



Webseiten mit HTML und CSS gestalten

Vom ersten HTML-Grundgerüst zur responsiven Oberfläche eines persönlichen Lernagenten

Seiten strukturieren · Layout gestalten · responsiv prüfen

BKW11 · BPE 4 · 40 Stunden Pflichtpfad und Leistungsnachweis
Didaktisch erweiterte Kursfassung · A4 · Schuljahr 2026/2027 · v1.1 · EDUCODE Serienmaster R7

Christine Janischek · Lizenz Custom License NC School-Only 1.0 · emotionalspirit.de

Dieses Lernskript wurde mit Unterstützung einer KI erstellt und anschließend redaktionell geprüft. Trotz sorgfältiger Bearbeitung können Fehler enthalten sein. Bitte melde Fehler und Verbesserungsvorschläge über die angegebene Kontaktmöglichkeit.

Inhaltsverzeichnis

Die Einträge führen direkt zu den Kapiteln und Hauptabschnitten. In den Überschriften bringt dich „↩ Inhalt“ hierher zurück.

- 01 So arbeitest du mit diesem Skript**
- 02 Planungsbezug**
- 03 Gesamtstruktur des Lernpfades**
- 04 Woche 8: Dein Arbeitsplan für drei Doppelstunden**
- 05 Fahrplan für die Wochen 9 bis 14**
- 06 Eine Website erkunden und das Web verstehen**
- 07 Projektstruktur, Dateinamen und relative Pfade**
- 08 Das HTML-Grundgerüst und erste Inhalte**
- 09 Semantische Seitenstruktur**
- 10 Listen, Links und Navigation**
- 11 HTML prüfen und erste Barrieren beseitigen**
- 12 CSS einbinden, auswählen und anwenden**
- 13 Typografie, Farben und Corporate Design**
- 14 Box-Modell und relative Größen**
- 15 Eindimensionale Layouts mit Flexbox**
- 16 Zweidimensionale Layouts mit CSS Grid**
- 17 Flexbox und Grid sinnvoll kombinieren**
- 18 Responsive Design entwickeln und testen**
- 19 Tabellarische Daten semantisch darstellen**
- 20 Bilder und Medien zugänglich einbinden**
- 21 Mediengalerie und dezente Effekte**
- 22 Barrierefreiheit, Quellen und rechtliche Mindestprüfung**
- 23 Abschlussprojekt: Oberfläche eines persönlichen Lernagenten**
- 24 Verbindliche Prüfungsübersicht**
- 25 Quellen und Arbeitsgrundlagen**
- 26 Anhang A: Kompetenz-, Aufgaben- und Materialmatrix**
- 27 Anhang B: Verbindliche Mockup-Routine**
- 28 Anhang C: Zeitmodell**

So arbeitest du mit diesem Skript [↩ Inhalt](#)

Jedes Modul führt von einer verständlichen Ausgangssituation zu einem sichtbaren und überprüfbaren Teilprodukt. Du versuchst zunächst selbst eine Lösung, nutzt Hilfen nur schrittweise und erklärst anschließend deinen Lösungsweg.

Wiederkehrende Strategie: VERSTEHEN → VERSUCHEN → PRÜFEN → ERKLÄREN → ÜBERTRAGEN

Mockup-Regel: Erst sichtbare Anforderungen untersuchen, dann Struktur und Layout planen, erst danach HTML und CSS schreiben. Jede Umsetzung endet mit einem Soll-Ist-Vergleich.

KI-Regel: Der Lernagent stellt Rückfragen, gibt gestufte Hinweise und prüft Begründungen. Eine fertige Lösung wird erst nach einem eigenen dokumentierten Versuch betrachtet.

Baustein	Bedeutung
Ziel	Was du am Ende erklären, anwenden oder beurteilen kannst.
Jetzt tun	Ein kurzer eigener Denk- oder Gestaltungsversuch vor der Erklärung.
Startprompt	Die KI stellt Rückfragen und begleitet dein Denken, ohne die fertige Lösung vorzugeben.
Grundlagen	Verbindliche Anforderungen des Bildungsplans.
Vertiefung	Transfer, Fehlersuche, Begründung und selbstständige Gestaltung.
Prüfnachweis	Kriterien, mit denen du dein Ergebnis kontrollierst.
Learning-Agent-Transfer	Ein wiederverwendbarer Bestandteil für den späteren Lernagenten.

REGEL FÜR DIE KI-NUTZUNG

Nutze die KI als Lerncoach: Lass dir Begriffe erklären, fordere Rückfragen und Hinweise an und prüfe jede Aussage. Übernimm keinen Quellcode, den du nicht selbst erklären und testen kannst.

OFFLINE ZUERST

Für die ersten Lernschritte benötigst du nur das Offline-Lernpaket, einen Texteditor und einen Browser. GitHub ist nicht erforderlich. Die Startprompts kannst du freiwillig in ChatGPT oder einen vergleichbaren KI-Assistenten einfügen. Ohne KI nutzt du dieselben Fragen als Partnergespräch und anschließend die gestuften Tutorhilfen.

Planungsbezug [← Inhalt](#)

Das Skript deckt BPE 4 Webentwicklung des Bildungsplans für das Berufskolleg Wirtschaftsinformatik vom 13. März 2025 ab. Der Zeitrichtwert beträgt 40 Unterrichtsstunden. Der BKWI1-Wochenplan 2026/2027 sieht dafür die Unterrichtswochen 8 bis 14 bei sechs Wochenstunden vor.

Woche	Modul	Schwerpunkt	Richtzeit
8	1-3	Web verstehen; Dateien und Pfade; HTML-Grundgerüst und erste Inhalte	6 UStd.
9	4-6	Semantische Seitenstruktur; Navigation; HTML validieren	6 UStd.
10	7-9	CSS-Grundlagen; Typografie, Farben und Box-Modell	6 UStd.
11	10-12	Corporate Design; Flexbox; CSS Grid	6 UStd.
12	13	Responsive Design; Übung und Leistungsfeststellung	6 UStd.
13	14-16	Tabellen; Bilder und Medien; Galerie und Effekte	6 UStd.
14	17-18	Barrierefreiheit; Recht; Projektabschluss und Reflexion	6 UStd.

Gesamtstruktur des Lernpfades [← Inhalt](#)

Teil A · Orientierung und HTML-Grundlagen

Web, Dateien, Pfade, HTML-Grundgerüst und grundlegende Inhaltselemente.

Teil B · Semantische und prüfbare Webseiten

Seitenstruktur, Navigation, Links, Tabellen, Medien und Validierung.

Teil C · Gestaltung mit CSS

Selektoren, Kaskade, Typografie, Farben, Box-Modell und Linkgestaltung.

Teil D · Flexible Layouts

Flexbox, CSS Grid, responsive Gestaltung und begründete Layoutentscheidungen.

Teil E · Zielgruppe, Qualität und Recht

Corporate Design, Barrierefreiheit, Bildrechte, Impressum und Datenschutz.

Teil F · Abschlussprojekt

Statische Oberfläche des persönlichen Lernagenten planen, umsetzen, prüfen und dokumentieren.

Woche 8: Dein Arbeitsplan für drei Doppelstunden [↔ Inhalt](#)

Jede Doppelstunde umfasst 90 Minuten beziehungsweise zwei Unterrichtsstunden. Arbeite zuerst am Pflichtteil. Hausarbeit entsteht nur, wenn der Pflichtteil noch offen ist. Wer früher fertig ist, bearbeitet die Vertiefung.

Doppelstunde	Pflichtziel bis zum Ende	Arbeitsphasen	Zu Hause	Vertiefung
1 · Orientierung	Fünf Grundbegriffe in eigenen Worten; Startprojekt im Browser und Editor untersucht; Bestandsaufnahme begonnen.	15 Min Einstieg · 25 Min Erkunden · 30 Min Grundlagen · 15 Min Sicherung · 5 Min Check	Pflichtteil fertigstellen, höchstens 15-20 Min.	Zwei Lernwebseiten vergleichen.
2 · Pfade A	Fehlerursachen bei Stylesheet und Logo erklärt; bereinigte Namen und vier Pfade entwickelt.	10 Min Fehlersuche · 20 Min Denkmodell · 40 Min Reparatur · 15 Min Test · 5 Min Check	Fehlerprotokoll fertigstellen, höchstens 15 Min.	Download-Ordner und weitere Pfade planen.
3 · HTML-Start	HTML-Grundgerüst erstellt; Überschriften, Absätze und Kommentare eingesetzt; Seite geprüft.	10 Min Rückblick · 20 Min Fachwissen · 40 Min Aufbau · 15 Min Test · 5 Min Check	Nur noch offenen Pflichtteil abschließen.	Zusätzlichen Abschnitt mit sinnvoller Überschriftenfolge ergänzen.

PFLICHTWISSEN FÜR DEN WOCHEN- UND KLASSENARBEITSHECK

Browser, Website und Webseite unterscheiden · Aufgaben von HTML und CSS erklären · Bedeutung von index.html kennen · zuverlässige Datei- und Ordernamen bilden · relative Pfade im selben Ordner, in Unterordner und mit ../ bilden · relative und absolute Verweise unterscheiden · defekte Verweise systematisch testen und einen 404-Hinweis einordnen.

DEIN WOCHENCHECKPOINT

Am Ende der dritten Doppelstunde zeigst du: ein lokal funktionierendes Projekt, deine Bestandsaufnahme, vier begründete Pfade, ein kurzes Fehler- und Testprotokoll sowie dein erstes eigenes HTML-Dokument. Diese Ergebnisse bilden den verbindlichen Lernstand.

Fahrplan für die Wochen 9 bis 14 [↔ Inhalt](#)

Auch in den folgenden Wochen gilt: Der Pflichtteil definiert den gemeinsamen Lernstand. Hausarbeit dient nur dem Abschluss begonnener Pflichtaufgaben. Vertiefungen sind zusätzliche Lernangebote und werden nicht vorausgesetzt.

Woche 9 · Semantische und prüfbare Webseiten

Doppelstunde	Pflichtziel	Zu Hause	Vertiefung
1 · Modul 4	header, nav, main, section, aside und footer semantisch einsetzen	Pflichtteil abschließen	zusätzliche Inhaltsseite
2 · Modul 5	Listen, interne/externe Links und zugängliche Navigation erstellen	Linktest vervollständigen	aktive Seite kennzeichnen
3 · Modul 6	HTML validieren und grundlegende Barrieren finden und verbessern	Prüfprotokoll abschließen	Tastaturtest vertiefen

Woche 10 · Gestaltung mit CSS

Doppelstunde	Pflichtziel	Zu Hause	Vertiefung
1 · Modul 7	Stylesheet einbinden; Selektoren, Kaskade und Vererbung anwenden	Pflichtteil abschließen	Spezifität untersuchen
2 · Modul 8	Typografie, Farben, Kontrast und kleines Corporate Design entwickeln	Designregeln dokumentieren	zweite Designvariante
3 · Modul 9	Box-Modell und relative Maße für robuste Komponenten einsetzen	Box-Analyse abschließen	Komponente skalierbar machen

Woche 11 · Flexible Layouts

Doppelstunde	Pflichtziel	Zu Hause	Vertiefung
1 · Modul 10	eindimensionales Layout mit Flexbox planen und umsetzen	Pflichtteil abschließen	Footer oder Kartenleiste
2 · Modul 11	zweidimensionales Layout mit CSS Grid umsetzen	Raster dokumentieren	grid-template-areas
3 · Modul 12	Flexbox und Grid zielgerichtet kombinieren und begründen	Testprotokoll abschließen	alternative Layoutlösung

Woche 12 · Responsive Design und Leistungsnachweis

Doppelstunde	Pflichtziel	Zu Hause	Vertiefung
1 · Modul 13A	Mobile-first-Grundlayout und fließende Größen erstellen	Pflichtteil abschließen	clamp() untersuchen
2 · Modul 13B	Media Query ergänzen und drei Ansichten systematisch testen	Fehler beheben	zusätzlicher Breakpoint
3 · Modul 13C	verbindlichen Praxischeck zu HTML, CSS und Layout bearbeiten	keine reguläre Hausarbeit	eigene Verbesserung begründen

Woche 13 · Daten und Medien

Doppelstunde	Pflichtziel	Zu Hause	Vertiefung
1 · Modul 14	echte Tabellendaten semantisch und responsiv darstellen	Pflichtteil abschließen	caption und scope vertiefen
2 · Modul 15	Bilder und Medien passend auswählen, optimieren und zugänglich einbinden	Mediennachweis ergänzen	picture oder srcset
3 · Modul 16	zugängliche Mediengalerie mit dezenten Effekten gestalten	Galerie fertigstellen	reduzierte Bewegung beachten

Woche 14 · Qualität, Recht und Abschluss

Doppelstunde	Pflichtziel	Zu Hause	Vertiefung
1 · Modul 17	Barrierefreiheit, Quellen, Bildrechte, Impressum und Datenschutz prüfen	Nachweise ergänzen	Prüfraster erweitern
2 · Modul 18A	Lernagenten-Oberfläche planen und funktionsfähigen Prototyp umsetzen	nur offenen Pflichtteil	zusätzliche Komponente
3 · Modul 18B	testen, überarbeiten, dokumentieren, vorstellen und reflektieren	keine reguläre Hausarbeit	zweite responsive Variante

Modul 1: Eine Website erkunden und das Web verstehen [← Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst Browser, Webseite, Website, HTML und CSS voneinander unterscheiden und die Aufgaben der wichtigsten Projektdateien erklären.

