

1. VS Code und Projektordner vorbereiten

Tagesziel

Ich kann in VS Code eine HTML-Datei anlegen, ihr Grundgerüst erklären und eine Überschrift mit eigenen Absätzen im Browser anzeigen.

Du beginnst mit Visual Studio Code, kurz VS Code. Dort schreibst du den Code deiner Webseite. Danach zeigt dir ein Browser, zum Beispiel Edge oder Chrome, das Ergebnis. In diesem Modul richtest du zuerst deinen Arbeitsplatz ein und baust dann Schritt für Schritt deine erste Seite.

VS Code öffnen und den Ordner anlegen

Starte Visual Studio Code über das Windows-Startmenü. Du brauchst dafür keine Anmeldung und keine zusätzliche Erweiterung. Öffne danach mit Windows + E den Windows-Datei-Explorer. Lege in deinem Schulordner einen neuen Ordner mit dem Namen mein-hobby an.

Den ganzen Ordner in VS Code öffnen

Wähle Datei → Ordner öffnen (File → Open Folder). Suche mein-hobby und bestätige Ordner auswählen. Falls VS Code nach Vertrauen fragt, kannst du deinem selbst angelegten Ordner vertrauen. Links im Explorer von VS Code siehst du nun dein Projekt.

Die Datei index.html erzeugen

Klicke links mit der rechten Maustaste auf den Ordernamen. Wähle Neue Datei (New File), tippe index.html und drücke Enter. Rechts erscheint der leere Editor. Hier schreibst du gleich das Grundgerüst. Die Datei muss auf .html enden, nicht auf .html.txt.

Dein Projektordner hält alles zusammen. Später kommen Bilder und eine CSS-Datei dazu. Öffne beim Weiterarbeiten immer diesen Ordner. VS Code ist zum Schreiben da; der Browser ist zum Ansehen da.

2. Das HTML-Grundgerüst verstehen

HTML beschreibt den Aufbau einer Webseite. Bevor du eigenen Text einfügst, legst du die äußere Form fest: das Grundgerüst. Tippe diesen Code in deine leere index.html. Der Bereich zwischen `<body>` und `</body>` bleibt zunächst leer.

```
<!doctype html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <title>Mein Hobby</title>
</head>
<body>

</body>
</html>
```

Was bedeuten die Bereiche?

`<!doctype html>` steht ganz oben. Diese Zeile sagt dem Browser, dass er die Datei als moderne HTML-Seite behandeln soll. `<html lang="de">` umschließt die Seite. `lang="de"` gibt an, dass ihre Sprache Deutsch ist.

Zwischen `<head>` und `</head>` stehen Angaben über die Seite. `<meta charset="utf-8">` hilft dem Browser, Zeichen wie ä, ö und ü richtig zu lesen. Zwischen `<title>` und `</title>` steht der Name, den du später im Browser-Tab siehst.

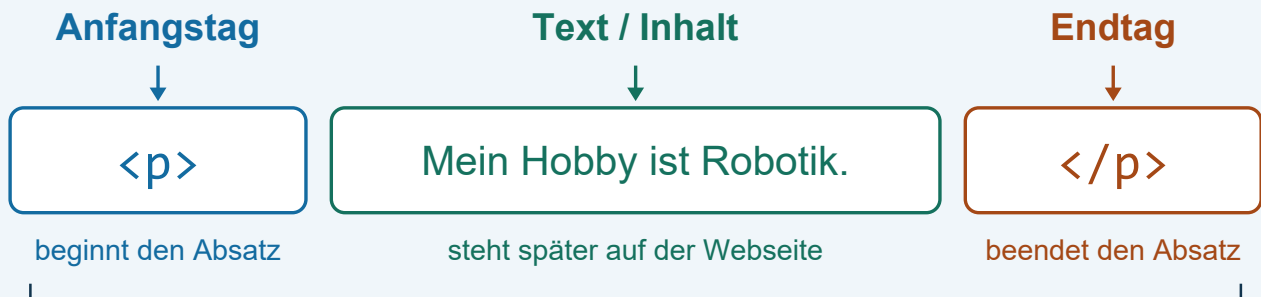
Zwischen `<body>` und `</body>` kommt der sichtbare Inhalt: später deine Überschrift und deine Absätze. In unserem Grundgerüst ist dieser Bereich noch leer. Deshalb wäre auch die eigentliche Browserseite zunächst leer.

