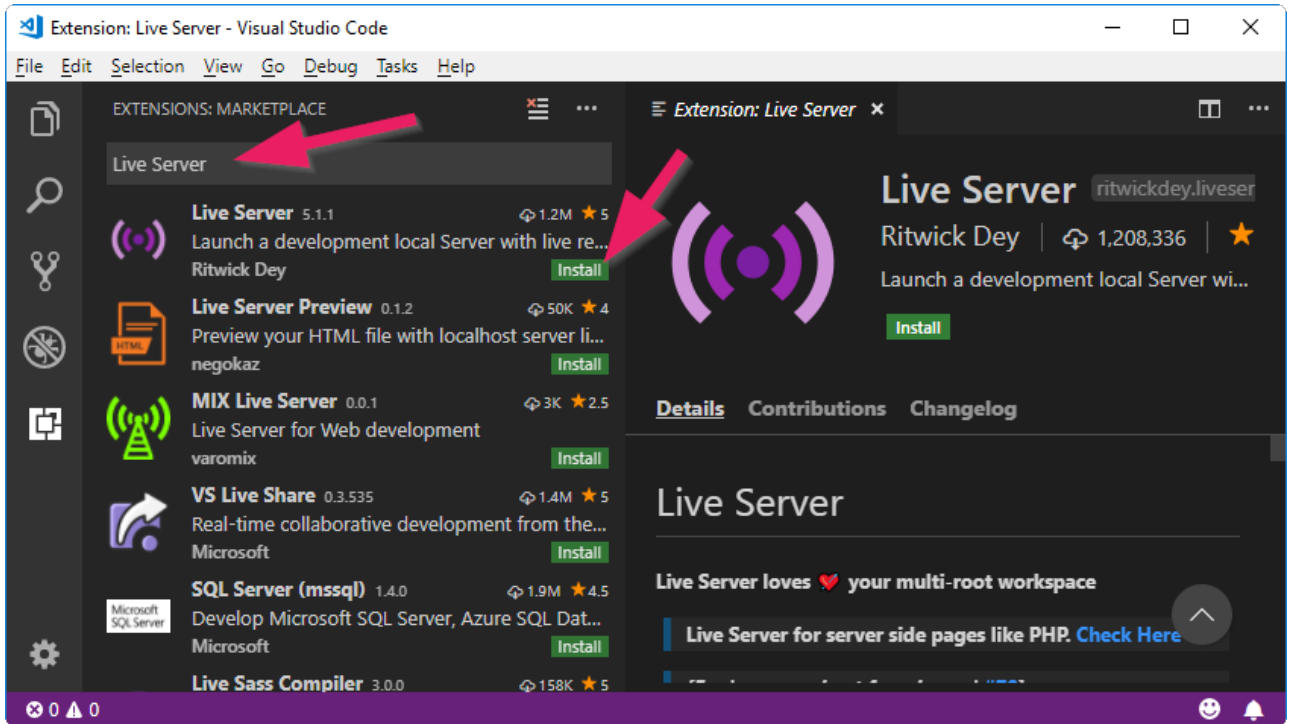
Editor Erweiterungen

Weil wir grad am Installieren sind, fügen wir gleich noch eine Erweiterungen zu VS Code hinzu, damit unser Editor Superkräfte hat: [Live Server](#).