Situation und jetzt tun

Zielansicht: So könnte der Lernagent aussehen



Abbildung 1: Startseite des Tierwelten-Lernagenten als visueller Impuls.

Mockup-Check 1/4 – möglicher Zielentwurf. Markiere zuerst Anforderungen, leite danach HTML- und CSS-Entscheidungen ab und vergleiche am Ende Soll und Ist. Die Gestaltung darf begründet verändert werden. Nutze die Offline-Mockup-Vorlage.

Untersuche die View, bevor du den Quellcode betrachtest.

1. Markiere sichtbare Bereiche, Navigation, Informationen und Aktionen.
2. Welche Dateien und Inhalte braucht die statische Version dieser Seite?
3. Welche sichtbaren Elemente könnten später dynamisch werden?

Deine Klasse soll eine Website für einen persönlichen Lernagenten entwickeln. Im bereitgestellten Offline-Projektordner findest du verschiedene Ordner und Dateien. Einige enden auf .html, andere auf .css. Bevor du selbst programmierst, musst du verstehen, welche Aufgabe diese Bestandteile erfüllen.

Öffne die bereitgestellte Startseite im Browser. Untersuche nur das sichtbare Ergebnis und notiere:

1. Welche Bereiche erkennst du auf der Seite?

2. Welche Bestandteile enthalten Informationen und welche gestalten hauptsächlich das Aussehen?
3. Was müsste später hinzukommen, damit der Lernagent auf Eingaben reagieren kann?

MÖGLICHER STARTPROMPT

Ich beginne gerade mit Webentwicklung und untersuche eine vorhandene Website. Stelle mir nacheinander höchstens fünf Fragen, damit ich selbst den Unterschied zwischen Browser, Website, Webseite, HTML und CSS herausarbeite. Gib nicht sofort alle Definitionen vor.

Warum brauchst du das später?

Der persönliche Lernagent entsteht als Website und wird schrittweise erweitert. Nur wenn du Inhalt, Gestaltung und späteres Verhalten unterscheiden kannst, kannst du Fehler gezielt suchen und neue Funktionen an der richtigen Stelle ergänzen.

Denkmodell: Was geschieht beim Aufruf?

Der Browser liest die Dateien einer Webseite und stellt daraus die sichtbare Oberfläche her. Für den Einstieg genügt folgendes Modell:

Begriff	Aufgabe	Beispiel im Projekt
Browser	Liest und stellt Webinhalte dar.	Chrome, Safari, Edge oder Firefox
Website	Zusammengehöriger Webauftritt aus einer oder mehreren Webseiten.	Der gesamte Webauftritt des Lernagenten
Webseite	Ein einzelnes Dokument innerhalb einer Website.	index.html oder lernmodule.html
HTML	Beschreibt Inhalt und semantische Struktur.	Überschrift, Navigation, Lernkarte
CSS	Gestaltet Darstellung und Layout.	Farben, Abstände, Flexbox und Grid
JavaScript	Ermöglicht später Verhalten und Interaktion.	Noch nicht Bestandteil dieses Moduls

MERKSATZ

HTML beschreibt, was auf einer Webseite steht und welche Bedeutung die Bestandteile haben. CSS beschreibt, wie diese Bestandteile dargestellt und angeordnet werden.

Dateien und Ordner im Startprojekt

Eine übersichtliche Projektstruktur verhindert viele Fehler. Für den ersten Lernschritt verwenden wir:

- index.html – Startseite und Einstiegspunkt des Projekts
- css/style.css – zentrale Gestaltungsdatei
- images/ – Ordner für Bilder und Grafiken
- README.md – kurze Projektdokumentation

Die Datei index.html hat eine besondere Rolle: Viele Webserver suchen bei einem Ordneraufruf automatisch nach einer Startdatei mit diesem Namen.

Pflichtauftrag · WEB-K01-A01 · Grundlagen – verbindlich

AUFTRAG

Bearbeite diesen Auftrag vollständig. Er sichert die verbindlichen Kompetenzen: Erkunde das bereitgestellte Offline-Startprojekt und erstelle eine verständliche Bestandsaufnahme, ohne zunächst Dateien zu verändern.

1. Öffne die Ordnerstruktur und notiere mindestens acht vorhandene Dateien oder Ordner.
2. Ordne jedem Eintrag eine vermutete Aufgabe zu.
3. Kennzeichne, welche Dateien statische Inhalte, Gestaltung oder bereits dynamisches Verhalten betreffen.
4. Öffne index.html im Browser und anschließend im Editor. Beschreibe den Unterschied zwischen beiden Ansichten.
5. Erkläre Browser, Website, Webseite, HTML und CSS mit eigenen Worten.
6. Formuliere zwei Fragen, die nach der Erkundung noch offen sind.

DEINE BEARBEITUNG

Trage hier deine Begriffserklärungen, Zuordnungen und offenen Fragen ein. Dokumentiere nur Ergebnisse, die du selbst erklären kannst.

Erwartetes Lernprodukt

Eine Bestandsaufnahme mit mindestens acht Einträgen, fünf eigenen Begriffserklärungen, einer Gegenüberstellung von Browser- und Editoransicht sowie zwei fachlich sinnvollen offenen Fragen.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 1. Doppelstunde: Bestandsaufnahme und Begriffserklärungen.

ZU HAUSE · nur falls der Pflichtteil offen ist: höchstens 15–20 Minuten fertigstellen.

VERTIEFUNG · wenn du früher fertig bist: Lernwebseiten vergleichen oder eigene Benennungsregeln entwickeln.

Gestufte Tutorhilfe

- Hilfe 1: Achte zuerst auf Dateiendungen und Ordnernamen.

- Hilfe 2: .html beschreibt normalerweise Inhalte und Struktur; .css beschreibt die Gestaltung.
- Hilfe 3: Vergleiche denselben Text im Editor und im Browser. Welche Zeichen zeigt der Browser nicht direkt an?

Prüfnachweis

- Die Begriffe Website und Webseite werden korrekt unterschieden.
- HTML und CSS erhalten unterschiedliche Aufgaben.
- Die Dateizuordnungen sind nachvollziehbar begründet.
- Dynamische Inhalte werden erkannt, aber noch nicht programmiert.
- Die Antworten sind in eigenen Worten formuliert.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Entwickle einen eigenen Lösungsweg, begründe Entscheidungen und überprüfe das Ergebnis.

- V1.** Vergleiche eine Webseite mit einem gedruckten Arbeitsblatt. Ordne HTML und CSS passenden Bestandteilen des Arbeitsblatts zu und begründe deine Zuordnung.
- V2.** Untersuche zwei verschiedene Webseiten für Lernangebote. Notiere drei Gemeinsamkeiten und drei Unterschiede ihrer sichtbaren Struktur.
- V3.** Entwickle drei Regeln für eine verständliche Ordner- und Dateibenennung im Webprojekt.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Plane zunächst nur die sichtbaren Bereiche deines späteren Lernagenten: Begrüßung, Themenauswahl, Aufgabenbereich, Hilfebereich und Lernfortschritt. Entscheide für jeden Bereich, welche Information dort später angezeigt werden soll.

HALTE FEST

sichtbarer Bereich · benötigter Inhalt · zugehörige Datei · erwartetes Ergebnis · einfacher Test

Mini-Check

SELBSTPRÜFUNG

Erkläre mit eigenen Worten: Was war die wichtigste fachliche Unterscheidung in diesem Modul und woran erkennst du, dass deine Bestandsaufnahme korrekt ist?

Modul 2: Projektstruktur, Dateinamen und relative Pfade [← Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst ein Webprojekt übersichtlich strukturieren, geeignete Datei- und Ordnernamen verwenden und relative Pfade zwischen HTML-, CSS- und Bilddateien bilden.

Situation und jetzt tun

Die Startseite des Lernagenten wird zwar im Browser geöffnet, aber das Logo fehlt und die Seite ist unformatiert. Im Projekt wurden Dateien verschoben und mehrere Namen geändert. Deine Aufgabe ist zunächst nicht, neuen Quellcode zu schreiben, sondern die Fehlerursachen systematisch zu finden.

FEHLERHAFTES AUSGANGSPROJEKT

```
lernagent/  
  index.html  
  CSS/Style.css  
  bilder/Logo Lernagent.svg  
  seiten/lernmodule.html
```

In index.html stehen unter anderem:

```
<link rel="stylesheet" href="css/style.css">  

```

Notiere vor dem Weiterlesen mindestens drei mögliche Ursachen dafür, dass Stylesheet und Logo nicht geladen werden.

MÖGLICHER STARTPROMPT

Ich untersuche eine fehlerhafte Ordnerstruktur einer Website. Stelle mir nacheinander Fragen zu Ausgangsdatei, Zielordner, Dateinamen, Groß- und Kleinschreibung und Pfadtrennzeichen. Verrate den fertigen Pfad erst, wenn ich meinen Lösungsweg erklärt habe.

Warum brauchst du das später?

Der Lernagent wird aus mehreren Seiten, Stylesheets, Bildern und später auch Skripten und Datenquellen bestehen. Eine klare Projektstruktur sorgt dafür, dass diese Bestandteile zuverlässig gefunden, gemeinsam genutzt und getrennt weiterentwickelt werden können.

Denkmodell: Ein Pfad ist eine Wegbeschreibung

Ein relativer Pfad beschreibt den Weg von der Datei, in der der Verweis steht, zur benötigten Zieldatei. Ausgangspunkt ist deshalb immer die aktuelle Datei – nicht automatisch der Projektordner.

Schreibweise	Bedeutung	Beispiel
datei.html	Datei im selben Ordner	href="lernmodule.html"
ordner/datei.css	in einen Unterordner wechseln	href="css/style.css"
../datei.html	einen Ordner nach oben wechseln	href="../index.html"
../ordner/datei.png	nach oben und dann in einen anderen Ordner	src="../images/logo.png"
https://...	absolute Webadresse	href="https://www.w3.org/"

PFADREGEL

Dateiname, Dateiendung, Groß- und Kleinschreibung sowie Ordnername müssen exakt zum tatsächlichen Speicherort passen. Verwende in Webpfaden den Schrägstrich /.

Gute Datei- und Ordnernamen

Webprojekte werden leichter übertragbar, wenn Namen einheitlich und technisch robust sind.