Speichere mit Strg + S. Die eingerückten Zeilen zeigen dir, welche Teile zusammengehören. Behalte dieses Grundgerüst bei. Neue Inhalte schreibst du gleich in den body.

3. Was ist ein Tag?

Ein Tag ist eine Markierung in spitzen Klammern. Mit solchen Markierungen teilst du dem Browser mit, was ein Inhalt ist. Ein Absatz ist zum Beispiel etwas anderes als eine Überschrift. Der Browser liest die Tags und zeigt den Inhalt passend an.

SO IST EIN HTML-ELEMENT AUFGEBAUT



Alles zusammen heißt HTML-Element.

Am Schrägstrich / erkennst du den Endtag.

Anfangstag, Text und Endtag

Der Anfangstag `<p>` öffnet einen Absatz. Danach kommt dein Text. Der Endtag `</p>` schließt den Absatz wieder. Anfang und Ende haben denselben Namen: hier den Buchstaben p. Der zusätzliche Schrägstrich / kennzeichnet das Ende.

Anfangstag, Inhalt und Endtag bilden zusammen ein HTML-Element. Du schreibst alle drei Teile in VS Code. Im Browser siehst du den Text, nicht die p-Tags. Das p stammt vom englischen Wort paragraph und bedeutet Absatz.

Viele HTML-Elemente arbeiten mit einem Anfangs- und Endtag. Es gibt auch Ausnahmen: meta aus dem Grundgerüst hat zum Beispiel keinen Endtag. Für unsere ersten Inhalte p und h1 brauchst du aber immer beide Tags.

4. Einen Absatz und eine Überschrift schreiben

Dein erstes Beispiel: ein Absatz mit p

Schreibe den folgenden Absatz in den leeren body. Die beiden Sätze gehören zu einem Absatz. Du setzt deshalb ein `<p>` vor den Text und ein `</p>` nach dem letzten Satz. Die neue Zeile im Editor dient nur der Übersicht.

```
<p>Mein Hobby ist Robotik.  
    Ich baue gerne kleine Roboter.</p>
```

Dein zweites Beispiel: eine Überschrift mit h1

Setze diese Zeile im body vor deinen Absatz. h1 ist die Hauptüberschrift der Seite. Die 1 ist eine Ziffer. Auch dieses Element hat einen Anfangstag, den Text in der Mitte und einen Endtag.

```
<h1>Mein Hobby Robotik</h1>
```

Mit einer Überschrift und zwei Absätzen sieht der body zum Beispiel so aus. Ändere deinen vorhandenen body entsprechend. Füge keinen zweiten body in die Datei ein.

```
<body>  
    <h1>Mein Hobby Robotik</h1>  
    <p>Ich baue gerne kleine Roboter.  
        Besonders spannend finde ich ihre Bewegung.</p>  
    <p>Angefangen habe ich mit einem Bausatz.  
        Jetzt probiere ich eigene Ideen aus.</p>  
</body>
```

title und h1 sind nicht dasselbe: title steht im head und benennt den Browser-Tab. h1 steht im body und erscheint als große Überschrift auf der Seite. Du darfst für beide ähnliche Wörter verwenden.

5. Im Browser ansehen und selbst weiterarbeiten

Die gespeicherte Seite öffnen

Drücke Strg + S in VS Code. Klicke links mit rechts auf index.html und wähle Im Datei-Explorer anzeigen (Reveal in File Explorer). Klicke im Windows-Fenster mit rechts auf die Datei und wähle Öffnen mit → Edge oder Chrome. Du siehst jetzt die Überschrift und deine Absätze.

Wenn du etwas ändern möchtest, gehe zurück zu VS Code. Ändere den Text und speichere mit Strg + S. Wechsle zum Browser und lade mit Strg + R neu. Merke dir diese Reihenfolge: schreiben, speichern, ansehen.

