

# 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.

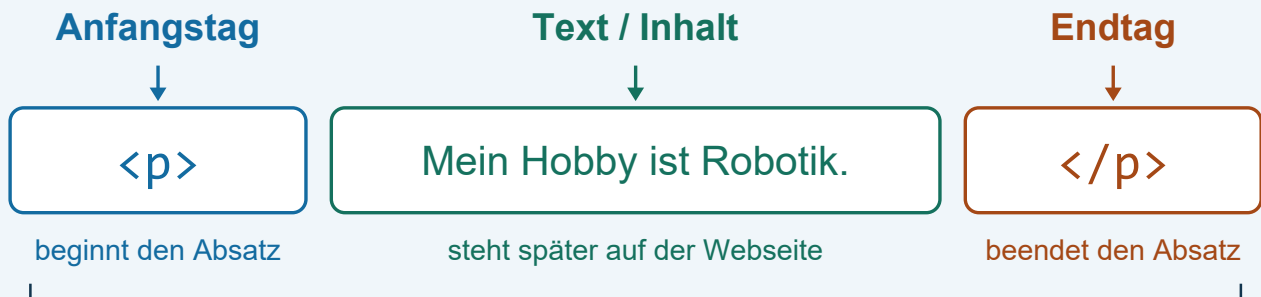
Zwischen `<body>` und `</body>` kommt der sichtbare Inhalt: später deine Überschrift und deine Absätze. In unserem Grundgerüst ist dieser Bereich noch leer. Deshalb wäre auch die eigentliche Browserseite zunächst leer.

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

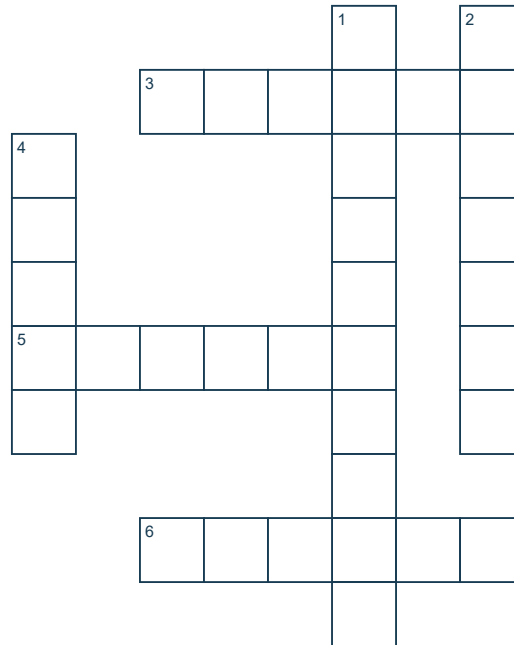
1. Öffne in VS Code deinen Ordner mein-hobby und darin index.html. Falls du während der Erklärung schon mitgeschrieben hast, arbeite in dieser Datei weiter. Erstelle keine zweite Startdatei.
2. Übernimm das vollständige HTML-Grundgerüst. Ändere nur den Text zwischen `<title>` und `</title>`, zum Beispiel 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.

# Absätze und Überschriften

## Tagesziel

Ich kann meine Hobbyseite mit einer Überschrift und mehreren Absätzen aufbauen.

Öffne in VS Code wieder deinen Ordner mein-hobby und die Datei index.html. Dein Grundgerüst bleibt erhalten. Neue sichtbare Inhalte schreibst du zwischen `<body>` und `</body>`.

## Ein Gedanke bekommt einen Absatz

Ein Besucher kennt dein Hobby vielleicht noch gar nicht. Erkläre ihm zuerst, was du machst. In einem zweiten Absatz kannst du erzählen, warum es dir Spaß macht. In einem dritten Absatz beschreibst du, wie du angefangen hast. Jeder Absatz behandelt einen eigenen Gedanken und darf aus mehreren Sätzen bestehen.

Die p-Tags gehören um den ganzen Absatz. Du brauchst nicht nach jedem Satz einen neuen Tag. Eine Leerzeile im HTML-Code hilft dir beim Lesen des Codes. Im Browser entsteht dadurch aber kein neuer Absatz. Dort sind die p-Tags entscheidend.

## Überschriften helfen beim Finden

Die h1 ist der Name deiner ganzen Seite, zum Beispiel Mein Hobby Mountainbiken. Darunter ordnen h2-Überschriften die Themen. Meine Ausrüstung und Meine Lieblingsstrecke sind zum Beispiel zwei Abschnitte. Schreibe unter jede Überschrift auch Text, damit der Besucher etwas über das Thema erfährt.

title und h1 haben verschiedene Plätze: title steht im head und erscheint im Browser-Tab. h1 steht im body und ist auf der Seite sichtbar. br erzeugt nur eine neue Zeile im selben Text. Für einen neuen Gedanken verwendest du einen neuen Absatz.

# Bausteine und Beispiel

Mit `<h1>` bekommt die Seite eine Hauptüberschrift. `<h2>` beginnt einen Abschnitt darunter. Ein Absatz steht zwischen `<p>` und `</p>`. Eine neue Zeile im Editor erzeugt noch keinen neuen Absatz im Browser. Dafür brauchst du ein neues `p`-Element.

| Schreibweise                                       | Bedeutung                                 |
|----------------------------------------------------|-------------------------------------------|
| <code>&lt;h1&gt;Mein Hobby&lt;/h1&gt;</code>       | Die Hauptüberschrift der Seite.           |
| <code>&lt;h2&gt;Meine Ausrüstung&lt;/h2&gt;</code> | Eine Unterüberschrift.                    |
| <code>&lt;p&gt;Ein Satz.&lt;/p&gt;</code>          | Ein eigener Absatz.                       |
| <code>&lt;br&gt;</code>                            | Ein Zeilenumbruch innerhalb eines Textes. |

## Diesen Ausschnitt schreibst du in den body

```
<h1>Mein Hobby Robotik</h1>
<p>Ich baue gerne kleine Roboter.
  Dabei verbinde ich Bauteile und teste ihre Bewegung.</p>
<p>Angefangen habe ich mit einem einfachen Bausatz.
  Heute probiere ich auch eigene Ideen aus.</p>

<h2>Mein nächstes Ziel</h2>
<p>Mein Roboter soll Hindernisse erkennen.
  Zuerst möchte ich dafür einen Sensor ausprobieren.</p>
```

## So verwendest du das Beispiel

h1 bis h6 sind Überschriftenebenen. Im Kurs reichen h1 und h2. Verwende auf deiner Seite eine h1. Wähle eine Ebene nach dem Inhalt; die Schriftgröße ändern wir später mit CSS.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

1. Öffne deine bestehende index.html. Lass head und body erhalten. Ergänze im body eine h1 mit dem Namen deines Hobbys.
2. Schreibe eine Einführung aus zwei Absätzen mit jeweils zwei bis drei Sätzen. Erkläre, was dein Hobby ist und wie du dazu gekommen bist. Schreibe für jemanden, der davon noch wenig weiß.
3. Ergänze die h2 Meine Ausrüstung. Beschreibe darunter in einem Absatz mindestens drei Dinge, die du für dein Hobby brauchst. Erkläre auch, wofür du eines davon verwendest.
4. Ergänze eine zweite h2 mit einem eigenen Thema, zum Beispiel Mein nächstes Ziel. Schreibe darunter zwei Absätze: Was möchtest du schaffen? Wie möchtest du beginnen?
5. Deine Seite soll jetzt mindestens fünf eigene Absätze enthalten. Verwende jeweils p-Tags. Teile lange Texte nach ihrem Inhalt auf, statt einzelne Wörter mit vielen br-Tags nach unten zu schieben.
6. Speichere die Seite und lies sie im Browser von oben nach unten. Verbessere unklare Sätze. Ändere auch den title so, dass der Browser-Tab zu deinem Hobby passt.

## Heute entsteht

Eine Hauptüberschrift, mindestens zwei Unterüberschriften und fünf verständliche eigene Absätze.