- kurz und inhaltlich verständlich: lernfortschritt.html statt seite2neu.html
- Kleinbuchstaben konsequent verwenden
- keine Leerzeichen; stattdessen Bindestriche: lernagent-logo.png
- Umlaute und Sonderzeichen in Datei- und Ordnernamen vermeiden
- Dateiendungen nicht verändern oder weglassen

Pflichtauftrag · WEB-K02-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite diesen Auftrag vollständig. Er sichert die verbindlichen Kompetenzen: Repariere die Dateistruktur einer statischen Lernagenten-Website und dokumentiere auch deinen Prüfweg.

1. Vergleiche die beiden Pfade aus dem Ausgangsprojekt mit den tatsächlichen Datei- und Ordnernamen.
2. Nenne für jeden Fehler die genaue Ursache.
3. Entwickle eine bereinigte Ordnerstruktur mit einheitlichen Namen.
4. Formuliere danach die beiden korrekten relativen Pfade in index.html.
5. Bestimme aus seiten/lernmodule.html den Pfad zurück zur Startseite.
6. Bestimme aus seiten/lernmodule.html den Pfad zum Logo im Ordner images.

7. Teste jeden Verweis im Browser und dokumentiere das sichtbare Ergebnis.

DEINE BEARBEITUNG

Dokumentiere hier die Fehlerursachen, die bereinigte Struktur, deine korrigierten Pfade und die Ergebnisse deiner Browserprüfung.

Deine bereinigte Projektstruktur

Deine korrigierten Pfade

Erwartetes Lernprodukt

Eine übersichtliche Projektstruktur, vier korrekte relative Pfade, eine nachvollziehbare Fehleranalyse und ein kurzer Testnachweis für Stylesheet, Logo und interne Navigation.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 2. Doppelstunde: Ursachen, bereinigte Namen, vier Pfade und Browserprüfung.

ZU HAUSE · nur falls der Pflichtteil offen ist: Fehlerprotokoll beziehungsweise Restarbeit abschließen.

VERTIEFUNG · wenn du früher fertig bist: Hilfeseite, Download-Pfade, eigene Struktur oder Peer-Test.

Gestufte Tutorhilfe

- Hilfe 1: Vergleiche Zeichen für Zeichen: CSS oder css? Style.css oder style.css?
- Hilfe 2: Bereinige zuerst die Namen. Bilde die Pfade erst danach.
- Hilfe 3: Von index.html führt der Weg direkt in Unterordner. Von seiten/lernmodule.html musst du zuerst mit ../ nach oben.
- Hilfe 4: Nutze die Netzwerkansicht der Browserwerkzeuge. Ein Status 404 weist häufig auf einen falschen Pfad hin.

Prüfnachweis

- Alle Datei- und Ordnernamen entsprechen den vereinbarten Regeln.
- Die CSS-Datei wird geladen und verändert sichtbar die Darstellung.
- Das Logo wird angezeigt und besitzt einen sinnvollen Alternativtext.
- Die Navigation zwischen Startseite und Lernmoduleseite funktioniert in beide Richtungen.
- Der Prüfungsweg ist so dokumentiert, dass eine andere Person ihn nachvollziehen kann.

Zusatzauftrag · WEB-K02-A02 · Pfad-Fehlersuche

Die Datei `seiten/hilfe.html` enthält die folgenden Verweise. Die tatsächlichen Ziele liegen unter `css/hilfe.css`, `images/hilfe-symbol.svg` und `index.html`.

ZU PRÜFENDER QUELLCODE

```
<link rel="stylesheet" href="css/hilfe.css">

<a href="index.html">Zur Startseite</a>
```

1. Markiere jeden fehlerhaften Pfad.
2. Formuliere den korrigierten Quellcode.
3. Begründe bei jedem Pfad, warum `../` benötigt wird oder nicht benötigt wird.
4. Verbessere außerdem den Alternativtext des Bildes.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Entwickle einen eigenen Lösungsweg, begründe Entscheidungen und überprüfe das Ergebnis.

- V1. Ergänze einen Ordner `downloads` mit einer Datei `lernplan.pdf`. Verlinke sie von `index.html` und von `seiten/lernmodule.html`.
- V2. Erkläre den Unterschied zwischen einem relativen Pfad und einer vollständigen Webadresse. Nenne je einen sinnvollen Anwendungsfall.
- V3. Entwickle eine Prüfcheckliste für defekte Bilder, Stylesheets und interne Links.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Lege für dein Lernagenten-Projekt eine verbindliche Ordnerstruktur fest. Plane mindestens Startseite, Lernmodule, Bilder, Stylesheets und Downloads. Begründe, welche Bestandteile gemeinsam genutzt werden.

HALTE FEST

Ausgangsdatei · Zielfile · relativer Pfad · Fehlerursache · sichtbares Testergebnis

Mini-Check

SELBSTPRÜFUNG

Erkläre mit eigenen Worten: Welche Regel ist beim Bilden eines relativen Pfades entscheidend und woran erkennst du, dass deine Projektstruktur zuverlässig funktioniert?

Modul 3: Das HTML-Grundgerüst und erste Inhalte [↩ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst ein gültiges HTML-Dokument anlegen, den head vom sichtbaren body unterscheiden und Texte semantisch gliedern.

Situation und jetzt tun

Der Lernagent benötigt nun eine eigene Startseite. Eine leere Datei zeigt im Browser zwar möglicherweise Text, enthält aber noch keine verlässliche Dokumentstruktur.

Skizziere aus dem Gedächtnis, welche Bestandteile jedes HTML-Dokument vermutlich benötigt. Markiere, welche Angaben sichtbar und welche nur für Browser oder Suchsysteme bestimmt sind.

MÖGLICHER STARTPROMPT

Ich lerne das HTML-Grundgerüst. Stelle mir nacheinander Fragen zu doctype, html, head, meta, title und body. Lass mich erst selbst antworten und gib danach nur einen Hinweis. Erzeuge keine vollständige Seite für mich.

Warum brauchst du das später?

Jede weitere Seite des Lernagenten baut auf demselben Grundgerüst auf. Ein sauberer Start verhindert Darstellungs-, Zeichen- und Zugänglichkeitsprobleme.

Denkmodell und Fachwissen

- `<!doctype html>` kennzeichnet ein modernes HTML5-Dokument.
- `<html lang="de">` legt die Dokumentsprache fest.
- Im head stehen Metadaten wie Zeichencodierung, Viewport und Seitentitel; sichtbare Inhalte stehen im body.
- `<meta charset="utf-8">` erlaubt direkte Umlaute; HTML-Zeichenreferenzen bleiben für reservierte Zeichen wie `<` und `&` wichtig.
- Überschriften h1 bis h6 bilden eine inhaltliche Hierarchie; ihre Nummer wird nicht nach gewünschter Schriftgröße gewählt.

- Kommentare werden mit `<!-- ... -->` geschrieben und erscheinen nicht als Seiteninhalt.

Pflichtauftrag · WEB-K03-A01 · Grundlagen – verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Erstelle die Datei `index.html` mit vollständigem HTML-Grundgerüst.
2. Setze Sprache, UTF-8, Viewport und einen aussagekräftigen `title`.
3. Ergänze eine `h1`, zwei logisch untergeordnete Überschriften, Absätze und eine kurze Hervorhebung.
4. Füge zwei hilfreiche HTML-Kommentare ein.
5. Öffne die Seite im Browser, prüfe Tabtitel und Umlaute und dokumentiere das Ergebnis.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine gültig aufgebaute `index.html` mit nachvollziehbarer Überschriftenhierarchie, korrekten Metadaten und einem kurzen Browsertest.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 3. Doppelstunde in Woche 8.

ZU HAUSE · nur einen begonnenen Pflichtteil fertigstellen.

VERTIEFUNG · zusätzlichen Abschnitt erstellen und die Überschriftenfolge begründen.

Gestufte Tutorhilfe

- Hilfe 1: Beginne mit `doctype`, `html`, `head` und `body`.
- Hilfe 2: Der `title` gehört in den `head`; die sichtbare `h1` gehört in den `body`.
- Hilfe 3: Prüfe Einrückung und schließende Tags paarweise.

Prüfnachweis

- Die Datei heißt `index.html` und öffnet sich lokal.
- UTF-8, Sprache, Viewport und `title` sind vorhanden.
- Es gibt genau eine inhaltliche Hauptüberschrift `h1`.
- Umlaute werden korrekt dargestellt und Kommentare bleiben unsichtbar.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Ergänze eine zweite Seite lernziele.html mit demselben Grundgerüst.
- V2.** Vergleiche direkte Umlaute mit einer passenden HTML-Zeichenreferenz und erkläre, wann Zeichenreferenzen weiterhin nötig sind.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Formuliere Begrüßung, Zweck und erste Lernziele für die Startseite deines Lernagenten. Die Seite bleibt zunächst vollständig statisch.

HALTE FEST

Dateiname · Seitentitel · sichtbare Hauptüberschrift · Sprache · getesteter Browser

Mini-Check

SELBSTPRÜFUNG

Erkläre den Unterschied zwischen title und h1 sowie zwischen head und body.
Welche Prüfung beweist, dass UTF-8 funktioniert?

Modul 4: Semantische Seitenstruktur [↔ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst Inhalte mit header, nav, main, section, article, aside und footer bedeutungsvoll strukturieren.

Situation und jetzt tun

Die Inhalte des Lernagenten stehen bisher untereinander. Menschen erkennen Bereiche durch Gestaltung; Browser und Hilfstechnologien benötigen zusätzlich eine sinnvolle semantische Struktur.

Ordne Begrüßung, Hauptnavigation, Lernkarte, Zusatzhinweis und Quellenangabe passenden Seitenbereichen zu. Begründe zwei Zuordnungen.

MÖGLICHER STARTPROMPT

Ich möchte eine Webseite semantisch gliedern. Frage mich nacheinander nach der Aufgabe von header, nav, main, section, article, aside und footer. Gib Beispiele, aber keinen fertigen Quellcode.

Warum brauchst du das später?

Semantik erleichtert Wartung, Suchmaschinenverständnis, Tastaturnavigation und spätere dynamische Erweiterungen.

Denkmodell und Fachwissen

- header kennzeichnet einleitende Inhalte eines Bereichs, footer abschließende Informationen.
- nav umfasst eine wesentliche Navigation; nicht jede Linkgruppe benötigt nav.
- main enthält den einmaligen Hauptinhalt einer Seite und kommt üblicherweise einmal vor.
- section bündelt einen thematischen Abschnitt, meist mit eigener Überschrift.
- article eignet sich für eigenständig verständliche Beiträge oder Karten.
- aside enthält ergänzende, nicht zentrale Informationen.
- Semantische Elemente ersetzen keine Überschriftenhierarchie.

Pflichtauftrag · WEB-K04-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Überarbeite deine Startseite mit header, nav, main und footer.
2. Gliedere den Hauptinhalt in mindestens zwei sinnvoll benannte sections.
3. Setze für eine eigenständige Lernkarte article und für einen ergänzenden Tipp aside ein.
4. Kontrolliere, ob jedes Element seiner Bedeutung entsprechend eingesetzt ist.
5. Teste die Seite ohne CSS: Die inhaltliche Reihenfolge muss weiterhin verständlich sein.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine semantisch gegliederte Startseite mit Strukturskizze und kurzer Begründung für section, article und aside.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 1. Doppelstunde in Woche 9.
 ZU HAUSE · nur offene Strukturbegründung ergänzen.
 VERTIEFUNG · zweite Inhaltsseite semantisch planen.

Gestufte Tutorhilfe

- Hilfe 1: Markiere zuerst in deiner Skizze Kopf, Navigation, Hauptinhalt und Abschluss.
- Hilfe 2: Eine section braucht einen erkennbaren thematischen Zweck.
- Hilfe 3: Prüfe die Lesereihenfolge, indem du CSS vorübergehend nicht lädst.