Deine Aufgabe

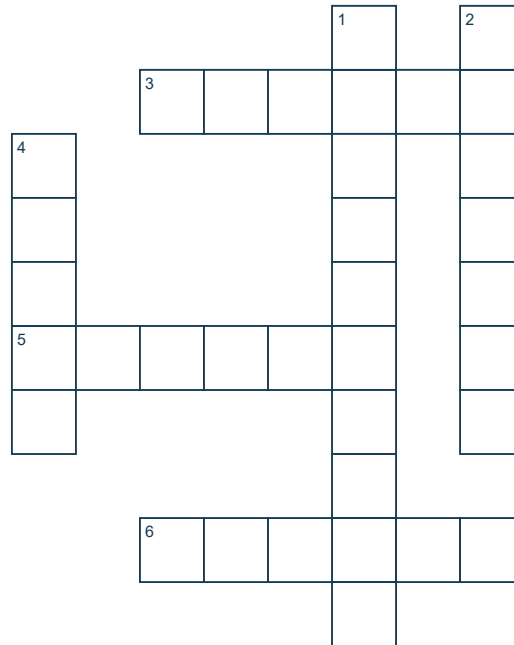
1. Öffne in VS Code deinen Ordner mein-hobby und darin index.html. Falls du während der Erklärung schon mitgeschrieben hast, arbeite in dieser Datei weiter. Erstelle keine zweite Startdatei.
2. Übernimm das vollständige HTML-Grundgerüst. Ändere nur den Text zwischen `<title>` und `</title>`, zum Beispiel zu Mein Hobby Fußball. Speichere die Datei.
3. Schreibe zwischen `<body>` und `</body>` zuerst einen Absatz mit `p`. Stelle dein Hobby in zwei eigenen Sätzen vor. Lass die `p`-Tags um den ganzen Absatz stehen.
4. Setze vor diesen Absatz eine `h1`-Überschrift mit dem Namen deines Hobbys. Achte auf die Ziffer 1 in beiden Tags: `<h1>` und `</h1>`.
5. Ergänze zwei weitere Absätze mit jeweils zwei bis drei eigenen Sätzen. Erzähle, wie du zu deinem Hobby gekommen bist und was du als Nächstes ausprobieren möchtest. Jeder Absatz bekommt sein eigenes Paar aus `<p>` und `</p>`.
6. Öffne index.html im Browser. Suche den Titel im Browser-Tab und die große Überschrift auf der Seite. Vergleiche beides mit `title` und `h1` in deinem Code.
7. Ändere einen Satz in VS Code. Drücke Strg + S, wechsle zum Browser und drücke Strg + R. Zeige im Code bei einem Absatz den Anfangstag, den Text und den Endtag.
8. Schließe VS Code und öffne den Projektordner erneut. Deine index.html soll noch vorhanden sein. Behalte den ganzen Ordner für Modul 2.

Heute entsteht

Eine gespeicherte Hobbyseite mit eigenem Tab-Titel, einer `h1`-Überschrift und drei eigenen Absätzen. Du kannst Anfangstag, Text und Endtag unterscheiden.

Zum Abschluss: Kreuzworträtsel

Trage die Begriffe zu diesem Modul ein. Ein Kästchen steht für einen Buchstaben. Schreibe groß, ohne Leerzeichen: Ä = AE, Ö = OE, Ü = UE, ß = SS.



Waagrecht →

- 3. Hier sammelst du die Dateien deines Projekts. (6 Buchstaben)
- 5. Diese Markierung enthält vor dem Namen einen Schrägstrich. (6 Buchstaben)
- 6. Diesen Textbereich erzeugt das p-Element. (6 Buchstaben)

Senkrecht ↓

- 1. Diese Markierung öffnet ein Element. (10 Buchstaben)
- 2. Programm, das deine Webseite anzeigt. (7 Buchstaben)
- 4. Er steht bei title und erscheint im Browser-Tab. (5 Buchstaben)

Zum Abschluss: Deine Antworten

Beantworte die drei Fragen in eigenen Worten. Du kannst dafür dein Modul und deine Hobbyseite verwenden.

1. Erkläre den Unterschied zwischen VS Code und dem Browser.

2. Nenne bei `<p>Mein Hobby</p>` Anfangstag, Inhalt und Endtag.

3. Warum stehen title und h1 an unterschiedlichen Stellen?

Online-Quiz: 10 Fragen

Öffne dein Modul auf der Kurswebsite und klicke am Schluss auf „Online-Quiz starten“. Wähle pro Frage eine Antwort. Nach der zehnten Frage siehst du deine Punkte und die Erklärungen.

Datei im Serverpaket: quiz-01.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.