## 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.

[illegible]

**Waagrecht** →

3. Textbereich mit einem zusammengehörigen Gedanken. (6 Buchstaben)
5. Diesen erzeugt br. (13 Buchstaben)
6. Sie gibt einem Abschnitt einen Namen. (12 Buchstaben)

## Senkrecht ↓

1. Der Name im Browser-Tab. (5 Buchstaben)
2. h1 und h2 stehen auf einer unterschiedlichen ... (5 Buchstaben)
4. Sichtbarer Text gehört im body zu diesem Teil der Seite. (6 Buchstaben)

# Zum Abschluss: Deine Antworten

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

## 1. Wann beginnt ein neuer Absatz?

---

---

## 2. Wie unterscheiden sich h1 und h2?

---

---

## 3. Warum ersetzt eine Leerzeile im Code keinen neuen Absatz?

---

---

## 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-02.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.

# Text fett kursiv und durchgestrichen

## Tagesziel

Ich kann einzelne Wörter hervorheben, unterstreichen und durchstreichen.

Tags können auch innerhalb eines Absatzes stehen. Du markierst damit nur einen Teil des Textes. Die umschließenden p-Tags bleiben bestehen. Schließe immer zuerst den inneren Tag und dann den äußeren.

## Nur die gewünschten Wörter verändern

Du musst nicht den ganzen Absatz fett machen. Setze `<b>` unmittelbar vor die Wörter, die auffallen sollen, und `</b>` direkt dahinter. Der restliche Satz bleibt normal. Mit `i` funktioniert es genauso. Die äußeren p-Tags umschließen weiterhin den vollständigen Absatz.

Lies den Code von außen nach innen: Zuerst beginnt der Absatz, dann beginnt die Hervorhebung. Beende zuerst die Hervorhebung und danach den Absatz. In `<p>Ich mag <b>Robotik</b>.</p>` wird nur Robotik fett. Der Punkt gehört noch zum Absatz.

## Gestaltung mit einem Grund

Hebe wenige wichtige Wörter hervor. Wenn alles fett oder unterstrichen ist, fällt nichts mehr besonders auf. `strong` passt zu einem wichtigen Hinweis, zum Beispiel zum sorgfältigen Umgang mit Werkzeug. `em` kannst du für ein Wort verwenden, das beim Vorlesen besonders betont werden soll.

Durchstreichen passt zu einer Angabe, die sich geändert hat: Ein Treffen war am Montag und findet jetzt am Dienstag statt. Der alte Termin steht in `s`, der neue daneben. Zum Unterstreichen übernimmst du die `span`-Zeile aus dem Beispiel. Wie die `style`-Angabe darin arbeitet, lernst du später genauer.

# Bausteine und Beispiel

Mit **b** wird Text normalerweise fett dargestellt, mit *i* kursiv. **strong** kennzeichnet etwas besonders Wichtiges und wird meist ebenfalls fett angezeigt. **em** betont ein Wort und wird meist kursiv angezeigt. **s** kennzeichnet eine Angabe, die nicht mehr stimmt. Für eine reine Unterstreichung verwenden wir eine kleine fertige Stil-Angabe; CSS erklären wir ab Modul 7.

| Schreibweise                                                                   | Bedeutung                                 |
|--------------------------------------------------------------------------------|-------------------------------------------|
| <code>&lt;b&gt;Text&lt;/b&gt;</code>                                           | Fett hervorheben.                         |
| <code>&lt;i&gt;Text&lt;/i&gt;</code>                                           | Kursiv darstellen.                        |
| <code>&lt;strong&gt;Wichtig&lt;/strong&gt;</code>                              | Besondere Wichtigkeit, meist fett.        |
| <code>&lt;em&gt;wirklich&lt;/em&gt;</code>                                     | Betonung, meist kursiv.                   |
| <code>&lt;s&gt;Alter Termin&lt;/s&gt;</code>                                   | Nicht mehr gültige Angabe durchstreichen. |
| <code>&lt;span style="text-decoration: underline;"&gt;Text&lt;/span&gt;</code> | Text unterstreichen.                      |

## Textgestaltung im body

```
<p>Mein Hobby ist <b>Robotik</b>.</p>
<p>Mein Projekt heißt <i>Mini-Bot</i>.</p>
<p><strong>Wichtig:</strong> Zuerst planen.</p>
<p>Das macht mir <em>wirklich</em> Spaß.</p>
<p>Treffen: <s>Montag</s> jetzt Dienstag.</p>
<p><span style="text-decoration: underline;">
  Mein nächstes Ziel
</span></p>
```

## So verwendest du das Beispiel

`span` umfasst hier nur die Wörter, die unterstrichen werden sollen. Unterstreiche sparsam: Im Web sehen unterstrichene Wörter häufig wie Links aus. Den älteren Tag `strike` verwenden wir nicht. `u` gibt es ebenfalls, er dient aber einer inhaltlichen Kennzeichnung, etwa eines Schreibfehlers, und ist kein allgemeiner Gestaltungstag.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

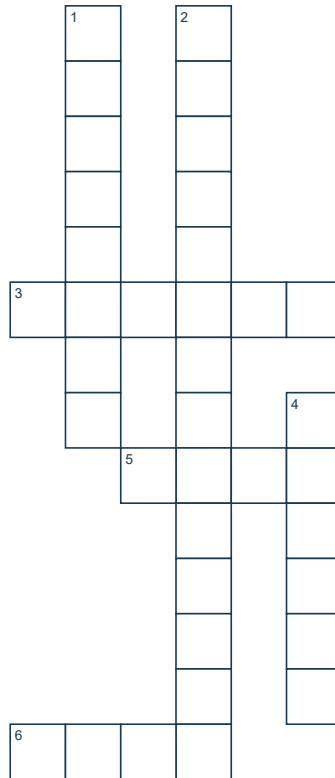
1. Arbeite in deinen vorhandenen Absätzen weiter. Wähle drei wichtige Begriffe und mache sie mit **b** fett. Der übrige Text soll normal bleiben.
2. Ergänze einen Absatz mit zwei bis drei Sätzen über ein besonderes Erlebnis bei deinem Hobby. Schreibe einen passenden Namen oder Begriff darin kursiv mit *i*.
3. Schreibe einen hilfreichen Hinweis für Anfänger. Kennzeichne die Wörter Wichtig für den Anfang mit **strong**. Erkläre danach in zwei Sätzen, was man beachten soll.
4. Ergänze einen Satz, in dem du ein Wort mit **em** betonst. Lies dir den Satz leise vor und überlege, ob die Betonung zu deiner Aussage passt.
5. Unterstreiche eine kurze Wortgruppe mit dem gezeigten `span`-Beispiel. Übernimm die Anführungszeichen, den Doppelpunkt und das Semikolon genau.
6. Schreibe einen geänderten Termin oder ein geändertes Ziel auf. Setze nur die alte Angabe in `s` und schreibe die neue daneben. Erkläre die Änderung in einem weiteren Satz.
7. Sieh dir die Seite im Browser an. Entscheide, ob wichtige Wörter gut auffallen. Nimm überflüssige Hervorhebungen zurück, damit längere Texte angenehm lesbar bleiben.

## Heute entsteht

Auf deiner Hobbyseite sieht man **fett**, *kursiv*, unterstrichen und ~~durchgestrichen~~.

# 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. So erscheint i normalerweise. (6 Buchstaben)
- 5. So erscheint b normalerweise. (4 Buchstaben)
- 6. Tag um die Wörter im Unterstreichungsbeispiel. (4 Buchstaben)

## Senkrecht ↓

- 1. Dafür steht em in einem Satz. (8 Buchstaben)
- 2. Das bewirkt text-decoration: underline; (14 Buchstaben)
- 4. Tag für besondere Wichtigkeit. (6 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 **b** und **strong**.

---

---

## 2. Schreibe einen Satz mit einem alten und einem neuen Termin. Markiere den alten mit **s**.

---

---

## 3. Warum sollte nicht der ganze Text fett und unterstrichen sein?

---

---

## 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-03.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.

# Listen Links und Bilder

## Tagesziel

Ich kann eine Liste, einen Link und ein eigenes Bild einfügen.

Für Ausrüstung oder Lieblingsdinge ist eine Liste übersichtlicher als ein langer Satz. Mit einem Link verweist du auf eine andere Seite. Mit einem Bild zeigst du etwas zu deinem Hobby.