Prüfnachweis

- main ist eindeutig und enthält den Hauptinhalt.
- Bereiche besitzen passende Überschriften.
- article und aside sind fachlich begründet.
- Die Reihenfolge funktioniert auch ohne Gestaltung.

Vertiefung - weiterdenken und übertragen**VERTIEFUNG**

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Erstelle für lernmodule.html eine semantische Struktur mit mehreren eigenständigen Lernkarten.
- V2.** Vergleiche eine div-basierte und eine semantische Struktur und erläutere den Informationsgewinn.

Transfer zum Learning Agent**AGENTENBAUSTEIN**

Ordne die geplanten Bereiche Begrüßung, Themenauswahl, Aufgabe, Hilfe und Lernfortschritt semantischen Elementen zu.

HALTE FEST

Inhalt · semantisches Element · Begründung · Überschrift · Lesereihenfolge

Mini-Check**SELBSTPRÜFUNG**

Woran erkennst du, ob section oder article geeigneter ist? Warum darf Gestaltung nicht die einzige Strukturinformation sein?

Modul 5: Listen, Links und Navigation [↩ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst Listen passend auswählen, interne und externe Links bilden und eine verständliche Navigation erstellen.

Situation und jetzt tun

Navigation in Desktop- und Mobilansicht

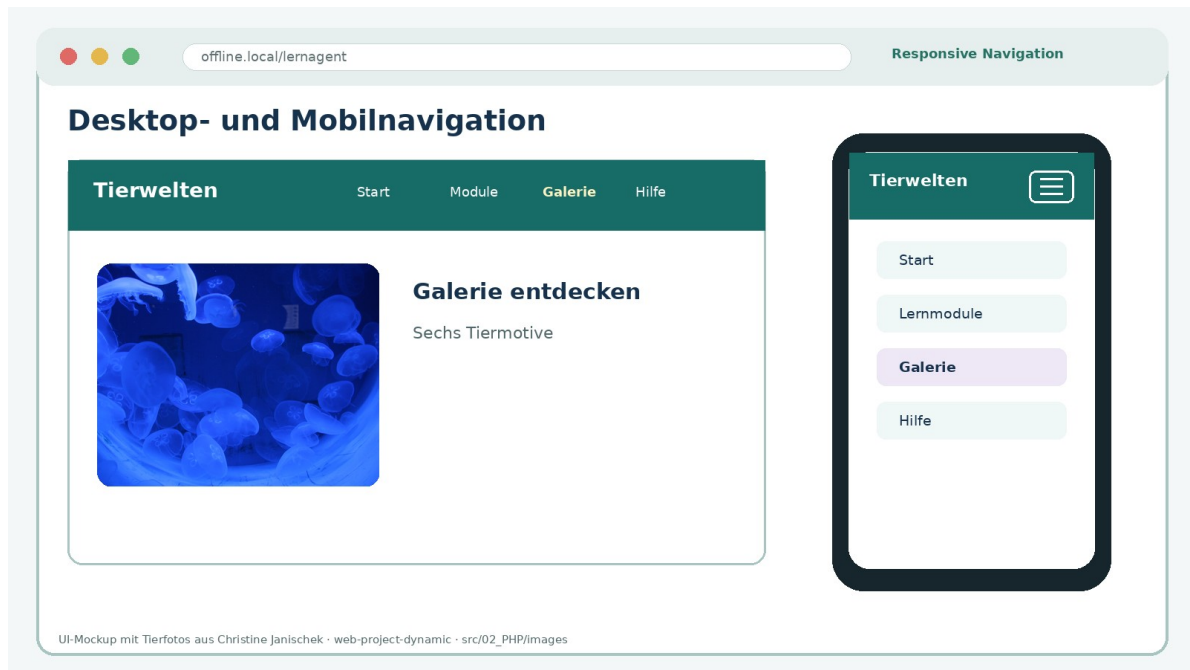


Abbildung 2: Dieselbe Navigation in einer breiten und einer schmalen Ansicht.

Mockup-Check 2/4 – möglicher Zielentwurf. Markiere zuerst Anforderungen, leite danach HTML- und CSS-Entscheidungen ab und vergleiche am Ende Soll und Ist. Die Gestaltung darf begründet verändert werden. Nutze die Offline-Mockup-Vorlage.

Leite aus beiden Ansichten Anforderungen an HTML-Struktur und Bedienbarkeit ab.

1. Welche Navigationselemente bleiben in beiden Ansichten erhalten?
2. Welche Zustände lassen sich mit HTML und CSS darstellen – und wofür wäre später JavaScript nötig?
3. Prüfe Tastaturreihenfolge, verständliche Linktexte und sichtbaren Fokus.

Der Lernagent soll mehrere Themen und Hilfeseiten erreichbar machen. Eine bloße Reihe unbeschrifteter Links bietet jedoch wenig Orientierung.

Plane eine Navigation mit Start, Lernmodulen, Hilfe und Über das Projekt. Entscheide außerdem, welche Inhalte als geordnete beziehungsweise ungeordnete Liste dargestellt werden.

MÖGLICHER STARTPROMPT

Ich plane Listen und Links für eine Lernwebsite. Stelle mir Fragen zu `ul`, `ol`, `li`, `href`, relativen Pfaden und aussagekräftigen Linktexten. Prüfe meine Antworten, ohne die Navigation fertig zu schreiben.

Warum brauchst du das später?

Navigation verbindet später alle Projektseiten. Gute Linktexte und Listenstrukturen helfen sowohl sehenden Nutzenden als auch Screenreader-Nutzenden.

Denkmodell und Fachwissen

- `ul` beschreibt eine ungeordnete Sammlung; `ol` eine Reihenfolge oder Abfolge.
- Listeneinträge stehen in `li`-Elementen.
- Das `a`-Element erhält sein Ziel über `href`.
- Interne Projektlings verwenden meist relative Pfade; externe Ziele vollständige Webadressen.
- Linktexte sollen das Ziel erkennen lassen; 'Hier klicken' ist meist zu ungenau.
- Ein neues Browserfenster ist keine Pflicht. Wenn `target="_blank"` begründet eingesetzt wird, muss die neue Situation erkennbar und technisch abgesichert sein.

Pflichtauftrag · WEB-K05-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Erstelle eine `ul`-basierte Hauptnavigation mit mindestens vier verständlichen Linktexten.
2. Verlinke mindestens zwei lokale HTML-Seiten korrekt.
3. Ergänze eine geordnete Liste für einen Lernablauf und eine ungeordnete Themenliste.
4. Füge einen externen, fachlich sinnvollen Link hinzu und kennzeichne ihn als extern.
5. Teste alle Links mit Tastatur und Maus und protokolliere Ziel und Ergebnis.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine mehrseitige Offline-Website mit Listen, interner Navigation, einem begründeten externen Link und vollständigem Linktest.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 2. Doppelstunde in Woche 9.

ZU HAUSE · nur den Linktest abschließen.

VERTIEFUNG · aktuelle Seite visuell und textlich kennzeichnen.

Gestufte Tutorhilfe

- Hilfe 1: Erstelle zuerst alle Zieldateien, dann die Links.
- Hilfe 2: Beginne interne Pfade bei der aktuellen Datei.
- Hilfe 3: Lies nur die Linktexte laut vor: Sind die Ziele verständlich?

Prüfnachweis

- Alle internen Links funktionieren offline.
- Listenart und Inhalt passen zusammen.
- Linktexte beschreiben ihre Ziele.
- Fokus ist sichtbar und die Navigation per Tab erreichbar.

Vertiefung - weiterdenken und übertragen**VERTIEFUNG**

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

V1. Ergänze Sprunglinks zu Abschnitten einer längeren Seite.

V2. Entwickle eine Kennzeichnung für die aktuell geöffnete Seite und erläutere aria-current="page".

Transfer zum Learning Agent**AGENTENBAUSTEIN**

Verbinde die Seiten des Lernagenten über eine konsistente Hauptnavigation und notiere die Navigationslogik.

HALTE FEST

Ausgangsseite · Linktext · Ziel · Pfadart · Testergebnis

Mini-Check**SELBSTPRÜFUNG**

Wann verwendest du ul, wann ol? Woran erkennst du einen guten Linktext und einen funktionierenden internen Pfad?

Modul 6: HTML prüfen und erste Barrieren beseitigen [← Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst HTML systematisch prüfen, typische Strukturfehler verbessern und grundlegende Zugänglichkeit testen.

Situation und jetzt tun

Die Seite sieht im Browser richtig aus. Trotzdem können verschachtelte Elemente, fehlende Beschriftungen oder unsichtbarer Tastaturfokus fehlerhaft sein.

Notiere drei Prüfungen, die über den bloßen Satz 'Bei mir sieht es gut aus' hinausgehen.

MÖGLICHER STARTPROMPT

Ich möchte meine HTML-Seite selbst prüfen. Frage mich Schritt für Schritt nach Syntax, Überschriften, Bildern, Links, Tastaturbedienung und verschiedenen Browserbreiten. Nenne erst nach meiner Antwort ein mögliches Prüfwerkzeug.

Warum brauchst du das später?

Qualität wird später nicht nur am Aussehen beurteilt. Systematische Prüfungen machen Fehler reproduzierbar und Ergebnisse nachvollziehbar.

Denkmodell und Fachwissen

- Ein Validator prüft formale HTML-Regeln; ein fehlerfreies Ergebnis garantiert noch keine gute Semantik.
- Jedes informative Bild benötigt einen passenden Alternativtext; rein dekorative Bilder erhalten alt="".
- Überschriften sollen die Dokumentstruktur beschreiben.
- Interaktive Elemente müssen mit der Tastatur erreichbar und der Fokus sichtbar sein.
- Automatische Prüfwerkzeuge ergänzen, aber ersetzen keine manuelle Prüfung.
- Offline kann zunächst mit Quellcodekontrolle, Browserkonsole, Tastaturtest und mehreren Fensterbreiten gearbeitet werden.

Pflichtauftrag · WEB-K06-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Prüfe dein HTML zunächst manuell auf Grundgerüst, Verschachtelung und schließende Tags.
2. Führe einen Überschriften-, Bild- und Linktest durch.

3. Bediene die komplette Navigation nur mit Tab, Umschalt+Tab und Enter.
4. Nutze bei vorhandener Internetverbindung zusätzlich einen HTML-Validator; dokumentiere höchstens drei relevante Meldungen.
5. Verbessere gefundene Fehler und wiederhole die betroffenen Tests.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Ein kurzes Prüfprotokoll mit Ausgangsfehler, Prüfverfahren, Korrektur und erneutem Testergebnis.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 3. Doppelstunde in Woche 9.

ZU HAUSE · nur begonnenes Prüfprotokoll abschließen.

VERTIEFUNG · ausführlicher Tastatur- oder Kontrasttest.

Gestufte Tutorhilfe

- Hilfe 1: Prüfe zuerst einen Bereich nach dem anderen.
- Hilfe 2: Ein Warnhinweis ist nicht automatisch ein Fehler; lies die Begründung.
- Hilfe 3: Nach jeder Korrektur muss der Test wiederholt werden.

Prüfnachweis

- Das Prüfprotokoll nennt konkrete Verfahren.
- Mindestens ein Fehler wurde nachvollziehbar verbessert.
- Navigation und Links sind per Tastatur erreichbar.
- Alternativtexte passen zur Funktion der Bilder.

Vertiefung - weiterdenken und übertragen**VERTIEFUNG**

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1. Lass eine andere Person die Seite ohne Maus bedienen und beobachte nur.
- V2. Vergleiche automatische und manuelle Prüfergebnisse und benenne ihre Grenzen.

Transfer zum Learning Agent**AGENTENBAUSTEIN**

Erstelle eine wiederverwendbare Qualitätscheckliste für jede spätere Lernagenten-Seite.