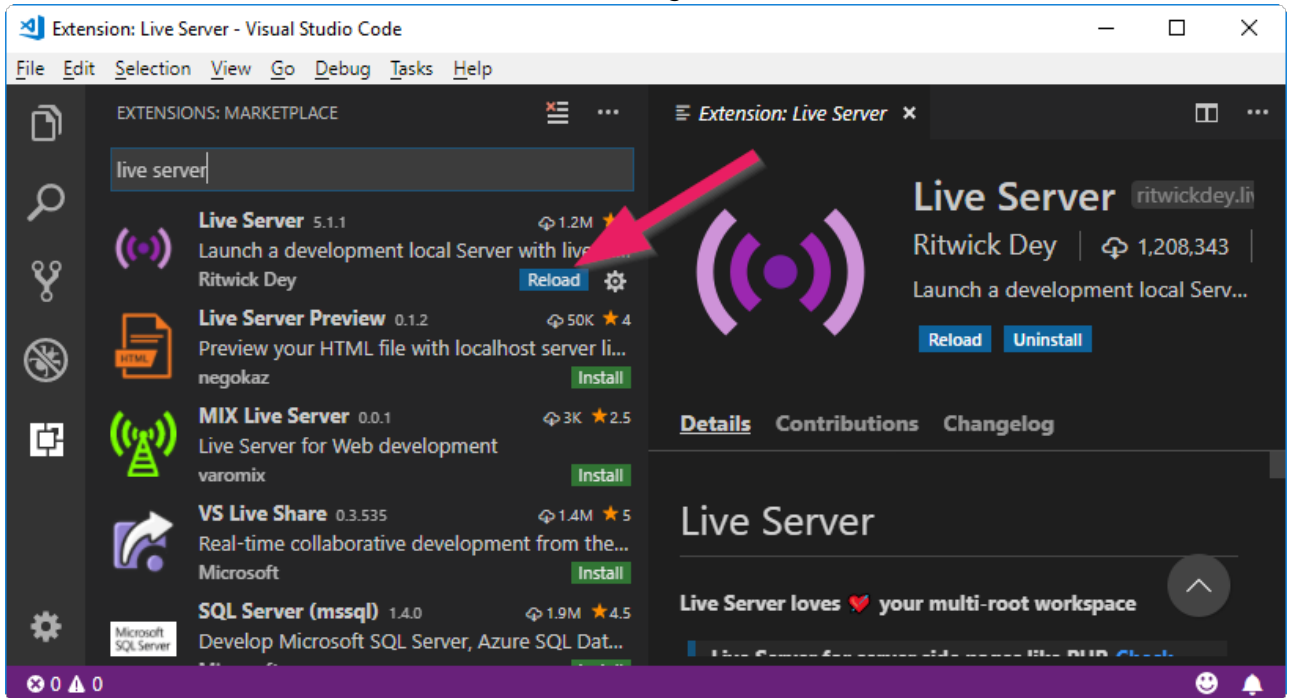
1. Visual Studio Code öffnen.
2. Öffnen Sie das Menu „Extensions“.



3. Tippen Sie in das Suchfeld „Live Server“ und installieren Sie diese Erweiterung.



4. Laden Sie Visual Studio neu, damit die Erweiterung aktiv wird.



Browser

Unsere Webseite sollte natürlich in allen gängigen Browsern (Edge, Firefox, Chrome und Safari) laufen. Für die Webentwicklung ist jedoch [Chrome](#) zu empfehlen. Chrome beinhaltet sehr nützliche Tools für Entwickler, die wir oft nutzen werden.




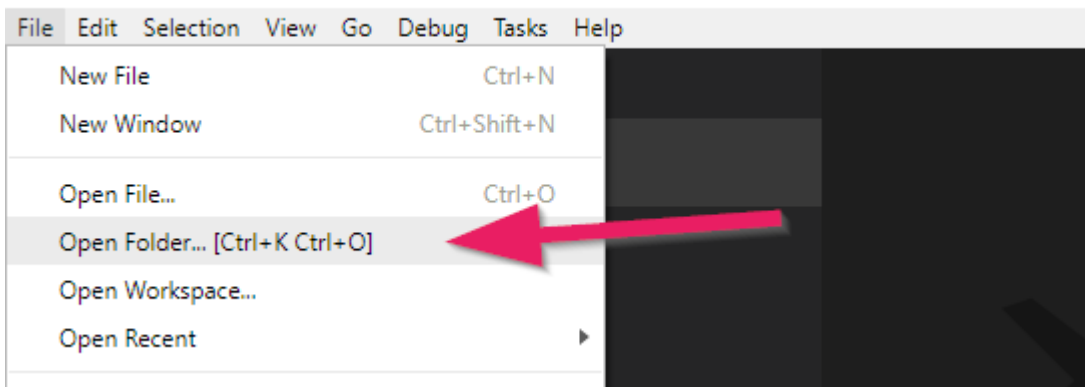
Falls Sie Chrome noch nicht haben, können Sie ihn [hier installieren](#).