## Eine Liste und ein Link Schritt für Schritt

Beginne eine Aufzählung mit `<ul>`. Schreibe für jeden Gegenstand eine eigene Zeile mit `<li>` und `</li>`. Schließe die ganze Liste mit `</ul>`. Der Browser setzt die Punkte automatisch. Wenn die Reihenfolge wichtig ist, zum Beispiel bei drei Arbeitsschritten, verwendest du `ol` statt `ul`.

Ein Link hat zwei Teile. Zwischen den Anführungszeichen bei `href` steht die Adresse. Zwischen `<a ...>` und `</a>` steht der Text, auf den man klickt. Schreibe zum Beispiel Zur Webseite unserer Schule. So kann ein Besucher das Ziel schon vor dem Klicken erkennen.

## Wie der Browser dein Bild findet

Speichere das Bild zuerst im Unterordner `bilder`. Der Pfad `bilder/hobby.svg` bedeutet: Öffne von `index.html` aus den Ordner `bilder` und darin die Datei `hobby.svg`. Liegt dein Foto unter `fahrrad.jpg`, muss der Pfad `bilder/fahrrad.jpg` heißen. Auf vielen Servern ist auch Groß- und Kleinschreibung wichtig.

`alt` beschreibt kurz, was auf dem Bild zu sehen ist, zum Beispiel Rotes Mountainbike auf einem Waldweg. Dieser Text hilft auch Menschen, die die Seite vorlesen lassen. `width="320"` gibt eine Breite in Pixeln vor. Schreibe unter das Bild zusätzlich, woher es stammt.

# Bausteine und Beispiel

ul erzeugt eine Liste mit Aufzählungspunkten, ol eine nummerierte Liste. Jeder Eintrag steht in li. Beim Link enthält href die Zieladresse. Beim Bild enthält src den Dateipfad. alt beschreibt den Bildinhalt in Worten. img hat keinen End-Tag.

| Schreibweise                                                            | Bedeutung                                |
|-------------------------------------------------------------------------|------------------------------------------|
| <code>&lt;ul&gt; ... &lt;/ul&gt;</code>                                 | Liste mit Punkten.                       |
| <code>&lt;ol&gt; ... &lt;/ol&gt;</code>                                 | Liste mit Nummern.                       |
| <code>&lt;li&gt; ... &lt;/li&gt;</code>                                 | Ein einzelner Listeneintrag.             |
| <code>&lt;a href="https://www.htl-lienz.at/"&gt;Schule&lt;/a&gt;</code> | Ein Link mit lesbarem Text.              |
| <code>&lt;img src="bilder/hobby.svg" alt="..." width="320"&gt;</code>   | Ein Bild mit Breite und Textalternative. |

## Liste Link und Bild im body

```
<h2>Meine Ausrüstung</h2>
<ul>
  <li>Bauteile</li>
  <li>Werkzeug</li>
  <li>Notizbuch</li>
</ul>
<p><a href="https://www.htl-lienz.at/">
  Zur Webseite unserer Schule
</a></p>

```

## So verwendest du das Beispiel

Lege in deinem Projekt den Unterordner bilder an. Kopiere die Kursgrafik hobby.svg hinein. Liegt dein eigenes Foto dort unter hobby.jpg, musst du im src auch bilder/hobby.jpg schreiben. Dateiname und Endung müssen genau passen.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

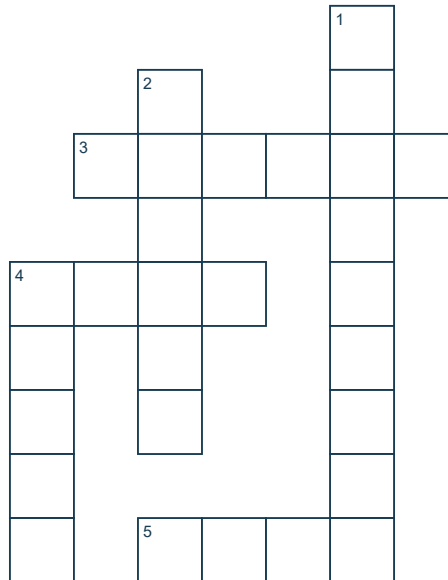
1. Ergänze unter einer passenden Überschrift eine ul-Liste mit mindestens fünf Dingen für dein Hobby. Schreibe bei jedem Eintrag in wenigen Worten dazu, wofür du das Ding brauchst.
2. Erstelle außerdem eine ol-Liste mit drei Schritten für den Beginn deines Hobbys. Die Schritte sollen in einer sinnvollen Reihenfolge stehen.
3. Lege den Unterordner bilder an. Kopiere ein eigenes Foto oder die Kursgrafik hinein. Notiere dir den vollständigen Dateinamen mit Endung, bevor du den img-Tag schreibst.
4. Füge das Bild unter einem passenden Absatz ein. Setze src auf deine Datei, ergänze einen treffenden alt-Text und beginne mit width="320".
5. Schreibe unter das Bild zwei bis drei Sätze. Beschreibe, was es zeigt und warum es zu deinem Hobby passt. Ergänze die Bildherkunft, zum Beispiel Bild: eigene Aufnahme.
6. Füge einen Link zur Schulwebsite oder zu einer passenden Informationsseite ein. Gib ihm einen verständlichen Linktext und erkläre in einem Satz, was dort zu finden ist.
7. Speichere und öffne die Seite im Browser. Klicke den Link an und kehre zurück. Wenn das Bild fehlt, vergleiche Ordnername, Dateiname und Endung mit deinem src.

## Heute entsteht

Zwei Listen, ein sinnvoll beschrifteter Link und ein Bild mit Beschreibung und Herkunftsangabe.

# 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. Diesen Wert legt width fest. (6 Buchstaben)
4. Darauf klickst du, um eine Zielseite zu öffnen. (4 Buchstaben)
5. Dieses fügst du mit img ein. (4 Buchstaben)

## Senkrecht ↓

1. Er zeigt dem Browser den Weg zur Bilddatei. (9 Buchstaben)
2. Der Bereich bilder im Projekt ist ein Unter... (gesuchter Grundbegriff). (6 Buchstaben)
4. Mehrere li-Einträge bilden gemeinsam eine ... (5 Buchstaben)

# Zum Abschluss: Deine Antworten

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

## 1. Wann verwendest du ul und wann ol?

---

---

## 2. Erkläre den Pfad bilder/hobby.jpg.

---

---

## 3. Formuliere einen passenden alt-Text für dein eigenes Bild.

---

---

## 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-04.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.

# Tabellen mit sichtbarem Rahmen

## Tagesziel

Ich kann eine kleine Tabelle erstellen, deren Zellen sichtbare Rahmen haben.

Eine Tabelle passt zu Angaben, die nach denselben Merkmalen verglichen werden: zum Beispiel Tag, Aktivität und Dauer. Eine waagrechte Reihe heißt Zeile, eine senkrechte Reihe Spalte. Ein einzelnes Feld ist eine Zelle.

## Eine Tabelle entsteht Zeile für Zeile

Zeichne zuerst eine kleine Tabelle auf Papier. Schreibe oben die Begriffe Tag, Aktivität und Dauer. Darunter steht zum Beispiel Dienstag, Training und 45 Minuten. Jedes Feld in dieser Zeichnung wird später eine Zelle im HTML-Code. So weißt du vor dem Tippen, welche Angaben zusammengehören.

Im Code beginnt eine Zeile mit `<tr>`. Darin stehen die Zellen nebeneinander. In der Kopfzeile verwendest du `th`, in den übrigen Zeilen `td`. Erst wenn alle Zellen einer Zeile fertig sind, schließt du mit `</tr>`. Danach beginnt die nächste Zeile.

## Warum der Rahmen an jede Zelle gehört

Ohne Gestaltung kann eine Tabelle wie Text ohne Linien aussehen. Die Angabe `border: 1px solid black;` zeichnet einen dünnen schwarzen Rahmen. Damit jedes einzelne Feld umrandet ist, steht diese Angabe an jedem `th` und `td`. Nur ein Rahmen um `table` reicht für die inneren Linien nicht.