HALTE FEST

Prüfgegenstand · Verfahren · Befund · Änderung · Nachtest

Mini-Check

SELBSTPRÜFUNG

Warum reicht ein fehlerfreier Validatorbericht nicht aus? Welche Prüfung kannst du immer offline durchführen?

Modul 7: CSS einbinden, auswählen und anwenden [↩ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst ein externes Stylesheet einbinden, Selektoren unterscheiden und Kaskade sowie Vererbung grundlegend erklären.

Situation und jetzt tun

Mehrere Lernagenten-Seiten sollen gleich aussehen. Einzelne style-Attribute würden sich wiederholen und wären schwer zu pflegen.

Markiere in einem kleinen HTML-Ausschnitt Elemente, die global, als Gruppe oder einzeln gestaltet werden sollten.

MÖGLICHER STARTPROMPT

Ich lerne CSS. Stelle mir nacheinander Fragen zu externem Stylesheet, Element-, Klassen- und ID-Selektor, Kaskade und Vererbung. Lass mich Vorhersagen zur Wirkung treffen, bevor du erklärst.

Warum brauchst du das später?

Eine zentrale CSS-Datei ist die Grundlage für wiederverwendbare Komponenten, Corporate Design und responsive Layouts.

Denkmodell und Fachwissen

- Ein externes Stylesheet wird im head mit link und rel="stylesheet" eingebunden.
- Ein Elementselektor wirkt auf alle passenden Elemente, eine Klasse kann mehrfach verwendet werden, eine ID soll innerhalb einer Seite eindeutig sein.

- Eine CSS-Regel besteht aus Selektor und Deklarationsblock; Deklarationen bestehen aus Eigenschaft und Wert.
- Die Kaskade berücksichtigt Herkunft, Wichtigkeit, Spezifität und Reihenfolge.
- Einige Eigenschaften wie color werden häufig vererbt, andere wie margin nicht.
- Browserwerkzeuge zeigen angewendete und überschriebene Regeln.

Pflichtauftrag · WEB-K07-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Lege css/style.css an und binde sie in alle HTML-Seiten ein.
2. Gestalte body und Überschriften mit Elementselektoren.
3. Nutze eine Klasse für mehrere Lernkarten und eine zweite Klasse für einen Statushinweis.
4. Erzeuge bewusst einen kleinen Kaskadenkonflikt und erkläre, welche Regel gewinnt.
5. Prüfe in den Browserwerkzeugen mindestens eine geerbte und eine überschriebene Eigenschaft.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Ein extern gestaltetes Mehrseitenprojekt mit Element- und Klassenselektoren sowie einer dokumentierten Kaskadenentscheidung.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 1. Doppelstunde in Woche 10.
ZU HAUSE · nur CSS-Kommentare und Begründung vervollständigen.
VERTIEFUNG · Spezifität an drei Regeln untersuchen.

Gestufte Tutorhilfe

- Hilfe 1: Prüfe zuerst den link-Pfad und speichere beide Dateien.
- Hilfe 2: Teste Klassen zunächst mit einer deutlich sichtbaren Eigenschaft.
- Hilfe 3: Die Browserwerkzeuge zeigen durchgestrichene, überschriebene Regeln.

Prüfnachweis

- Stylesheet wird auf allen Seiten geladen.
- Klassen werden wiederverwendet und IDs nicht als allgemeine Gestaltungs-klasse missbraucht.
- CSS-Syntax ist korrekt.

- Die Kaskadenentscheidung ist fachlich begründet.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Vergleiche Klasse, Nachfahrenselektor und ID hinsichtlich Wiederverwendbarkeit und Spezifität.
- V2.** Formuliere eine Regel, wie unnötig hohe Spezifität vermieden wird.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Lege erste wiederverwendbare Klassen für Lernkarte, Hinweis und Statusanzeige des Lernagenten an.

HALTE FEST

Selektor · betroffene Elemente · Eigenschaft · Vererbung/Kaskade · sichtbarer Test

Mini-Check

SELBSTPRÜFUNG

Wann ist eine Klasse geeigneter als eine ID? Welche vier Einflussgrößen bestimmen die Kaskade?

Modul 8: Typografie, Farben und Corporate Design [↔ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst ein zielgruppengerechtes Mini-Designsystem entwickeln und Lesbarkeit sowie Farbkontrast begründen.

Situation und jetzt tun

Der Lernagent funktioniert, wirkt aber uneinheitlich. Farben und Schriften sollen nicht zufällig, sondern als wiederholbares Gestaltungssystem eingesetzt werden.

Wähle für eine Lernwebsite eine Hauptfarbe, eine Akzentfarbe, eine Textfarbe und eine Hintergrundfarbe. Begründe die Wirkung und vermute mögliche Kontrastprobleme.

MÖGLICHER STARTPROMPT

Ich entwickle ein kleines Corporate Design für eine Lernwebsite. Stelle mir Fragen zu Zielgruppe, Farbrollen, Kontrast, Schriftfamilie, Schriftgrößen und Wiederverwendung. Bewerte meine Entscheidung erst nach meiner Begründung.

Warum brauchst du das später?

Ein konsistentes Design unterstützt Orientierung und Vertrauen. Später können alle Lernmodule dieselben Gestaltungsvariablen und Komponenten verwenden.

Denkmodell und Fachwissen

- Corporate Design beschreibt ein konsistentes visuelles Erscheinungsbild, nicht nur ein Logo.
- Farben benötigen feste Rollen, etwa Hintergrund, Text, Primär- und Akzentfarbe.
- CSS Custom Properties im `:root` erleichtern konsistente Änderungen.
- Systemschriften funktionieren offline und vermeiden externe Abhängigkeiten.
- Lesbarer Fließtext benötigt ausreichende Größe, Zeilenhöhe und begrenzte Zeilenlänge.
- Farbkontrast muss geprüft werden; Farbe allein darf keine notwendige Information vermitteln.

Pflichtauftrag · WEB-K08-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Definiere im `:root` mindestens vier verständlich benannte Farbvariablen.
2. Lege eine offline verfügbare Schriftfamilie, Grundgröße und Zeilenhöhe fest.
3. Gestalte Überschriften, Navigation, Lernkarten und Hinweise konsistent.
4. Prüfe zwei wichtige Farbkombinationen mit einem Kontrastwerkzeug oder einer bereitgestellten Kontrasttabelle.
5. Dokumentiere dein Mini-Designsystem mit Farbrollen, Typografie und drei Gestaltungsregeln.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine konsistent gestaltete Lernagenten-Seite und ein kompaktes Designblatt mit begründeten Farb- und Typografieentscheidungen.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 2. Doppelstunde in Woche 10.

ZU HAUSE · fehlende Designregeln dokumentieren.

VERTIEFUNG · zweite, begründete Designvariante erstellen.

Gestufte Tutorhilfe

- Hilfe 1: Benenne Farben nach ihrer Funktion statt nach ihrem Farbton.
- Hilfe 2: Beginne mit Text- und Hintergrundkontrast.
- Hilfe 3: Ändere immer nur eine Gestaltungsrolle und vergleiche vorher/nachher.

Prüfnachweis

- Farben werden über wiederverwendbare Variablen gesteuert.
- Fließtext bleibt gut lesbar.
- Mindestens zwei Kontraste sind dokumentiert.
- Zielgruppe und Gestaltung sind nachvollziehbar verbunden.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Entwickle einen hellen und einen kontraststärkeren Darstellungsentwurf mit denselben Komponenten.
- V2.** Analysiere eine professionelle Lernseite hinsichtlich Wiederholung, Hierarchie und Blickführung.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Übertrage Farben, Schriften und Komponentenrollen als kleines Designsystem auf den persönlichen Lernagenten.

HALTE FEST

Zielgruppe · Farbrolle · CSS-Variable · Typografieregeln · Kontrastnachweis

Mini-Check

SELBSTPRÜFUNG

Warum ist Corporate Design mehr als Farbwahl? Welche Gestaltung darf niemals die einzige Trägerin einer Information sein?

Modul 9: Box-Modell und relative Größen [← Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst Content, Padding, Border und Margin unterscheiden und robuste Größen mit rem, %, min(), max() oder clamp() vorbereiten.

Situation und jetzt tun

Die Lernkarten haben unerwartete Breiten und Abstände. Eine feste Pixelbreite funktioniert auf einem Bildschirm, läuft aber auf einem anderen über.

Zeichne das Box-Modell einer Lernkarte und beschrifte Inhalt, Innenabstand, Rahmen und Außenabstand. Sage voraus, was box-sizing: border-box verändert.

MÖGLICHER STARTPROMPT

Ich untersuche das CSS-Box-Modell. Lass mich nacheinander Content, Padding, Border, Margin, width und box-sizing erklären. Gib mir anschließend kleine Rechen- oder Vorhersagefragen, aber keinen fertigen Kartenstil.

Warum brauchst du das später?

Flexbox, Grid und responsive Komponenten lassen sich nur zuverlässig steuern, wenn Größen und Abstände verstanden werden.

Denkmodell und Fachwissen

- Das Box-Modell besteht aus Inhaltsbereich, Padding, Border und Margin.
- Mit box-sizing: border-box umfasst eine gesetzte Breite auch Padding und Border.
- rem bezieht sich auf die Schriftgröße des Wurzelements, em auf den jeweiligen Kontext.
- % und Viewport-Einheiten reagieren auf verfügbare Größen, müssen aber passend eingesetzt werden.
- max-width verhindert zu breite Inhaltsbereiche; width: 100% kann mit einer Obergrenze kombiniert werden.
- Ein einheitliches Abstandssystem verbessert Rhythmus und Wartbarkeit.

Pflichtauftrag · WEB-K09-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Setze global box-sizing: border-box einschließlich Pseudoelementen.
2. Gestalte Lernkarten mit nachvollziehbarem Padding, Border und Margin beziehungsweise gap.
3. Begrenze den Hauptinhalt mit einer flexiblen Breite und max-width.
4. Ersetze mindestens zwei unnötige feste Pixelgrößen durch geeignete relative Werte.
5. Prüfe die Seite bei schmalem und breitem Browserfenster und dokumentiere die Unterschiede.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine skalierbare Kartenkomponente, eine beschriftete Box-Modell-Skizze und ein Test zweier Fensterbreiten.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 3. Doppelstunde in Woche 10.

ZU HAUSE · nur Box-Analyse abschließen.

VERTIEFUNG · Schrift oder Abstand mit clamp() begrenzt skalieren.

Gestufte Tutorhilfe

- Hilfe 1: Färbe testweise Content, Padding und Border unterschiedlich.
- Hilfe 2: Prüfe die berechneten Größen in den Browserwerkzeugen.
- Hilfe 3: Eine flexible Breite benötigt häufig zusätzlich eine sinnvolle Obergrenze.

Prüfnachweis

- Die vier Bereiche des Box-Modells sind korrekt erklärt.
- Kein wichtiger Inhalt läuft bei schmaler Ansicht horizontal über.
- Relative Einheiten sind passend begründet.
- Abstände werden konsistent eingesetzt.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Vergleiche content-box und border-box an derselben Karte.
- V2.** Entwickle eine kleine Skala für Abstände und ersetze zufällige Werte.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Erstelle eine robuste Basiskomponente .lernkarte, die später in Flexbox und Grid verwendet werden kann.

HALTE FEST

Content · Padding · Border · Margin/gap · Breitenregel · Testbreite

Mini-Check

SELBSTPRÜFUNG

Welche Teile umfasst border-box? Wann ist rem geeigneter als Prozent und weshalb benötigt width häufig eine max-width?

Modul 10: Eindimensionale Layouts mit Flexbox [↔ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst Flex-Container und Flex-Items unterscheiden und Elemente in einer Zeile oder Spalte flexibel anordnen.

Situation und jetzt tun

Navigation, Lernkartenleiste und Footer sollen sich an verfügbaren Platz anpassen. Die Elemente werden jeweils entlang einer Hauptachse angeordnet.