Ein HTML Dokument erstellen

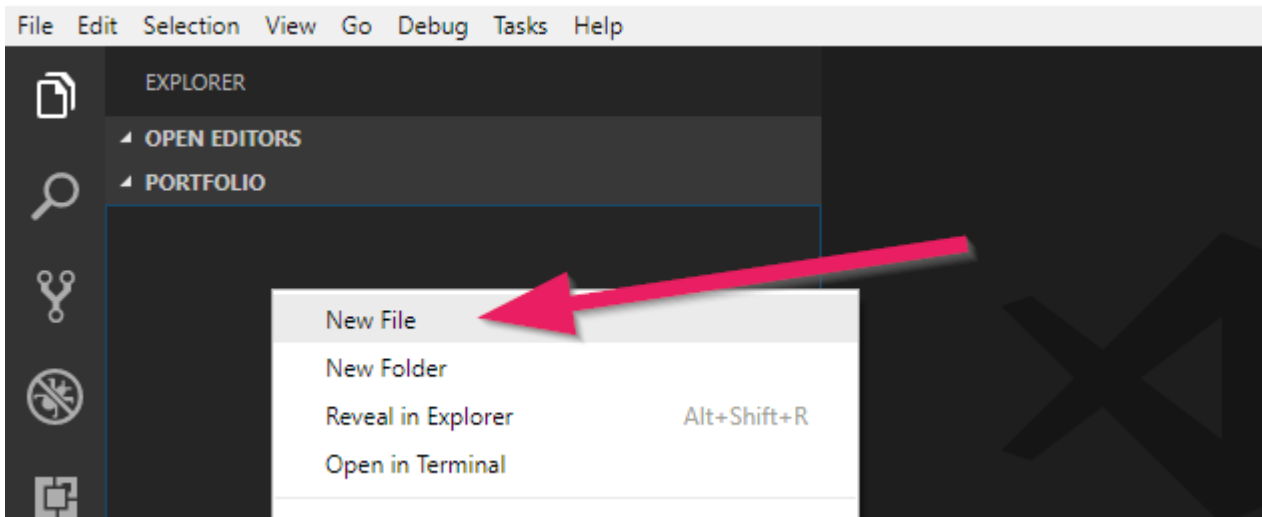
Ausgerüstet mit Editor und Browser erstellen wir das erste HTML Dokument für unsere Webseite:

1. Erstellen Sie zuerst einen Ordner auf ihrem Computer für unser erstes Projekt. Nennen Sie den Ordner zum Beispiel Portfolio.
2. Öffnen Sie VS Code.
3. Öffnen Sie das Menu *File* und wählen Sie **Open Folder...**. Suchen Sie dort den vorhin erstellten Portfolio Ordner und öffnen Sie diesen.

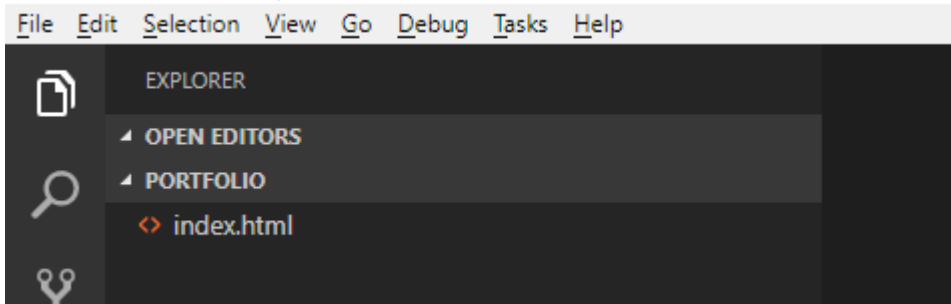
 Portfolio - Visual Studio Code



4. Klicken Sie unterhalb des Ordners *Portfolio* mit der *rechten Maustaste* und wählen Sie *New File*. Geben Sie der Datei den Namen `index.html`.