`border-collapse: collapse;` führt benachbarte Rahmen zu einer gemeinsamen Linie zusammen. Übernimm diese Schreibweise zunächst genau. Wenn du eine Spalte ergänzt, braucht jede Zeile eine zusätzliche Zelle. Eine dritte Spalte bedeutet also drei Zellen in jeder Zeile, auch in der Kopfzeile.

# Bausteine und Beispiel

table umschließt die Tabelle. tr beginnt eine Zeile. th ist eine Überschriftszelle, td eine normale Datenzeile. Die Rahmen entstehen durch style="border: 1px solid black;". Das bedeutet: eine einen Pixel breite, durchgezogene schwarze Linie. Die Angabe kommt auch auf die einzelnen Zellen.

| Schreibweise               | Bedeutung                                  |
|----------------------------|--------------------------------------------|
| <table> ... </table>       | Die ganze Tabelle.                         |
| <tr> ... </tr>             | Eine Tabellenzeile.                        |
| <th> ... </th>             | Eine Überschriftszelle.                    |
| <td> ... </td>             | Eine Datenzeile.                           |
| border: 1px solid black;   | Sichtbarer schwarzer Rahmen.               |
| border-collapse: collapse; | Benachbarte Rahmen werden zusammengeführt. |

## Tabelle für den body

```
<table style="border-collapse: collapse;">
  <tr>
    <th style="border: 1px solid black;">Tag</th>
    <th style="border: 1px solid black;">Aktivität</th>
  </tr>
  <tr>
    <td style="border: 1px solid black;">Dienstag</td>
    <td style="border: 1px solid black;">Bauen</td>
  </tr>
  <tr>
    <td style="border: 1px solid black;">Freitag</td>
    <td style="border: 1px solid black;">Testen</td>
  </tr>
</table>
```

## So verwendest du das Beispiel

Die style-Angaben sind bereits kleine CSS-Anweisungen. Heute übernehmen wir sie als fertiges Rezept. Später schreiben wir diese Regeln einmal in eine CSS-Datei, statt sie an jeder Zelle zu wiederholen. Das veraltete HTML-Attribut border brauchen wir nicht.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

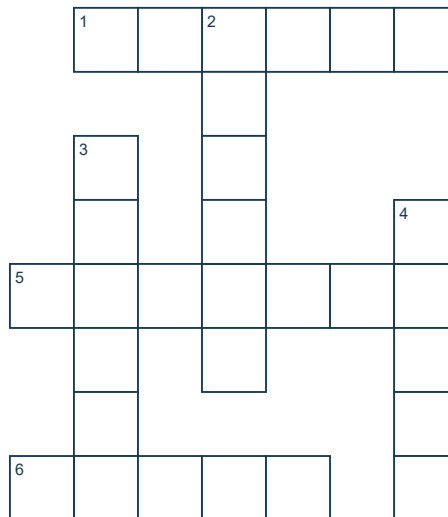
1. Plane auf Papier eine Tabelle zu deinem Hobby. Verwende drei Spalten: zum Beispiel Tag, Aktivität und Dauer. Denke dir vier passende Datenzeilen aus.
2. Schreibe eine h2 Mein Hobbyplan und darunter zwei erklärende Sätze. Beschreibe, ob die Tabelle eine typische Woche oder einen geplanten Ablauf zeigt.
3. Tippe zuerst die Beispieltabelle mit zwei Spalten ab. Ersetze die Inhalte durch deine Angaben. Behalte die Rahmen-Angaben an den Zellen.
4. Ergänze die dritte Spalte: eine zusätzliche th-Zelle in der Kopfzeile und eine zusätzliche td-Zelle in jeder Datenzeile. Schreibe die Dauer mit einer Einheit, zum Beispiel 30 Minuten.
5. Erweitere die Tabelle auf vier Datenzeilen. Jede Zeile soll drei Zellen enthalten. Füge bei jeder neuen Zelle die gleiche border-Angabe ein.
6. Schreibe nach der Tabelle einen Absatz mit drei Sätzen. Erkläre, auf welche Aktivität du dich besonders freust und warum.
7. Speichere und betrachte die Tabelle im Browser. Vergleiche sie mit deiner Zeichnung. Alle Zellen sollen sichtbar umrandet sein und die Angaben unter der richtigen Überschrift stehen.

## Heute entsteht

Ein erklärter Hobbyplan mit Kopfzeile, vier Datenzeilen und drei Spalten. Jede Zelle hat einen sichtbaren Rahmen.

# 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 →

1. CSS-Eigenschaft für die Umrandung. (6 Buchstaben)
5. Sie stellt Angaben in Zeilen und Spalten dar. (7 Buchstaben)
6. Ein einzelnes Feld in der Tabelle. (5 Buchstaben)

## Senkrecht ↓

2. Die sichtbare Linie um ein Feld. (6 Buchstaben)
3. Eine senkrechte Reihe in der Tabelle. (6 Buchstaben)
4. Eine waagrechte Reihe in der Tabelle. (5 Buchstaben)

# Zum Abschluss: Deine Antworten

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

## 1. Beschreibe den Unterschied zwischen Zeile, Spalte und Zelle.

---

---

## 2. Warum reicht ein äußerer Tabellenrahmen nicht für Linien um alle Zellen?

---

---

## 3. Wie ergänzt du eine dritte Spalte?

---

---

## 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-05.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.

# Ein YouTube Video einbetten

## Tagesziel

Ich kann den Einbettungscode eines YouTube-Videos in meine HTML-Seite einsetzen.

Ein eingebettetes Video wird innerhalb deiner Webseite angezeigt. Die Videodatei bleibt dabei auf YouTube. Du lädst sie nicht in deinen Projektordner herunter. Zum Abspielen braucht der Computer Internet.

## Ein Link öffnet eine Seite, ein iframe zeigt sie hier

Ein normaler Videolink führt den Besucher zu YouTube. Ein eingebetteter Player bleibt auf deiner Hobbyseite sichtbar. Dafür verwendest du iframe. Du lädst keine Videodatei herunter. Deine Webseite zeigt nur ein Fenster zum Video auf YouTube.

Die Adresse hinter src muss eine Einbettungsadresse sein. In unserem Beispiel erkennst du sie am Abschnitt /embed/. Die Zeichen dahinter sind die Video-ID. Kopiere den Einbettungscode über Teilen und Einbetten. Eine normale Adresse mit watch?v= funktioniert an dieser Stelle nicht unverändert.

## Das Video braucht auch deinen eigenen Text

Schreibe vor den Player eine passende Überschrift. Erkläre darunter in mehreren Sätzen, was man sieht und warum du dieses Video ausgewählt hast. Nenne auch den Videotitel und den Kanal. Ein Besucher soll verstehen, was das Video mit deinem Hobby zu tun hat.

Teste die Wiedergabe über den Webzugang deiner Lehrkraft. Bei einer direkt geöffneten Datei kann YouTube den Player sperren, obwohl dein HTML richtig ist. Manche Videos dürfen außerdem nicht eingebettet werden. Ergänze deshalb auch einen normalen Link zum Video.