Ordne einer Navigation die Begriffe Hauptachse, Querachse, Abstand, Umbruch und Ausrichtung zu. Skizziere eine breite und eine schmale Ansicht.

MÖGLICHER STARTPROMPT

Ich lerne Flexbox an einer Navigation und Kartenleiste. Frage mich nach Container, Items, flex-direction, justify-content, align-items, gap, flex-wrap und flex. Lass mich Werte vorhersagen, bevor du korrigierst.

Warum brauchst du das später?

Flexbox eignet sich besonders für Navigationen, Werkzeugleisten, Kartenreihen und Footer-Spalten. Es ist ein Kernwerkzeug für flexible Oberflächen.

Denkmodell und Fachwissen

- display: flex wird am Container gesetzt; dessen direkte Kinder werden Flex-Items.
- flex-direction bestimmt die Hauptachse.
- justify-content ordnet entlang der Hauptachse, align-items entlang der Querachse an.
- gap erzeugt kontrollierte Abstände ohne Randtricks.
- flex-wrap erlaubt Umbruch, wenn der Platz nicht reicht.
- Mit flex: grow shrink basis können Items wachsen, schrumpfen und eine Ausgangsgröße erhalten.

Pflichtauftrag · WEB-K10-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Baue die Hauptnavigation als Flex-Container mit gut sichtbarem Fokus.
2. Erstelle eine Kartenleiste aus mindestens drei Lernkarten.
3. Erlaube mit flex-wrap einen sinnvollen Umbruch.
4. Setze gap statt äußerer Einzelabstände zwischen den Items ein.
5. Teste schmale und breite Ansichten und erkläre Haupt- und Querachse deiner Lösung.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine flexible Navigation und Kartenleiste mit kommentierten Containerregeln, Umbruch und dokumentiertem Breitentest.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 1. Doppelstunde in Woche 11.

ZU HAUSE · nur Pflichtlayout fertigstellen.

VERTIEFUNG · dreispaltigen Footer oder Statusleiste mit Flexbox gestalten.

Gestufte Tutorhilfe

- Hilfe 1: Setze display: flex nur am gemeinsamen Elternelement.
- Hilfe 2: Wenn Items zu eng werden, prüfe flex-wrap und flex-basis.
- Hilfe 3: justify-content hängt von der aktuellen flex-direction ab.

Prüfnachweis

- Container und Items werden korrekt unterschieden.
- Layout verwendet gap und kann umbrechen.
- Navigation bleibt per Tastatur bedienbar.
- Die Achsen und wichtigsten Eigenschaften sind erklärt.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

V1. Vergleiche row und column einschließlich der veränderten Achsen.

V2. Erstelle einen Footer mit drei flexiblen Bereichen ohne feste Prozentbreiten.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Nutze Flexbox für die Navigation und eine wiederverwendbare Reihe von Lernkarten oder Statusanzeigen.

HALTE FEST

Flex-Container · direkte Items · Hauptachse · Querachse · Umbruch · Breitentest

Mini-Check

SELBSTPRÜFUNG

Wo wird display: flex gesetzt? Wie unterscheiden sich justify-content und align-items, und wann hilft flex-wrap?

Modul 11: Zweidimensionale Layouts mit CSS Grid [↩ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst ein Raster mit Zeilen und Spalten definieren und Seitenbereiche nachvollziehbar darin anordnen.

Situation und jetzt tun

Die Gesamtseite des Lernagenten benötigt gleichzeitig Spalten und Zeilen: Kopf, Navigation, Hauptbereich, optionale Seitenleiste und Abschluss.

Zeichne ein Raster für Desktop und beschrifte Zeilen, Spalten und Bereiche. Markiere, welche innere Komponente weiterhin besser mit Flexbox gelöst wird.

MÖGLICHER STARTPROMPT

Ich lerne CSS Grid für ein Seitenlayout. Frage mich nach Grid-Container, Grid-Items, fr, repeat(), gap und grid-template-areas. Lass mich zuerst ein Raster beschreiben und gib keinen fertigen Code aus.

Warum brauchst du das später?

Grid eignet sich für zweidimensionale Seiten- und Komponentenlayouts. Gemeinsam mit Flexbox lassen sich klare, wartbare Oberflächen entwickeln.

Denkmodell und Fachwissen

- display: grid aktiviert Grid am Container; direkte Kinder werden Grid-Items.
- grid-template-columns und grid-template-rows definieren das Raster.
- Die Einheit fr verteilt verfügbaren Platz in Bruchteilen.
- repeat() vermeidet Wiederholungen, minmax() verbindet Mindest- und Höchstwerte.
- gap trennt Rasterzellen ohne zusätzliche Außenränder.
- grid-template-areas kann ein Seitenlayout lesbar benennen.
- Grid ersetzt keine semantische HTML-Struktur und keine Datentabelle.

Pflichtauftrag · WEB-K11-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Lege für den Hauptbereich ein Grid mit Inhalts- und optionaler Hilfsspalte an.
2. Verwende fr-Einheiten und gap; vermeide starre Gesamtbreiten.
3. Ordne Bereiche entweder über Linien oder benannte grid-template-areas zu.
4. Nutze semantische HTML-Elemente als Grid-Items.
5. Teste, ob die Quellreihenfolge auch ohne Grid sinnvoll bleibt.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Ein zweidimensionales Seitenraster mit Skizze, kommentierter CSS-Regel und Prüfung der semantischen Quellreihenfolge.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 2. Doppelstunde in Woche 11.

ZU HAUSE · Rasterbegründung vervollständigen.

VERTIEFUNG · dasselbe Raster mit grid-template-areas ausdrücken.

Gestufte Tutorhilfe

- Hilfe 1: Beginne mit zwei einfachen Spalten.
- Hilfe 2: Färbe Grid-Items vorübergehend unterschiedlich.
- Hilfe 3: Wenn ein Item falsch liegt, prüfe zuerst direkte Eltern-Kind-Beziehung und Bereichsnamen.

Prüfnachweis

- Zeilen und Spalten sind nachvollziehbar definiert.
- Semantische Elemente bleiben erhalten.
- Keine Datentabelle wird durch unsemantische Grid-Zellen ersetzt.
- Quellreihenfolge und sichtbare Reihenfolge widersprechen sich nicht.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

V1. Erzeuge ein automatisches Kartenraster mit `repeat(auto-fit, minmax(...))`.

V2. Vergleiche Linienplatzierung und grid-template-areas hinsichtlich Lesbarkeit.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Ordne Hauptlernfläche, Hilfebereich und Fortschrittsübersicht in einem verständlichen Grid an.

HALTE FEST

Grid-Container · Spalten/Zeilen · Bereiche · gap · Quellreihenfolge · Test

Mini-Check

SELBSTPRÜFUNG

Wann ist Grid geeigneter als Flexbox? Warum darf die sichtbare Rasterposition nicht die einzige Bedeutungsstruktur sein?

Modul 12: Flexbox und Grid sinnvoll kombinieren [↔ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst für verschiedene Teilprobleme begründet Flexbox oder Grid wählen und beide Layoutsysteme kombinieren.

Situation und jetzt tun

Eine Webseite besitzt nicht nur ein Layoutproblem: Die Gesamtseite braucht ein Raster, während Navigation, Karteninhalt und Schaltflächen meist entlang einer Achse angeordnet werden.

Untersuche dein Projekt und kennzeichne drei Container. Entscheide jeweils Flexbox, Grid oder normales Dokumentlayout und notiere den Grund.

MÖGLICHER STARTPROMPT

Ich muss zwischen Flexbox und Grid entscheiden. Gib mir nacheinander kleine Layoutsituationen. Lass mich jeweils ein Werkzeug auswählen und mit Achsen, Reihen oder Spalten begründen. Korrigiere erst danach.

Warum brauchst du das später?

Eine begründete Layoutentscheidung ist wichtiger als möglichst viele CSS-Eigenschaften. Sie macht das Projekt einfacher, stabiler und besser erklärbar.

Denkmodell und Fachwissen

- Normales Dokumentlayout bleibt für viele Textabschnitte die beste Ausgangslage.
- Flexbox steuert vor allem eine Hauptachse; Grid steuert Zeilen und Spalten gemeinsam.
- Container können verschachtelt werden: Ein Grid-Item kann selbst Flex-Container sein.
- Semantik bestimmt das HTML, Layout bestimmt CSS.
- Feste Höhen und absolute Positionierung sind für Hauptlayouts meist unnötig.
- Layoutentscheidungen werden an Inhalt, Zielgruppe und getesteten Breiten gemessen.

Pflichtauftrag · WEB-K12-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Erstelle eine Layoutkarte deines Projekts mit mindestens drei Containern.
2. Nutze Grid für den Hauptbereich und Flexbox für Navigation oder Karteninhalt.
3. Entferne mindestens eine unnötige feste Breite, Höhe oder absolute Positionierung.
4. Teste die Kombination in zwei Fensterbreiten.
5. Begründe für jeden Container die Wahl des Layoutsystems.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine kombinierte Lernagenten-Seite mit Layoutkarte, drei begründeten Entscheidungen und dokumentiertem Breitentest.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 3. Doppelstunde in Woche 11.

ZU HAUSE · Testprotokoll abschließen.

VERTIEFUNG · eine alternative Layoutlösung erstellen und vergleichen.

Gestufte Tutorhilfe

- Hilfe 1: Frage zuerst: eine Achse oder zwei?
- Hilfe 2: Setze Layout nur am jeweils zuständigen Container.
- Hilfe 3: Prüfe bei Problemen feste Größen und die direkte Eltern-Kind-Struktur.

Prüfnachweis

- Mindestens ein sinnvoller Flex- und ein Grid-Container sind vorhanden.
- Die Wahl ist fachlich begründet.
- HTML-Semantik wurde nicht für das Layout verbogen.
- Die Seite bleibt bei beiden Testbreiten nutzbar.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Löse dieselbe Kartenanordnung einmal mit Flexbox und einmal mit Grid und vergleiche Aufwand und Verhalten.
- V2.** Optimierte die Anordnung speziell für eine jüngere oder ältere Zielgruppe.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Erstelle den grundlegenden Seitenrahmen des Lernagenten: Grid für die Hauptzonen, Flexbox für Navigation und Komponenten.

HALTE FEST

Container · Layoutproblem · Werkzeug · Schlüsselregel · Begründung · Test

Mini-Check

SELBSTPRÜFUNG

Welche Frage hilft dir bei der Wahl zwischen Flexbox und Grid? Warum darf normales Dokumentlayout ebenfalls eine bewusste Entscheidung sein?

Modul 13: Responsive Design entwickeln und testen [↶ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst eine mobile-first-Oberfläche mit flexiblen Größen und gezielten Media Queries gestalten, prüfen und in einem Praxischeck nachweisen.