5. Nun haben wir eine ganz einfache, leere Textdatei namens `index.html`.

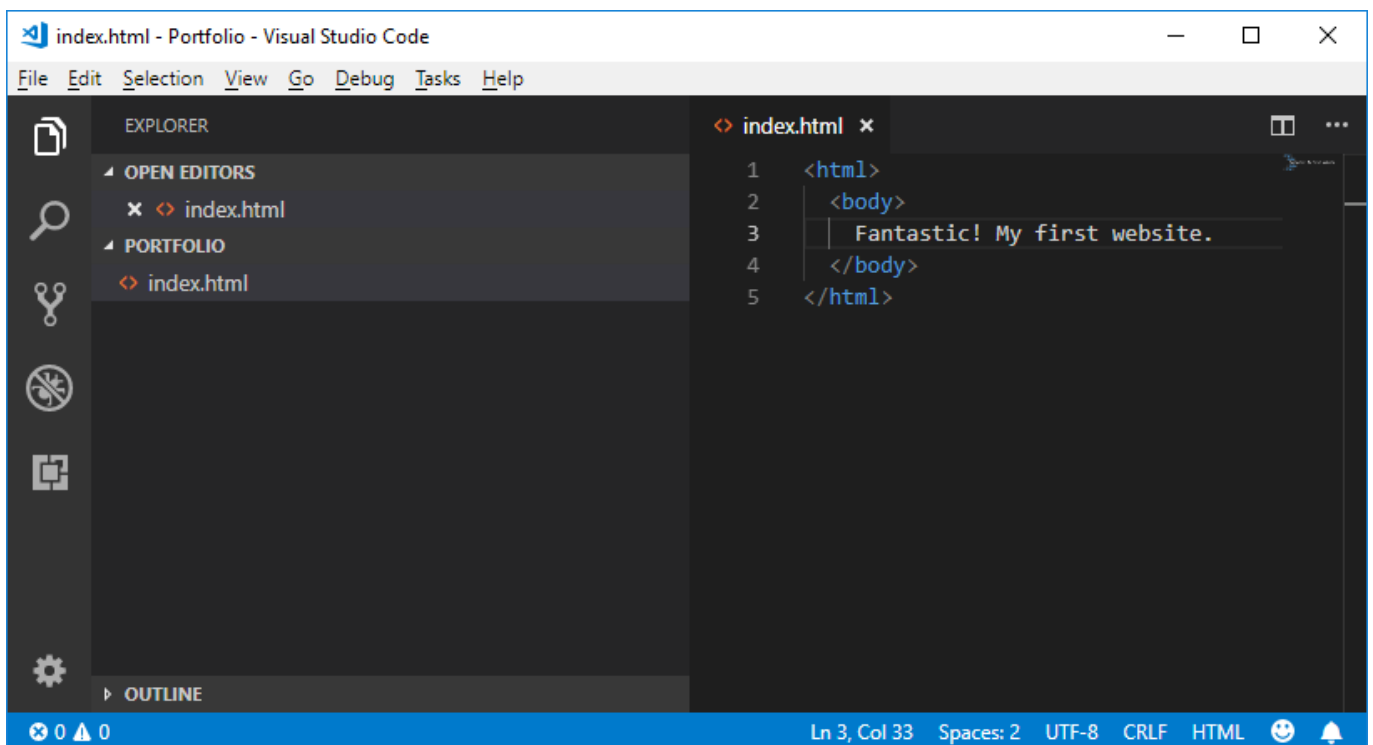


Warum index.html?

Wie Sie wahrscheinlich ahnen konnten hat der Name `index.html` eine spezielle Bedeutung. Wenn nämlich eine Webseitenadresse aufgerufen wird, z.B. `https://www.example.com`, dann wird automatisch zuerst die Datei `index.html` angezeigt, in diesem Fall `https://www.example.com/index.html`. Für uns wird `index.html` also quasi zur Startseite.

Anzeigen und aktualisieren

Nun wollen wir das Dokument mit Inhalt füllen. Tippen Sie die folgenden Zeilen ab. Als Text können Sie natürlich irgend etwas wählen.

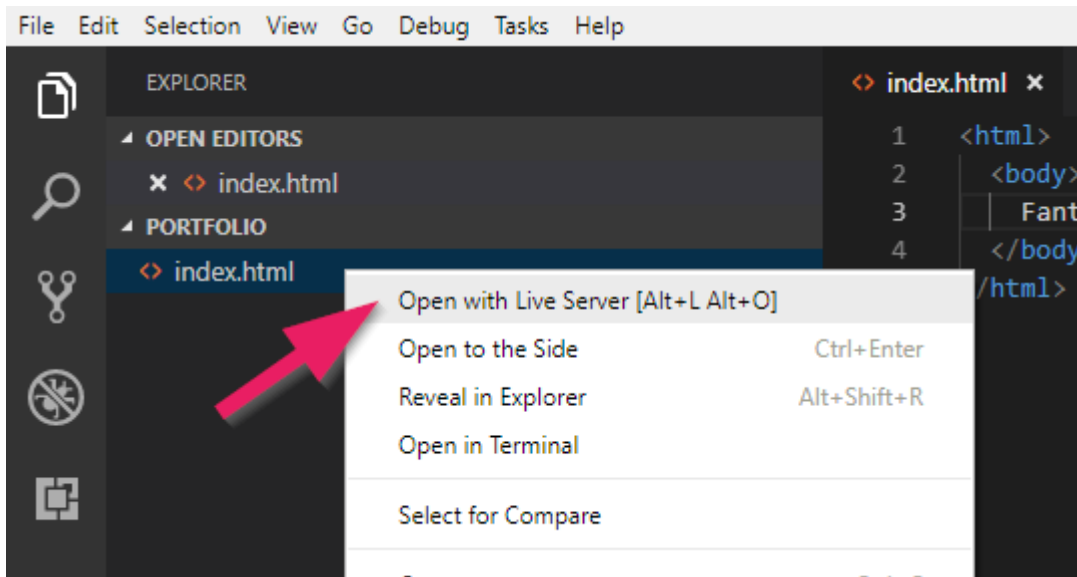


The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'Portfolio' with an 'index.html' file. The main editor area shows the content of 'index.html' with the following code:

```
1 <html>
2   <body>
3     Fantastic! My first website.
4   </body>
5 </html>
```

The status bar at the bottom indicates the cursor is at line 3, column 33, with 2 spaces, UTF-8 encoding, CRLF line endings, and the HTML language mode selected.