# Bausteine und Beispiel

Ein iframe zeigt einen Inhalt von einer anderen Webseite in einem Rahmen. src enthält die Einbettungsadresse. Bei YouTube steht darin /embed/ und die Video-ID. Ein normaler Link mit watch?v= gehört nicht unverändert in das src eines iframe. title beschreibt den Inhalt des Rahmens.

| Schreibweise                                                  | Bedeutung                                                   |
|---------------------------------------------------------------|-------------------------------------------------------------|
| <code>&lt;iframe&gt; ... &lt;/iframe&gt;</code>               | Ein eingebetteter fremder Inhalt.                           |
| <code>src="https://www.youtube-nocookie.com/embed/..."</code> | YouTube-Einbettungsadresse im erweiterten Datenschutzmodus. |
| <code>title="Video über mein Hobby"</code>                    | Beschreibung des eingebetteten Inhalts.                     |
| <code>width</code> und <code>height</code>                    | Breite und Höhe des Rahmens.                                |
| <code>allowfullscreen</code>                                  | Erlaubt die Vollbildansicht.                                |

## Beispiel im body

```
<h2>Ein Roboter in Bewegung</h2>
<iframe width="560" height="315"
  src="https://www.youtube-nocookie.com/embed/29ECwExc-_M"
  title="All New Atlas von Boston Dynamics"
  referrerpolicy="strict-origin-when-cross-origin"
  allowfullscreen>
</iframe>
<p>Video: Boston Dynamics, All New Atlas.</p>
```

## So verwendest du das Beispiel

So holst du ein anderes Video: Öffne es auf YouTube am Computer. Wähle Teilen → Einbetten → Kopieren. Füge den Code in deinen body ein. Wähle, falls angeboten, den erweiterten Datenschutzmodus. Ergänze einen passenden title. Ein normaler YouTube-Link unter dem Video ist eine gute zusätzliche Möglichkeit zum Öffnen.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

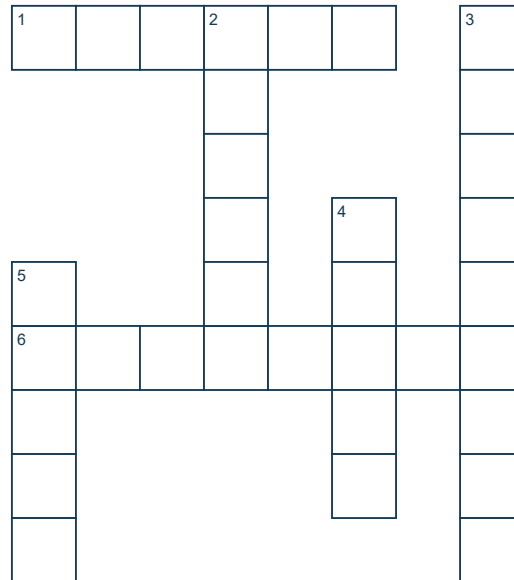
1. Wähle gemeinsam mit der Lehrkraft ein passendes Video oder verwende das Kursbeispiel. Überlege vor dem Einfügen, welchen Teil deines Hobbys es erklärt.
2. Schreibe eine passende h2 und einen kurzen Einleitungstext mit zwei Sätzen. Erkläre darin, worauf die Besucher beim Ansehen achten sollen.
3. Hole den Einbettungscode über Teilen und Einbetten oder tippe das Beispiel ab. Füge den iframe im body unter dem Einleitungstext ein. Ergänze einen passenden title.
4. Schreibe unter dem Player einen Absatz aus drei bis vier eigenen Sätzen. Beschreibe den Inhalt und begründe deine Auswahl. Nenne außerdem den Kanal und den Videotitel.
5. Ergänze einen normalen Link zum Video mit einem verständlichen Linktext. Der Besucher soll das Video bei Bedarf direkt auf YouTube öffnen können.
6. Speichere und teste die Seite über den Webzugang der Lehrkraft. Wenn die Wiedergabe blockiert ist, kläre zuerst, ob die Einbettung erlaubt ist. Deine Erklärung und der Videolink sollen auch dann lesbar bleiben.

## Heute entsteht

Ein eingebettetes Video mit Einleitung, eigener Beschreibung, Quellenangabe und zusätzlichem Videolink.

# 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 →

1. Diesen Wert legt width am iframe fest. (6 Buchstaben)
6. Diese Verbindung brauchst du zur Wiedergabe. (8 Buchstaben)

## Senkrecht ↓

2. HTML-Element für den eingebetteten Player. (6 Buchstaben)
3. So heißt der Menüpunkt nach Teilen bei YouTube. (9 Buchstaben)
4. Diesen Urheberbereich nennst du unter dem Video. (5 Buchstaben)
5. Bewegte Bilder, die du von YouTube einbettetest. (5 Buchstaben)

# Zum Abschluss: Deine Antworten

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

## 1. Wie unterscheidet sich ein Videolink von einem eingebetteten Player?

---

---

## 2. Welche Informationen schreibst du unter das Video?

---

---

## 3. Was machst du, wenn das Video beim Öffnen der lokalen Datei blockiert ist?

---

---

## 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-06.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.

# CSS direkt im HTML

## Tagesziel

Ich kann mit einer style-Angabe die Farbe, Schriftgröße und den Hintergrund ändern.

HTML sagt, was ein Inhalt ist: Absatz, Überschrift, Bild oder Tabelle. CSS sagt, wie er aussieht. CSS direkt im Start-Tag heißt Inline-CSS. Die Rahmen deiner Tabelle und die Unterstreichung haben diese Möglichkeit schon verwendet.

## Eine style-Angabe genau lesen

Im Start-Tag eines Elements kannst du style ergänzen. Die ganze Gestaltung steht zwischen Anführungszeichen. In style="color: navy;" sagt color, was du ändern möchtest: die Textfarbe. navy sagt, welche Farbe du möchtest: dunkelblau. Der Doppelpunkt verbindet Eigenschaft und Wert.

Nach einer Angabe folgt ein Semikolon. Danach kann die nächste Angabe kommen, zum Beispiel padding: 12px;. Schreibe alles innerhalb desselben style-Attributs. Die HTML-Tags und dein Text bleiben erhalten. Du ergänzt nur die Gestaltung des Elements.

## Langsam verändern und vergleichen

Ändere zuerst nur eine Eigenschaft. Speichere und sieh dir die Seite an. Ändere danach den nächsten Wert. So erkennst du, welche Angabe welche Wirkung hat. padding ist zum Beispiel der freie Platz zwischen dem Text und dem Rand seines Bereichs.

Eine style-Angabe an einem Absatz betrifft diesen Absatz. Eine Schriftart am body wird normalerweise auch für seine Texte verwendet. Ergänze die Angaben am vorhandenen body-Start-Tag. Ein zweiter body ist dafür nicht nötig. Wähle dunkle Schrift auf einem hellen Hintergrund.

# Bausteine und Beispiel

Im Attribut style stehen Eigenschaft und Wert. In color: navy; ist color die Eigenschaft und navy der Wert. Der Doppelpunkt trennt beide. Das Semikolon beendet die Angabe. Mehrere Angaben dürfen in derselben style-Zeichenkette stehen.

| Schreibweise                    | Bedeutung                     |
|---------------------------------|-------------------------------|
| color: navy;                    | Textfarbe dunkelblau.         |
| background-color: lightyellow;  | Hintergrundfarbe hellgelb.    |
| font-size: 24px;                | Schriftgröße 24 Pixel.        |
| font-family: Arial, sans-serif; | Schriftart und Ersatzschrift. |
| text-align: center;             | Text zentrieren.              |
| padding: 12px;                  | Innenabstand zum Rand.        |

## Inline-CSS an einzelnen Elementen

```
<h1 style="color: navy;">Mein Hobby Robotik</h1>
```

```
<p style="font-size: 24px;">
  Ich baue kleine Roboter und probiere neue Ideen aus.
  Besonders gern teste ich, wie sie sich bewegen.
</p>
```

```
<p style="background-color: lightyellow; padding: 12px;">
  Beginne mit einer einfachen Aufgabe.
  Plane zuerst, was dein Roboter können soll.
  Teste danach immer nur eine Änderung auf einmal.
</p>
```

## So verwendest du das Beispiel

Für die ganze Seite kannst du den vorhandenen body-Start-Tag erweitern: `<body style="font-family: Arial, sans-serif; background-color: lightblue;">`. Erstelle keinen zweiten body. Wähle Farben, bei denen die Schrift gut lesbar bleibt.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