Situation und jetzt tun

Responsive Zielansichten vergleichen

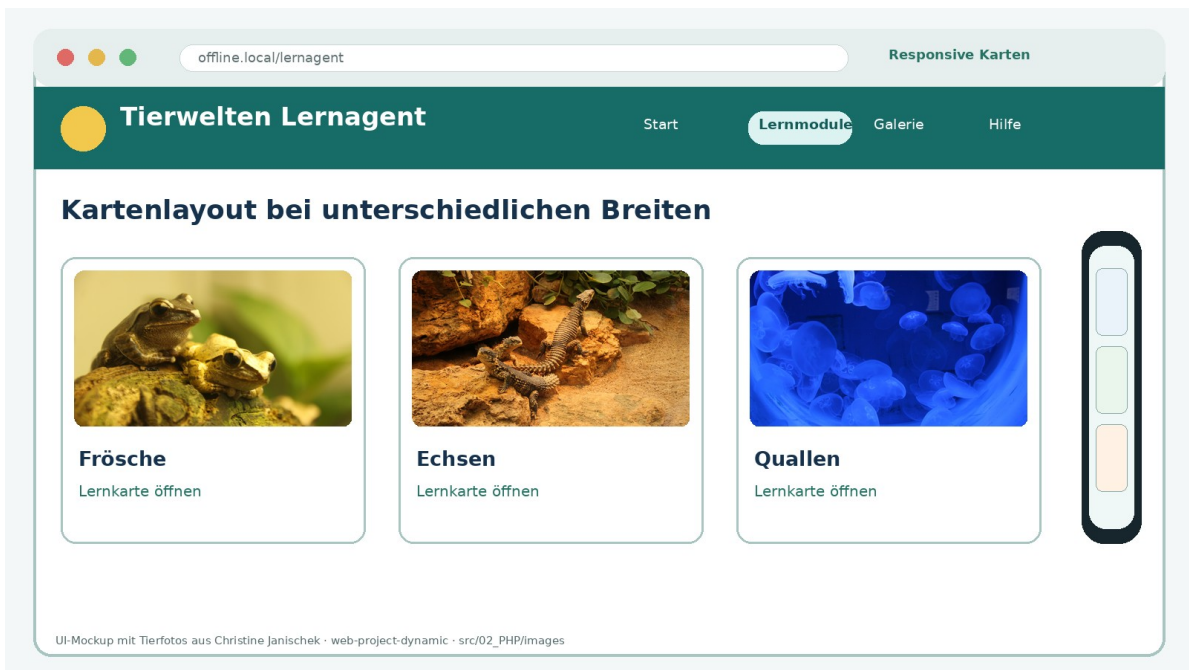


Abbildung 3: Lernkarten in breiter Ansicht mit Hinweis auf die einspaltige Mobilansicht.

Mockup-Check 3/4 – möglicher Zielentwurf. Markiere zuerst Anforderungen, leite danach HTML- und CSS-Entscheidungen ab und vergleiche am Ende Soll und Ist. Die Gestaltung darf begründet verändert werden. Nutze die Offline-Mockup-Vorlage.

Betrachte nicht Geräte, sondern den Platzbedarf des Inhalts.

1. Beschreibe, wann eine, zwei oder drei Spalten sinnvoll sind.
2. Leite flexible Größen und mindestens einen inhaltlich begründeten Breakpoint ab.
3. Vergleiche deine fertige Seite später mit allen drei Zuständen.

Der Lernagent funktioniert am Desktop, soll aber auch auf Smartphone und Tablet ohne Zoomen, Überlagerungen oder horizontales Scrollen nutzbar sein.

Betrachte die Seite bei sehr schmalen Fenstern. Notiere zuerst drei konkrete Probleme und priorisiere sie nach Nutzungswirkung.

MÖGLICHER STARTPROMPT

Ich optimiere eine Website responsive. Frage mich nacheinander nach Viewport, mobile first, flexiblen Größen, Umbruch, Media Query und Testbreiten. Gib nur Hinweise zu meinem konkreten Problem und keine komplette Lösung.

Warum brauchst du das später?

Responsive Design ist kein nachträglicher Effekt, sondern verbindet Inhalt, Layout, Typografie und Bedienbarkeit für unterschiedliche Geräte.

Denkmodell und Fachwissen

- Das Viewport-Metaelement ist Voraussetzung für erwartbares mobiles Verhalten.
- Mobile first beginnt mit der schmalen Grundansicht und erweitert bei ausreichendem Platz.
- Flexible Breiten, max-width, Flex-Umbruch und automatische Grid-Spalten lösen viele Fälle ohne Media Query.
- Media Queries werden eingesetzt, wenn Inhalt oder Layout einen begründeten Umschaltpunkt benötigen.
- Breakpoints orientieren sich am Inhalt, nicht nur an bestimmten Gerätemodellen.
- Bilder benötigen max-width: 100% und height: auto, wenn ihr Seitenverhältnis erhalten bleiben soll.
- Getestet werden mindestens schmale, mittlere und breite Ansicht sowie Zoom und Tastaturbedienung.

Pflichtauftrag · WEB-K13-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Ergänze beziehungsweise prüfe das Viewport-Metaelement.
2. Baue eine nutzbare mobile Grundansicht ohne feste Seitenbreite.
3. Nutze flexible Karten und Bilder, die nicht horizontal überlaufen.
4. Ergänze mindestens eine inhaltlich begründete Media Query für eine größere Ansicht.
5. Teste bei drei Breiten, mit Tastatur und bei vergrößerter Darstellung; dokumentiere Befund und Korrektur.
6. Bearbeite anschließend den Praxischeck WEB-K13-A02 ohne fertige KI-Lösung.

Praxischeck · WEB-K13-A02 · Responsive Reparatur

Zeit: 30 Minuten · Hilfsmittel: Skript und Offline-Prüfliste · Abgabe: korrigierte Dateien plus Kurzprotokoll · 20 Punkte

1. Erkläre die vorhandene HTML-Struktur in drei fachlich richtigen Sätzen.
2. Finde und behebe den Layoutfehler, der horizontalen Bildlauf erzeugt.
3. Ergänze eine mobile-first-Regel und einen inhaltlich begründeten Breakpoint.
4. Prüfe 360, 768 und 1280 Pixel und dokumentiere pro Breite einen Befund.
5. Begründe eine bewusste Abweichung vom Mockup oder stelle die Übereinstimmung fest.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine responsive Lernagenten-Seite, ein Testprotokoll mit drei Breiten und ein abgeschlossener individueller Praxischeck.

DEIN ARBEITSTEMPO

PFLICHT · gesamte Woche 12: Doppelstunde 1 mobile Grundansicht, Doppelstunde 2 Erweiterung und Tests, Doppelstunde 3 Praxischeck.

ZU HAUSE · nur zwischen Stunde 1 und 2 offene Fehler korrigieren.

VERTIEFUNG · clamp(), auto-fit oder zusätzlichen inhaltsbezogenen Breakpoint untersuchen.

Gestufte Tutorhilfe

- Hilfe 1: Suche zuerst das Element, das breiter als der Viewport wird.
- Hilfe 2: Entferne testweise feste width- und height-Werte.
- Hilfe 3: Setze die Media Query erst dort, wo die Grundansicht tatsächlich nicht mehr gut funktioniert.
- Hilfe 4: Prüfe Browserzoom und lange Texte, nicht nur voreingestellte Gerätesymbole.

Prüfnachweis

- Keine horizontale Scrollleiste bei der vereinbarten schmalen Testbreite.
- Text, Navigation und Bedienflächen bleiben nutzbar.
- Mindestens eine Media Query ist inhaltlich begründet.
- Drei Breitentests und ein Nachtest sind dokumentiert.
- Der Praxischeck wurde ohne fertige KI-Lösung bearbeitet.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

V1. Verwende clamp() für eine fließend begrenzte Überschrift und erkläre alle drei Werte.

V2. Teste eine sehr lange Überschrift und eine Karte mit ungewöhnlich viel Inhalt.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Mache die zentrale Lernagenten-Oberfläche mobile first nutzbar und lege verbindliche Testbreiten für spätere Module fest.

HALTE FEST

Grundansicht · Breakpoint-Grund · Testbreite · Problem · Änderung · Nachtest

Mini-Check

SELBSTPRÜFUNG

Was bedeutet mobile first? Wann ist eine Media Query nötig, und warum reicht die Gerätevorschau allein nicht als Test?

Modul 14: Tabellarische Daten semantisch darstellen [← Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst echte Zeilen-Spalten-Daten mit table, caption, thead, tbody, th und td zugänglich auszeichnen und responsiv einbetten.

Situation und jetzt tun

Der Lernagent soll einen Wochenlernplan mit Tagen, Themen, Zeit und Status darstellen. Diese Beziehungen müssen auch ohne visuelles Raster verständlich bleiben.

Entscheide für Lernkarten, einen Stundenplan und eine zweispaltige Begriffsliste jeweils zwischen normalem Text, CSS Grid und HTML-Tabelle. Begründe.

MÖGLICHER STARTPROMPT

Ich lerne semantische Datentabellen. Frage mich nach caption, thead, tbody, tr, th, td und scope. Prüfe zuerst, ob meine Daten überhaupt tabellarisch sind. Erzeuge keine fertige Tabelle.

Warum brauchst du das später?

Tabellen sind für echte Datenbeziehungen richtig und zugänglich. CSS Grid gestaltet Layouts, ersetzt aber nicht die Semantik einer Datentabelle.

Denkmodell und Fachwissen

- table wird für Daten verwendet, nicht für das allgemeine Seitenlayout.
- caption beschreibt Zweck und Inhalt der Tabelle.
- thead und tbody gliedern Kopf- und Datenbereich.
- th kennzeichnet Kopfzellen; scope="col" oder scope="row" verdeutlicht einfache Beziehungen.
- colspan und rowspan sind möglich, erhöhen aber die Komplexität und sollten begründet eingesetzt werden.
- Auf kleinen Bildschirmen kann ein umgebender Bereich horizontal scrollen; wichtige Inhalte dürfen nicht abgeschnitten werden.

Pflichtauftrag · WEB-K14-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Erstelle eine Tabelle für einen Wochenlernplan mit mindestens vier Spalten und vier Datenzeilen.
2. Nutze caption, thead, tbody und passende th-Elemente mit scope.
3. Gestalte Abstände, Rahmen und Zebra- oder Hover-Effekt ohne die Lesbarkeit zu verschlechtern.
4. Lege die Tabelle in einen benannten Scrollbereich für schmale Ansichten.
5. Teste Struktur, Tastaturzugang und schmale Ansicht.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine semantische und responsiv nutzbare Datentabelle mit erklärter Kopfzellenstruktur und Testnachweis.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 1. Doppelstunde in Woche 13.

ZU HAUSE · nur Tabelle oder Test fertigstellen.

VERTIEFUNG · komplexere Kopfstruktur untersuchen, ohne sie zwingend umzusetzen.

Gestufte Tutorhilfe

- Hilfe 1: Prüfe zuerst, ob sich jede Information durch Zeile und Spalte eindeutig zuordnen lässt.
- Hilfe 2: Beginne mit caption und einfacher Kopfzeile.
- Hilfe 3: Wenn die Tabelle zu breit ist, schneide keine Spalten ab; ermögliche kontrolliertes Scrollen.

Prüfnachweis

- Die Daten sind tatsächlich tabellarisch.
- Kopf- und Datenzellen sind korrekt ausgezeichnet.
- caption und scope unterstützen das Verständnis.
- Alle Inhalte bleiben auf schmaler Ansicht erreichbar.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Vergleiche die Tabelle mit einer rein visuellen Grid-Nachbildung hinsichtlich Semantik.
- V2.** Ergänze eine Zeilenkopf-Spalte und erkläre `scope="row"`.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Stelle Lernplan oder Lernfortschritt des Lernagenten als echte Datentabelle dar.

HALTE FEST

Tabellenzweck · caption · Spaltenköpfe · Zeilenköpfe · responsive Lösung · Test

Mini-Check

SELBSTPRÜFUNG

Wann ist table fachlich richtig und wann Grid? Welche Aufgabe haben caption und scope?

Modul 15: Bilder und Medien zugänglich einbinden [← Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst Bilder passend auswählen, optimieren, mit korrekten Pfaden einbinden und durch Alternativtexte beziehungsweise Bildunterschriften verständlich machen.

Situation und jetzt tun

Die Lernwebsite soll visuell ansprechender werden, darf aber weder unnötig groß noch ohne Internet unvollständig sein.

Wähle für Logo, erklärende Grafik und rein dekorative Fläche jeweils Dateityp, Alternativtextstrategie und sinnvolle Größenbegrenzung.

MÖGLICHER STARTPROMPT

Ich plane Bilder für eine Offline-Lernwebsite. Frage mich nach Zweck, Dateiformat, Dateigröße, src, alt, figure, figcaption, width/height und Urhebernachweis. Gib keine fertige Bilderliste vor.

Warum brauchst du das später?