Zum Anzeigen der Seite verwenden wir den oben installierten **Live Server**. Klicken Sie dazu mit der *rechten Maustaste* auf die Datei `index.html` (unterhalb des Portfolio-Ordners, nicht oben wo die offenen Dateien angezeigt werden). Wählen Sie **Open with Live Server**. Nun sollte sich ein Browser-Fenster öffnen mit Ihrer ersten Website. Wann immer Sie Änderungen speichern, sollte die Website automatisch aktualisiert werden.



Falls die Seite **nicht im gewünschten Browser angezeigt wird**: Ändern Sie auf ihrem Computer die Einstellungen so, dass ein anderer Browser der Standard-Browser ist. Alternativ können Sie auch die Adresse (z.B. `http://127.0.0.1:5500/index.html`) aus dem einen Browser rauskopieren und in einem anderen Browser einfügen.

Falls die Seite **nicht aktualisiert wird** bei Änderungen: Speichern Sie alle Dateien und klicken Sie im Browser die Taste F5 oder `Ctrl+R`.

Gratuliere! Sie haben soeben Ihre erste Webseite erstellt!

Richtiges Einrücken

Damit Sie die Übersicht behalten ist es wichtig, dass Sie die Zeilen immer korrekt mit der **Tabulator-Taste** einrücken. Schauen Sie den **Beispielcode immer ganz genau an** und rücken Sie Ihren Code entsprechend ein. Für den Browser spielt das keine Rolle. Aber für uns als Programmierer ist das wichtig, damit wir den Überblick behalten.

Tipp 1: Mit `Shift+Tab` können Sie nach den Code nach links schieben.

Tipp 2: Wenn Sie mehrere Zeilen markieren können Sie diese mit `Tab` oder `Shift+Tab` gleichzeitig einrücken.

Tipp 3 (automatisch): Rechte Maustaste irgendwo im Programmcode und dann `Format Document`.

HTML-Elemente

Tags

Im Beispiel oben haben Sie bereits die für HTML typischen Zeichen mit den spitzen Klammern

gesehen. Man nennt diese **Tags**.

HTML-Elemente bestehen meist (aber nicht immer) aus zwei Tags, einem öffnenden und einem schliessenden. In unserem Beispiel ist `<html>` ein öffnendes Tag und `</html>` mit dem Schrägstrich ein schliessendes Tag.

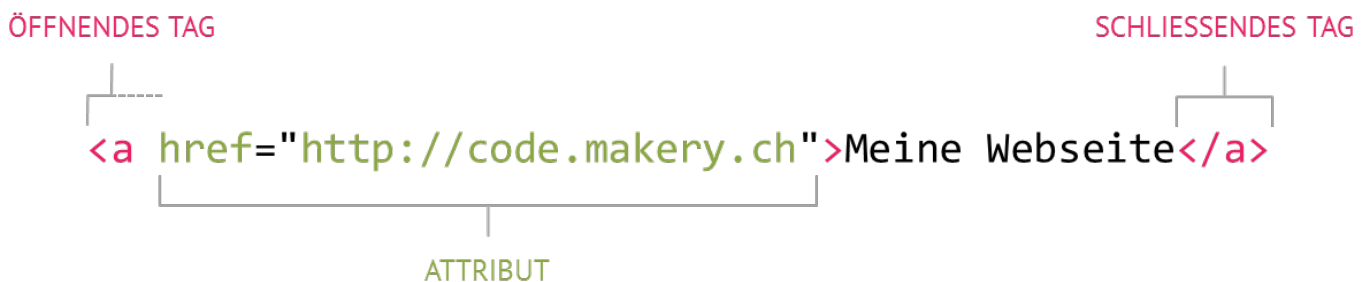
Das, was zwischen dem öffnenden und dem schliessenden Tag steht, ist der Inhalt des HTML-Elements. Mit `<html>` und `</html>` sagt wir also dem Browser, dass dazwischen das HTML unserer Webseite steht.