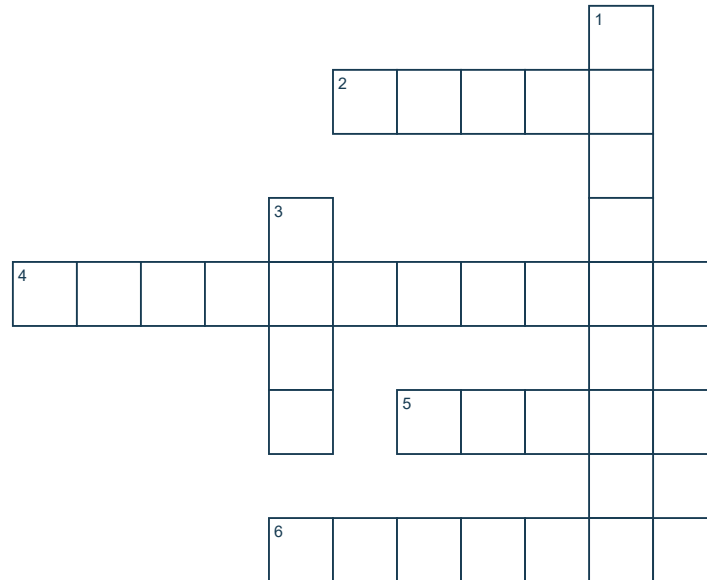
1. Öffne deine Hobbyseite und wähle zwei zusammenpassende Farben: eine dunkle Schriftfarbe und eine helle Hintergrundfarbe. Verwende zunächst Farbnamen aus dem Beispiel.
2. Ergänze `style` an deiner `h1` und ändere ihre Farbe. Speichere und betrachte nur diese erste Änderung im Browser, bevor du weiterarbeitest.
3. Wähle den ersten Einführungsabsatz und setze `font-size` auf 24px. Vergleiche ihn mit den anderen Absätzen. Ändere den Wert danach einmal auf 20px und entscheide, was besser passt.
4. Schreibe einen neuen Hinweis für Anfänger mit mindestens drei Sätzen. Gib diesem Absatz einen hellgelben Hintergrund und `padding: 12px;`. So wird aus deinem Text ein gut sichtbarer Hinweisbereich.
5. Ändere am vorhandenen `body`-Start-Tag die Schriftart auf Arial und die Hintergrundfarbe auf einen hellen Farbton. Behalte alle vorhandenen Inhalte im `body`.
6. Ergänze `text-align: center;` an einer passenden Überschrift. Lass längere Absätze linksbündig. Betrachte, wie sich die Lesbarkeit verändert.
7. Lies deine Seite im Browser. Verbessere Farben, bei denen die Schrift schlecht zu erkennen ist. Beschreibe in einem neuen Absatz mit zwei Sätzen, warum du deine Farben gewählt hast.

## Heute entsteht

Eine Seite mit bewusst gewählten Farben und einem hervorgehobenen Absatz.

# 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 →

2. Diese Eigenschaft des Textes ändert color. (5 Buchstaben)
4. Dieses Zeichen trennt Eigenschaft und Wert. (11 Buchstaben)
5. In diesem Attribut steht Inline-CSS. (5 Buchstaben)
6. CSS-Eigenschaft für den Innenabstand. (7 Buchstaben)

## Senkrecht ↓

1. Dieses Zeichen beendet eine CSS-Angabe. (9 Buchstaben)
3. In color: navy; ist navy der ... (4 Buchstaben)

# Zum Abschluss: Deine Antworten

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

## 1. Erkläre color: navy; mit den Wörtern Eigenschaft und Wert.

---

---

## 2. Was verändert padding an deinem Hinweisabsatz?

---

---

## 3. Wie wählst du gut lesbare Farben?

---

---

## 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-07.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.

# Eine eigene CSS Datei

## Tagesziel

Ich kann style.css erstellen und mit meiner HTML-Datei verknüpfen.

Wenn du viele Absätze gleich gestalten willst, wird Inline-CSS schnell unübersichtlich. Deshalb sammeln wir die Gestaltung in einer eigenen Datei. Sie heißt style.css und liegt neben index.html.

## Zwei Dateien gehören jetzt zusammen

Bisher hast du Gestaltung direkt an mehrere HTML-Tags geschrieben. Jetzt sammelst du gemeinsame Regeln in style.css. Beide Dateien liegen im selben Ordner: index.html enthält den Inhalt, style.css die Gestaltung. Du kannst beide in VS Code gleichzeitig als Tabs geöffnet lassen.

Die Verbindung steht im head der HTML-Datei: `<link rel="stylesheet" href="style.css">`. href nennt hier deine CSS-Datei. Ohne diese Verbindung kennt der Browser deine Regeln nicht. Speichere deshalb sowohl die HTML-Datei als auch die CSS-Datei.

## Eine Regel gilt für mehrere Elemente

In `h1 { color: navy; }` steht h1 für die Elemente, die gestaltet werden. Zwischen den geschweiften Klammern stehen die Angaben. Eine Regel für p gilt für alle Absätze. Bei th, td trennt das Komma zwei Elementnamen: Dieselbe Rahmenregel gilt für beide Arten von Tabellenzellen.

In style.css schreibst du keine HTML-Tags und kein `style="..."` um die Regeln. Entferne passende alte Inline-Angaben erst, wenn die Verbindung funktioniert. Eine alte Farbe direkt an h1 kann sonst die neue Farbe aus der Datei überdecken. Die Unterstreichung aus Modul 3 darf vorerst bleiben.

# Bausteine und Beispiel

Ein link-Element im head verbindet beide Dateien. In der CSS-Datei steht zum Beispiel `p { color: navy; }`. `p` wählt alle Absätze aus. Die geschweiften Klammern umschließen die Angaben. In `style.css` stehen keine HTML-Tags und auch keine `<style>`-Tags.

| Schreibweise                                                | Bedeutung                                         |
|-------------------------------------------------------------|---------------------------------------------------|
| <code>&lt;link rel="stylesheet" href="style.css"&gt;</code> | Verknüpfung im head.                              |
| <code>p { color: navy; }</code>                             | Alle Absätze dunkelblau gestalten.                |
| <code>h1 { color: darkgreen; }</code>                       | Alle h1-Überschriften gestalten.                  |
| <code>th, td { border: 1px solid black; }</code>            | Überschrifts- und Datenzellen gemeinsam umranden. |

## Diese Regeln stehen in style.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #eef4f8;
  color: #202b33;
}
h1 { color: navy; }
table { border-collapse: collapse; }
th, td {
  border: 1px solid black;
  padding: 8px;
}
```

## So verwendest du das Beispiel

Entferne alte Inline-Angaben, die du jetzt in `style.css` regelst. Sonst können sie weiterhin Vorrang haben. Beispiel: Ein `style="color: red;"` direkt an `h1` schlägt die normale externe `h1`-Farbregel. Den Rahmen der Tabelle kannst du jetzt zentral ändern.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

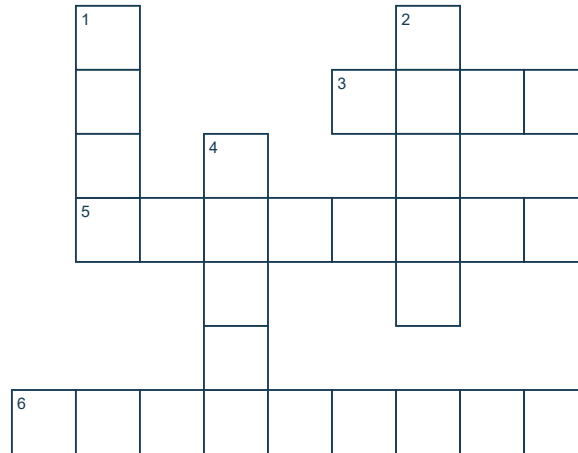
1. Erstelle style.css im selben Ordner wie index.html. Öffne beide Dateien in VS Code und schreibe die link-Verknüpfung in den head der HTML-Datei.
2. Beginne in style.css mit einer h1-Regel für eine deutlich erkennbare Farbe. Entferne die alte Farbangabe direkt an h1. Speichere beide Dateien und prüfe im Browser, ob die Verbindung funktioniert.
3. Übernimm danach die body-Regel für Schrift und Hintergrund. Entferne nur die entsprechenden Inline-Angaben am body. Dein Text und die HTML-Tags bleiben erhalten.
4. Übertrage die Tabellenregeln. Entferne die border-Angaben an th und td sowie border-collapse am table-Start-Tag. Schreibe die gemeinsamen Regeln einmal in style.css.
5. Ändere den Rahmen aller Tabellenzellen über die gemeinsame Regel auf dunkelblau. Ergänze padding: 8px; bei th, td, damit der Text nicht direkt an der Linie steht.
6. Schreibe eine h2-Regel mit einer passenden Farbe. Sie soll alle Unterüberschriften gemeinsam gestalten. Schau im Browser nach, ob tatsächlich mehrere Überschriften betroffen sind.
7. Ergänze einen Absatz mit drei Sätzen über deine bisherige Arbeit an der Hobbyseite. Speichere beide Dateien. Die Seite soll ihren Inhalt behalten und ihre gemeinsame Gestaltung jetzt aus style.css bekommen.

## Heute entsteht

Die HTML-Datei enthält die Inhalte; style.css gestaltet die Seite und die Tabellenrahmen.

# 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 steht die Stylesheet-Verknüpfung. (4 Buchstaben)
5. Diese geschweiften Zeichen umschließen die CSS-Angaben. (8 Buchstaben)
6. Das musst du nach Änderungen an beiden Dateien tun. (9 Buchstaben)

## Senkrecht ↓

1. Dieses Element verbindet das HTML mit der CSS-Datei. (4 Buchstaben)
2. Sie besteht zum Beispiel aus h1 und Angaben in geschweiften Klammern. (5 Buchstaben)
4. style.css ist eine eigene ... (5 Buchstaben)

# Zum Abschluss: Deine Antworten

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

## 1. Wie verbindest du index.html mit style.css?

---

---

## 2. Warum ist eine gemeinsame Regel für Tabellenrahmen praktisch?

---

---

## 3. Was prüfst du, wenn eine neue externe Farbe nicht erscheint?

---

---

## 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-08.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.

# Meine Hobbyseite fertig gestalten

## Tagesziel

Ich kann ein Hintergrundbild verwenden und meine bisherigen Inhalte zu einer Hobbyseite zusammenführen.

Deine Hobbyseite enthält bereits Texte, ein Bild, eine Tabelle und ein Video. Jetzt machst du daraus eine zusammenhängende Seite. Das Thema bestimmst du. Eine einzelne gut lesbare Seite reicht.

## Deine Seite erzählt von einem Hobby

Stell dir vor, eine Person liest deine Seite und kennt dich nicht. Sie möchte wissen: Was machst du? Was brauchst du dafür? Was möchtest du als Nächstes lernen? Ordne deine vorhandenen Texte in dieser Reihenfolge. Verbinde kurze Einzelinformationen zu verständlichen Absätzen.

Bild, Tabelle und Video sollen zu diesem Thema passen. Erkläre jedes dieser Elemente mit eigenem Text. Ein Bild kann deine Ausrüstung zeigen, eine Tabelle deinen Wochenplan und ein Video eine interessante Technik. Du arbeitest an deiner bestehenden Seite weiter.

## Ein ruhiger Platz für den Text

Das Hintergrundbild liegt hinter den Inhalten. Auf einem unruhigen Foto ist Schrift oft schwer lesbar. Deshalb bekommt dein Hauptinhalt mit `main` eine weiße Fläche. Setze `<main>` direkt vor deine erste Überschrift und `</main>` nach dem letzten Inhalt, noch vor `</body>`.

`max-width` begrenzt die Breite der Lesefläche. `padding` lässt innen Platz um den Text. `margin: 24px auto;` lässt oben und unten Abstand und verteilt freien Platz links und rechts gleichmäßig. Das Bild im Hintergrund darf am Rand sichtbar bleiben. Wenn `cover` Teile davon abschneidet, ist das normal.

# Bausteine und Beispiel

Ein normales Bild ist Inhalt und wird mit `img` eingefügt. Ein Hintergrundbild ist Gestaltung und wird in CSS mit `background-image` gesetzt. `url(...)` nennt die Datei. Der Pfad wird von der CSS-Datei aus gelesen. Unsere `style.css` liegt neben dem Ordner `bilder`.

| Schreibweise                                                  | Bedeutung                                                                           |
|---------------------------------------------------------------|-------------------------------------------------------------------------------------|
| <code>background-image: url("bilder/hintergrund.svg");</code> | Bild als Hintergrund verwenden.                                                     |
| <code>background-size: cover;</code>                          | Bild so vergrößern, dass es den Bereich abdeckt; Teile können abgeschnitten werden. |
| <code>background-position: center;</code>                     | Hintergrundbild zentrieren.                                                         |
| <code>background-repeat: no-repeat;</code>                    | Hintergrundbild nicht kacheln.                                                      |
| <code>main</code>                                             | Ein Bereich für den Hauptinhalt.                                                    |
| <code>max-width</code> und <code>padding</code>               | Breite begrenzen und innen Abstand lassen.                                          |

## Ergänzung in `style.css`

```
body {
  background-image: url("bilder/hintergrund.svg");
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
}
main {
  max-width: 800px;
  margin: 24px auto;
  padding: 20px;
  background-color: white;
}
img { max-width: 100%; height: auto; }
iframe { max-width: 100%; border: 0; }
```

## So verwendest du das Beispiel

Schreibe im `body` einmal `<main>` vor deine Inhalte und `</main>` nach deine Inhalte. Das erzeugt mit unseren CSS-Regeln eine weiße Lesefläche vor dem Hintergrundbild. Das Hintergrundbild darf auch ein eigenes Foto sein. Passe dann den Dateinamen in `url(...)` an.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

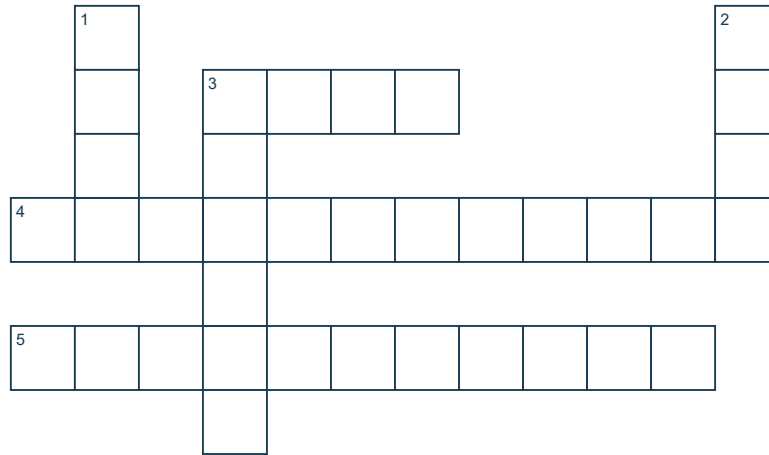
1. Lies deine ganze Hobbyseite und ordne die Abschnitte. Beginne mit einer Einführung, beschreibe anschließend Ausrüstung und Aktivitäten und schließe mit einem eigenen Ziel ab.
2. Überarbeite die Texte zu mindestens fünf eigenen Absätzen mit jeweils zwei bis vier Sätzen. Erkläre unbekannte Begriffe einfach. Behalte eine h1 und mindestens zwei passende h2-Überschriften.
3. Setze einmal main um deinen gesamten sichtbaren Inhalt. Überschriften, Texte, Listen, Bild, Tabelle und Video sollen innerhalb dieses Bereichs stehen.
4. Kopiere das Kurs-Hintergrundbild in bilder oder verwende ein eigenes geeignetes Foto. Ergänze background-image und die weiteren Hintergrundregeln in style.css. Achte auf den richtigen Dateinamen.
5. Gestalte main mit der weißen Fläche, der begrenzten Breite und dem Innenabstand aus dem Beispiel. Ergänze die Regeln für img und iframe, damit diese nicht breiter als ihre Umgebung werden.
6. Baue Bild, Tabelle und Video an passenden Stellen ein. Erkläre sie jeweils mit eigenem Text. Deine Tabelle soll eine Kopfzeile, mindestens vier Datenzeilen und sichtbare Zellrahmen haben.
7. Sieh dir die Seite in einem breiten und in einem schmalen Browserfenster an. Verbessere Stellen, an denen Text schwer lesbar ist. Behalte deine Textgestaltung und alle bisherigen Inhalte, die zu deinem Thema passen.

## Heute entsteht

Eine eigene Hobbyseite mit Textgestaltung, Bild, Hintergrundbild, Tabelle und Video.

# 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. Es kann Inhalt oder Gestaltung sein. (4 Buchstaben)
- 4. Deutscher Begriff für padding. (12 Buchstaben)
- 5. Dort liegt das Bild aus background-image. (11 Buchstaben)

## Senkrecht ↓

- 1. Element für unseren Hauptinhalt. (4 Buchstaben)
- 2. Er gibt den Weg zur Hintergrunddatei an. (4 Buchstaben)
- 3. max-width begrenzt diese Abmessung. (6 Buchstaben)

# Zum Abschluss: Deine Antworten

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

## 1. Was unterscheidet ein img-Bild vom Hintergrundbild?

---

---

## 2. Warum verwenden wir main mit einer weißen Fläche?

---

---

## 3. Worauf achtest du bei einem schmalen Browserfenster?

---

---

## 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-09.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.

# Klassen ganz einfach

## Tagesziel

Ich kann mit einer Klasse einzelne Absätze gleich gestalten und meine Hobbyseite fertig abgeben.

Die Regel `p` gestaltet alle Absätze. Manchmal sollen aber nur einzelne Absätze wie ein Tipp aussehen. Dafür bekommen diese Absätze einen gemeinsamen Namen: eine Klasse. Wir verwenden heute nur eine Klasse namens `tipp`.

## Eine Klasse ist ein gemeinsamer Name

Du hast viele Absätze, aber nur zwei sollen als Tipps auffallen. Gib genau diesen Absätzen im Start-Tag `class="tipp"`. Die anderen Absätze bekommen diese Klasse nicht. Eine Klasse kann also eine kleine Gruppe von Elementen auswählen.

In `style.css` beginnt der Klassenname mit einem Punkt: `.tipp`. Im HTML steht nur `tipp` zwischen den Anführungszeichen. Schreibe die Namen genau gleich. Wenn du in der CSS-Regel die Farbe änderst, ändern sich beide Tippabsätze gleichzeitig.

## Der Inhalt zeigt, ob du es verstanden hast

Schreibe hilfreiche Tipps für jemanden, der dein Hobby ausprobieren möchte. Beginne zum Beispiel mit `Nimm dir am Anfang ...` oder `Achte besonders auf ....` Ein Tipp darf zwei oder drei Sätze lang sein. Der gemeinsame Hintergrund zeigt, dass diese Texte zusammengehören.

Am Ende gibst du den ganzen Projektordner ab. Nur `index.html` reicht nicht, denn die Seite braucht auch `style.css` und die Bilder. Zeige anschließend an einem eigenen Beispiel, welcher Teil Inhalt und welcher Teil Gestaltung ist. Du musst dafür keine Begriffe auswendig aufsagen.

# Bausteine und Beispiel

Im HTML steht `class="tipp"`. In CSS steht davor ein Punkt: `.tipp`. Du darfst dieselbe Klasse bei mehreren Absätzen verwenden. Der Punkt gehört zum CSS-Selektor, nicht in das `class`-Attribut.

| Schreibweise                                       | Bedeutung                                           |
|----------------------------------------------------|-----------------------------------------------------|
| <code>&lt;p class="tipp"&gt; ... &lt;/p&gt;</code> | Dieser Absatz gehört zur Klasse <code>tipp</code> . |
| <code>.tipp { ... }</code>                         | CSS-Regel für die Klasse <code>tipp</code> .        |
| <code>background-color</code>                      | Hintergrund des Tippabsatzes.                       |
| <code>padding</code>                               | Abstand zwischen Text und Rand.                     |
| <code>border</code>                                | Sichtbarer Rahmen um den Tipp.                      |

## Die Klasse im HTML verwenden