Das zweite Tag, welches wir gesehen haben, ist das `<body>`-Tag. Es besagt, dass alles zwischen dem öffnenden `<body>`-Tag und dem schliessenden `</body>`-Tag im Hauptbereich des Browsers angezeigt werden soll.

Attribute

Mit Attributen werden zusätzliche Informationen zu einem Element angegeben. Sie stehen im öffnenden Tag des Elements und beinhalten einen Namen und einen Wert.

Als Beispiel schauen wir uns das HTML-Element für einen Link an. Es ist wahrscheinlich eines der wichtigsten Elemente — ja was wäre das Internet ohne Links?



Das abgebildete `a`-Element enthält ein `href`-Attribut (kurz für „Hypertext Reference“) mit dem Wert `https://code.makery.ch`. Der Browser weiss damit, dass er den folgenden Link anzeigen muss: [Meine Webseite](https://code.makery.ch)

Grundgerüst einer HTML-Seite

Die Elemente `<html>` und `<body>` haben wir bereits kennen gelernt. Die Grundstruktur einer HTML-Seite beinhaltet aber üblicherweise noch ein paar weitere Elemente. Übernehmen Sie für Ihre Webseite den folgenden Code. Anschliessend werden wir ihn besprechen.

HTML Grundgerüst

```
<!DOCTYPE html>
<html>
  <head>
```

```
<meta charset="utf-8">
<title>Web Portfolio von Marco</title>
</head>
<body>
  <h1>Web Portfolio von Marco</h1>
  <p>Hier kann irgend ein Bla Bla stehen.</p>
</body>
</html>
```

Erklärung

- Setzen Sie `<!DOCTYPE html>` immer an die erste Stelle. Es sagt dem Browser, welcher Dokumententyp verwendet wird.
- Das `<html>`-Tag bezeichnet den Start und `</html>` das Ende des Dokumentes.
- Das `<head>`-Element beinhaltet zusätzliche Informationen über die Seite. Im Gegensatz zum `<body>`-Element werden diese Informationen nicht im Hauptbereich des Browsers angezeigt.
 - Innerhalb vom `<head>` sollte eine Angabe stehen über den Zeichensatz: `<meta charset="utf-8">`. Wenn Sie den Zeichensatz nicht angeben, kann es vorkommen, dass zum Beispiel die Umlaute ä, ö und ü falsch angezeigt werden. Vielleicht haben Sie gemerkt, dass das `<meta>` Element kein schliessendes Tag besitzt. Es gibt ein paar Elemente ohne schliessende Tags (`
`, ``, etc.), sie sind aber die Ausnahme.
 - Weiter steht dort meist das Titelement `<title>`. Der Titel wird in der Titelleiste oben im Browserfenster angezeigt.
- Alles innerhalb des `<body>`-Elements wird im Hauptbereich des Browsers angezeigt.
 - Ein `<h1>`-Element bezeichnet die Hauptüberschrift. Untergeordnete Überschriften können mit `<h2>`, `<h3>`, `<h4>`, `<h5>` und `<h6>` erstellt werden.
 - Alles, was zwischen `<p>` und `</p>` steht ist ein Textabsatz (auch **Paragraph** genannt).
- Nach jedem öffnenden Tag wird das nächste Element eingerückt (mit Tabulator oder zwei Leerschlägen) für eine bessere Übersicht. Dies sollten Sie sich unbedingt auch angewöhnen.

Tipp 1: Halten Sie das HTML Grundgerüst stets griffbereit. Wir werden es für jede neue HTML-Seite verwenden!

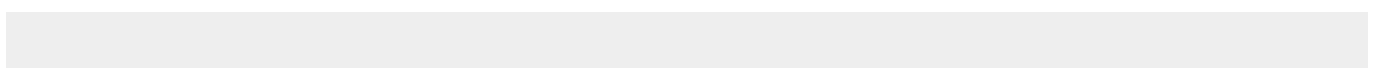
Tipp 2: Verwenden Sie die Tastenkombination `Ctrl+S` zum Speichern.

Tipp 3: Verwenden Sie die Tastenkombination `Ctrl+Z` für Rückgängig.

Mit diesen Grundlegenden HTML-Elementen sind wir nun bestens vorbereitet, um unsere Webseite auf ein neues Level zu bringen. Als erstes fügen wir noch ein Bild ein, damit die Startseite unseres Portfolios etwas interessanter wird.