```
<!-- In index.html im body -->
<p class="tipp">Beginne mit einem kleinen Projekt.</p>
<p>Dieser Absatz bleibt ein normaler Absatz.</p>
<p class="tipp">Notiere deine nächste Idee.</p>
```

## Die Regel in style.css

```
.tipp {
  background-color: lightyellow;
  padding: 12px;
  border: 1px solid goldenrod;
}
```

## So verwendest du das Beispiel

Die zweite Codebox kommt in `style.css`. Wenn du dort die Hintergrundfarbe änderst, verändern sich beide Tippabsätze gleichzeitig. Der normale Absatz erhält diese Regel nicht.

# Deine Aufgabe

Arbeite in deinem Ordner mein-hobby weiter. Bearbeite die Schritte der Reihe nach. Verwende dein eigenes Hobby und formuliere eigene Sätze. Speichere nach jedem größeren Arbeitsschritt und sieh dir das Ergebnis im Browser an.

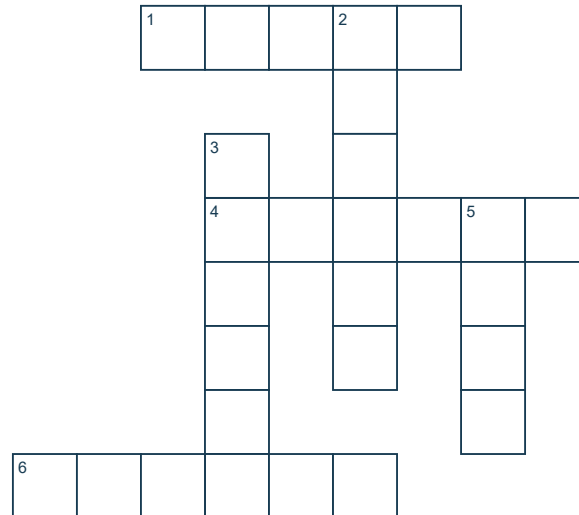
1. Schreibe zwei hilfreiche Tipps für Anfänger in deinem Hobby. Jeder Tipp soll aus zwei bis drei Sätzen bestehen und einen konkreten Rat erklären.
2. Gib genau diesen beiden Absätzen `class="tipp"`. Lass mindestens einen normalen Absatz zwischen ihnen stehen, damit du die unterschiedliche Gestaltung gut erkennen kannst.
3. Ergänze die `.tipp`-Regel in `style.css`. Speichere beide Dateien und betrachte die Seite. Nur die beiden Tippabsätze sollen diesen Hintergrund und Rahmen erhalten.
4. Ändere die Hintergrundfarbe in der gemeinsamen Regel einmal auf `lightblue`. Beobachte beide Tipps und wähle danach deine endgültige Farbe. Entferne alte `Inline`-Farben an den Tipps, falls sie die Regel verdecken.
5. Stelle deine Hobbyseite fertig. Sie soll mindestens fünf eigene Absätze, passende Überschriften, gestaltete Wörter, Listen, einen Link, ein Bild, ein Hintergrundbild, ein beschriebenes Video und eine Tabelle mit vier Datenzeilen enthalten.
6. Ergänze einen abschließenden Absatz mit drei Sätzen: Was gefällt dir an deiner fertigen Seite? Was hast du beim Erstellen gelernt? Was würdest du später noch ergänzen?
7. Gib den vollständigen Ordner mein-hobby mit `index.html`, `style.css` und `bilder` ab. Zeige deine Seite kurz. Erkläre an deinem Code einen Absatz, den Bildpfad und die gemeinsame Gestaltung der beiden Tipps.

## Heute entsteht

Eine fertig abgegebene Hobbyseite mit ausführlichen eigenen Texten und zwei gemeinsam gestalteten Tippabsätzen.

# 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 →

1. Dieses Zeichen steht vor tipp im CSS. (5 Buchstaben)
4. Das p-Element bildet diesen Textbereich. (6 Buchstaben)
6. Diesen gibst du mit allen Projektdaten ab. (6 Buchstaben)

## Senkrecht ↓

2. Gemeinsamer Name für ausgewählte Elemente. (6 Buchstaben)
3. Diesen erzeugt border um den Tipp. (6 Buchstaben)
5. So heißt die verwendete Klasse im Kurs. (4 Buchstaben)

# Zum Abschluss: Deine Antworten

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

## 1. Schreibe ein HTML-Beispiel für die Klasse `tipp` und den passenden CSS-Selektor.

---

---

## 2. Warum ändern sich zwei Tippabsätze durch dieselbe Regel?

---

---

## 3. Welche Dateien müssen bei der Abgabe mitkommen?

---

---

## 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-10.html. Keine Anmeldung nötig. Ergebnisse werden nicht an die Lehrkraft übertragen.