Ein Bild einfügen

Um ein Bild einzufügen verwenden wir das ``-Element. Das folgende Beispiel würde ein Bild von mir einfügen:




```

```

Das ``-Element hat nur ein öffnendes aber kein schliessendes Tag. Es enthält ein `src`- und ein `alt`-Attribut. Mit dem `src`-Attribut wird die URL, d.h. der Ort und Name des Bildes angegeben. Das `alt`-Attribut ist ein „alternativer Text“, der den Inhalt des Bildes beschreibt. Dieser Text wird von Suchmaschinen verwendet oder wenn aus irgend einem Grund das Bild nicht angezeigt werden kann, z.B. bei einem Screenreader für Blinde.

Relative und absolute URLs

Für das `src`-Attribut von Bildern, aber auch für das `href`-Attribut von Links, werden URLs verwendet. Mit der URL wird also die „Adresse“ einer anderen Datei (z.B. eine andere Webseite oder ein Bild) angegeben. Je nach dem, wo diese Datei liegt, muss entweder eine *relative* oder eine *absolute* URL verwendet werden.

Wenn eine Datei **auf der gleichen Internetseite** liegt, dann wird eine **relative URL** verwendet. Diese besteht, wie wir es oben im Beispiel gesehen haben, nur aus dem Namen der Datei.

Eine *relative URL* ist entweder *relativ* zur aktuellen HTML-Seite oder *relativ* zum Hauptordner unserer Website. Wenn die Zieldatei in einem anderen Ordner liegt, muss dies berücksichtigt werden. Wenn das Bild aus dem Beispiel oben in einem Unterordner namens `bilder` liegt, dann würde die *relative URL* `bilder/marco.jpg` lauten. Falls die Datei in einem übergeordneten Ordner liegt, dann können wir mit `../` in diesen Ordner gelangen. Die URL wäre in diesem Fall `../marco.jpg`.

Anstatt von der aktuellen Datei auszugehen, können wir auch im Hauptordner unserer Website starten indem wir mit einem Slash beginnen wie `/marco.jpg`. Wir können diese URL auf irgendeiner Unterseite oder in einem Unterordner verwenden und es wird immer im Hauptordner nach dieser Datei suchen. Dies ist hilfreich zum Beispiel für Navigationslinks, wie wir in einem späteren Teil sehen werden.

Wenn die Datei **auf einer anderen Internetseite** liegt, dann wird eine **absolute URL** verwendet. Dabei wird der ganze Domainname angegeben. Ein Beispiel wäre `https://code.makery.ch/bilder/marco.jpg`.

Merken Sie sich zu URLs:

- Im gleichen Ordner verwenden wir den direkten Dateinamen, z.B. `marco.jpg`.
- Zwei Punkte (`..`) verweist auf das übergeordnete Verzeichnis.
- Wenn wir beim Hauptordner starten möchten, dann setzen wir ein `/` vorne hin, z.B. `/marco.jpg`.

Das Portfolio mit Bild

Wenn Sie es noch nicht gemacht haben, versuchen Sie nun ein Bild in Ihr Portfolio einzufügen. Dazu müssen Sie ein Bild in den *Portfolio*-Ordner auf Ihrem Computer kopieren. Achten Sie darauf, dass Sie den exakten Dateinamen inklusive der Dateiendung angeben.

Der gesamte Code könnte nun etwa so aussehen (ich habe noch eine Unterüberschrift und etwas mehr Text eingefügt):

index.html mit fertigem Portfolio-Code

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Web Portfolio von Marco</title>
  </head>
  <body>
    <h1>Web Portfolio von Marco</h1>

    <h2>Willkommen!</h2>

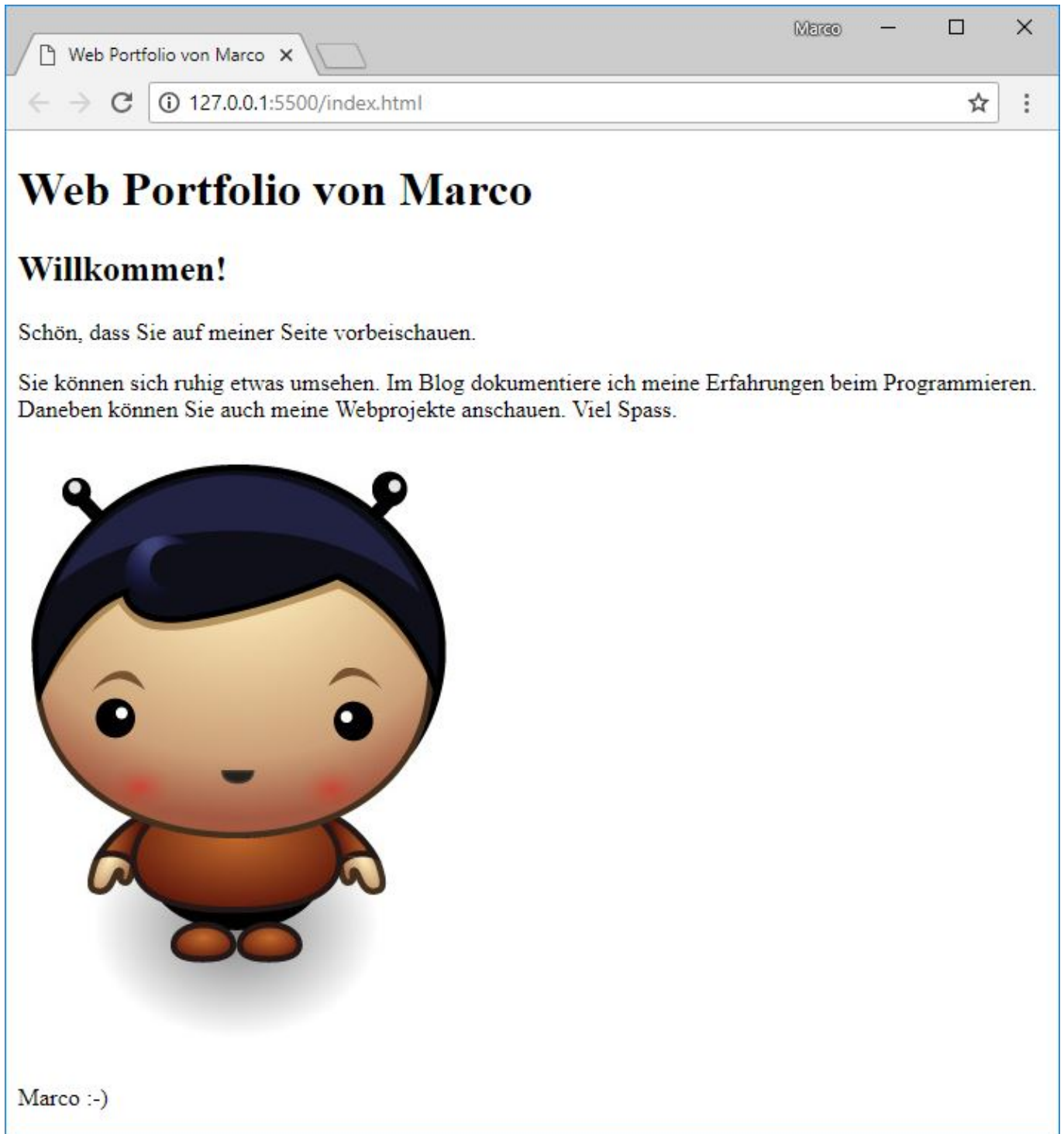
    <p>Schön, dass Sie auf meiner Seite vorbeischaun.</p>

    <p>Sie können sich ruhig etwas umsehen. Im Blog dokumentiere ich meine
    Erfahrungen beim Programmieren. Daneben können Sie auch meine Webprojekte
    anschauen. Viel Spass.</p>

    <p>Marco :-)</p>
  </body>
</html>
```

So sieht im Moment mein Portfolio im Browser aus:



The screenshot shows a web browser window with the title "Web Portfolio von Marco". The address bar displays "127.0.0.1:5500/index.html". The main content of the page includes a large heading "Web Portfolio von Marco", a sub-heading "Willkommen!", and a welcome message: "Schön, dass Sie auf meiner Seite vorbeischaauen." followed by "Sie können sich ruhig etwas umsehen. Im Blog dokumentiere ich meine Erfahrungen beim Programmieren. Daneben können Sie auch meine Webprojekte anschauen. Viel Spass." Below the text is a cartoon illustration of a character with a dark blue head, large eyes, and a brown body. At the bottom left of the page, the text "Marco :-)" is visible.

→ Im nächsten Teil lernen Sie, wie Sie die Webseite im Internet veröffentlichen können. Weiter geht's mit [Teil 2: Webseite veröffentlichen](#).

Diese Seite ist abgeleitet von [code.makery](#) von [Marco Jakob](#), verwendet unter [CC BY](#).
Sie ist lizenziert unter [CC BY-NC-SA](#) von Daniel Fahrni daniel.fahrni@bzz.ch.

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m293/tutorials/html-css/part1/start>

Last update: **2024/03/28 14:07**