Medien beeinflussen Ladezeit, Verständlichkeit, Barrierefreiheit und Rechte. Diese Entscheidungen werden auch für spätere dynamische Inhalte benötigt.

Denkmodell und Fachwissen

- img bindet ein Bild über src ein; alt beschreibt die Funktion oder Information im Kontext.
- Dekorative Bilder erhalten ein leeres alt-Attribut, informative Bilder einen knappen passenden Alternativtext.
- figure und figcaption verbinden Medium und sichtbare Erläuterung.
- SVG eignet sich für skalierbare Grafiken, WebP/JPEG häufig für Fotos, PNG für bestimmte verlustfreie oder transparente Grafiken.
- width und height im HTML können Platz reservieren; CSS begrenzt die Darstellung responsiv.
- Dateinamen, Kompression, Quelle und Lizenz gehören zur Medienorganisation.
- Externe eingebettete Medien widersprechen einem vollständig offline nutzbaren Paket.

Pflichtauftrag · WEB-K15-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Ergänze Logo, erklärende Grafik und ein dekoratives Element mit korrekten lokalen Pfaden.
2. Formuliere passende Alternativtexte beziehungsweise alt="" und begründe sie.
3. Nutze für die erklärende Grafik figure und figcaption.
4. Begrenze Bilder responsiv und prüfe ihr Verhalten bei drei Breiten.
5. Dokumentiere Dateiformat, Dateigröße, Quelle, Lizenz und vorgenommene Optimierung.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine offline funktionierende Medienseite mit drei unterschiedlich begründeten Bildrollen, responsiver Darstellung und Mediennachweis.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 2. Doppelstunde in Woche 13.

ZU HAUSE · Mediennachweis vervollständigen.

VERTIEFUNG · picture oder srcset für alternative Bildgrößen untersuchen.

Gestufte Tutorhilfe

- Hilfe 1: Frage bei alt zuerst: Welche Information fehlt ohne das Bild?
- Hilfe 2: Prüfe Pfad und Großschreibung zeichenweise.
- Hilfe 3: Verkleinere die Bilddatei selbst und nicht nur ihre CSS-Anzeige.

Prüfnachweis

- Alle Medien funktionieren offline.
- Alternativtexte passen zur jeweiligen Bildrolle.
- Bilder verzerren nicht und laufen nicht über.
- Quelle, Lizenz und Bearbeitung sind dokumentiert.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

V1. Erstelle mit picture eine alternative Darstellung für schmale Ansichten.

V2. Vergleiche zwei Bildformate hinsichtlich Qualität und Dateigröße.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Wähle ein Logo und eine erklärende Lernillustration für den Lernagenten und dokumentiere deren barrierefreie und rechtliche Verwendung.

HALTE FEST

Bildrolle · Datei/Pfad · alt/figcaption · Format/Größe · Quelle/Lizenz · Test

Mini-Check

SELBSTPRÜFUNG

Wann ist alt leer, wann beschreibend? Warum genügt es nicht, ein großes Bild nur mit CSS kleiner anzuzeigen?

Modul 16: Mediengalerie und dezente Effekte ↵

Inhalt

DEIN KAPITELSTART

Ziel: Du kannst eine flexible, zugängliche Galerie mit Grid oder Flexbox gestalten und Effekte zurückhaltend sowie bedienbar einsetzen.

Situation und jetzt tun

Galerie mit sichtbarem Tastaturfokus

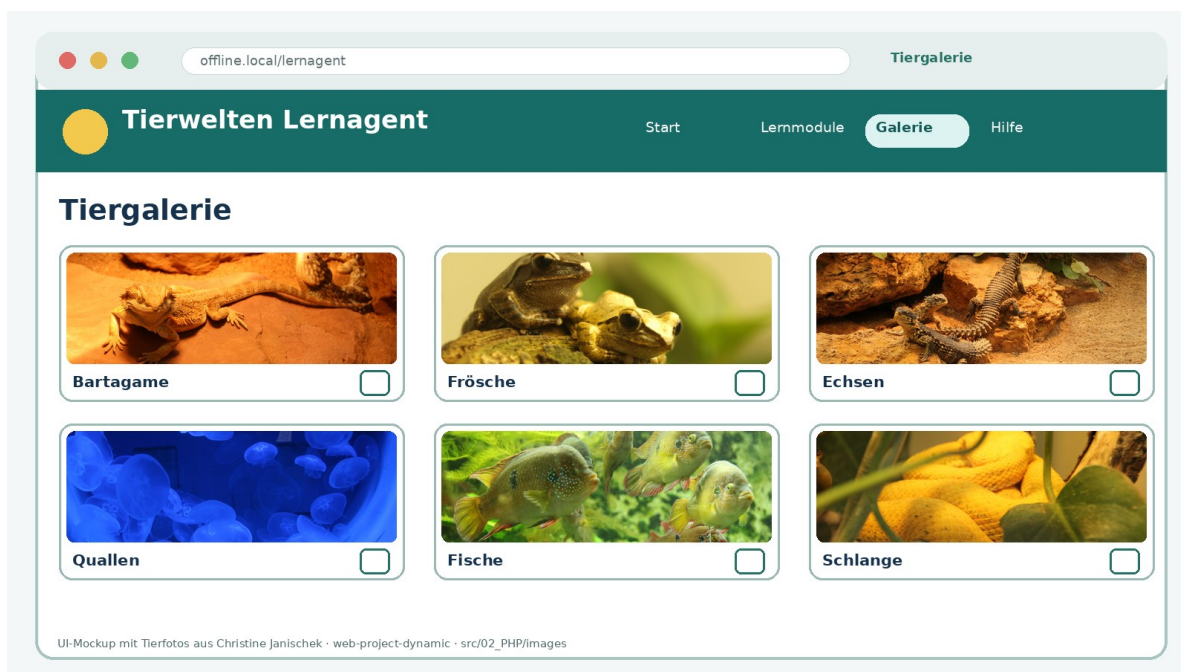


Abbildung 4: Responsive Tiergalerie mit sechs Motiven und sichtbaren Bedienelementen.

Mockup-Check 4/4 – möglicher Zielentwurf. Markiere zuerst Anforderungen, leite danach HTML- und CSS-Entscheidungen ab und vergleiche am Ende Soll und Ist. Die Gestaltung darf begründet verändert werden. Nutze die Offline-Mockup-Vorlage.

Die Grafik zeigt nicht nur Gestaltung, sondern auch einen wichtigen Bedienzustand.

1. Bestimme semantische Struktur, Karteninhalt und Ziel jeder Aktion.
2. Plane gleichwertige hover- und focus-visible-Zustände.
3. Prüfe, ob alle Informationen ohne Maus und ohne Bewegungseffekt erreichbar bleiben.

Mehrere Lernmodule sollen über Bilder oder Vorschau-elemente auswählbar sein. Die Galerie muss auf verschiedenen Breiten funktionieren und darf Informationen nicht nur beim Darüberfahren zeigen.

Skizziere eine Galerie mit sechs Einträgen. Kennzeichne Bild, Titel, Beschreibung, Link und Fokuszustand eines Eintrags.

MÖGLICHER STARTPROMPT

Ich plane eine zugängliche Mediengalerie. Frage mich nach semantischer Struktur, Grid oder Flexbox, auto-fit/minmax, Alternativtext, Linkfläche, hover, focus-visible und reduzierter Bewegung. Gib keine fertige Galerie aus.

Warum brauchst du das später?

Galerien verbinden nahezu alle bisherigen Kompetenzen: Semantik, Bilder, Layout, Responsivität, Interaktion, Kontrast und Rechte.

Denkmodell und Fachwissen

- Eine Galerie kann als Liste eigenständiger Karten aufgebaut werden.
- `repeat(auto-fit, minmax(...))` erzeugt häufig ein flexibles Kartenraster ohne viele Breakpoints.
- Hover-Effekte benötigen einen gleichwertigen Fokuszustand.
- Wichtige Inhalte dürfen nicht nur bei hover sichtbar werden.
- Übergänge sollten kurz und dezent sein; `prefers-reduced-motion` respektiert reduzierte Bewegung.
- Bildquellen und Lizenzen bleiben auch in einer Galerie erforderlich.

Pflichtauftrag · WEB-K16-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Erstelle eine Galerie mit mindestens sechs Lernkarten und sinnvollen Überschriften.
2. Nutze Grid oder Flexbox mit flexibler Mindestbreite und `gap`.
3. Mache jede Karte verständlich per Tastatur erreichbar.
4. Ergänze einen dezenten hover- und gleichwertigen focus-visible-Effekt.
5. Prüfe schmale, mittlere und breite Ansicht sowie reduzierte Bewegung.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Eine responsive Galerie mit sechs zugänglichen Karten, nachvollziehbarer Layoutentscheidung und vollständigem Testprotokoll.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 3. Doppelstunde in Woche 13.

ZU HAUSE · nur Galerie fertigstellen.

VERTIEFUNG · Bewegung über prefers-reduced-motion anpassen.

Gestufte Tutorhilfe

- Hilfe 1: Baue zuerst eine einzelne semantisch vollständige Karte.
- Hilfe 2: Erst danach entsteht der Galerie-Container.
- Hilfe 3: Teste Fokus ohne Maus und vergleiche ihn mit hover.

Prüfnachweis

- Alle Karten besitzen Titel und verständliches Ziel.
- Galerie passt sich ohne abgeschnittene Inhalte an.
- Hover und Fokus sind gleichwertig.
- Reduzierte Bewegung und Bildnachweise wurden berücksichtigt.

Vertiefung - weiterdenken und übertragen**VERTIEFUNG**

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

V1. Ergänze eine CSS-Regel für prefers-reduced-motion.

V2. Lass eine andere Person zwei Lernmodule ausschließlich per Tastatur auswählen.

Transfer zum Learning Agent**AGENTENBAUSTEIN**

Nutze die Galerie als visuelle Themenauswahl des Lernagenten.

HALTE FEST

Karteninhalt · Layoutregel · Mindestbreite · Hover/Fokus · Bewegung · Test

Mini-Check**SELBSTPRÜFUNG**

Warum reicht hover nicht aus? Wie erzeugt auto-fit mit minmax ein flexibles Raster?

Modul 17: Barrierefreiheit, Quellen und rechtliche Mindestprüfung [↔ Inhalt](#)

DEIN KAPITELSTART

Ziel: Du kannst dein Projekt mit einer verständlichen Prüfroutine auf Zugänglichkeit, Quellen, Bildrechte, Impressums- und Datenschutzfragen untersuchen.

Situation und jetzt tun

Der Lernagent soll präsentiert oder möglicherweise veröffentlicht werden. Vorher müssen technische, gestalterische und rechtliche Risiken sichtbar gemacht werden.

Erstelle drei Spalten mit 'technisch', 'zugänglich' und 'rechtlich'. Ordne mindestens neun bisherige Prüfungen zu und markiere offene Fragen.

MÖGLICHER STARTPROMPT

Ich prüfe eine Lernwebsite vor einer möglichen Veröffentlichung. Stelle mir Fragen zu Tastatur, Überschriften, Alternativtexten, Kontrast, Quellen, Lizenzen, Impressum und Datenschutz. Weise bei Rechtsfragen ausdrücklich darauf hin, dass eine aktuelle schulische oder fachkundige Prüfung nötig ist.

Warum brauchst du das später?

Veröffentlichungsqualität entsteht nicht erst am Ende. Die Prüfroutine schützt Nutzende, Urheberinnen und Urheber sowie die verantwortliche Stelle.

Denkmodell und Fachwissen

- Barrierefreiheit umfasst wahrnehmbare, bedienbare, verständliche und robuste Inhalte.
- Tastatur, Fokus, Überschriften, Linktexte, Alternativtexte, Kontrast, Zoom und Bewegung werden manuell geprüft.
- Fremde Texte, Bilder, Schriften und Medien benötigen nachvollziehbare Quellen- und Nutzungsrechte.
- KI-generierte oder KI-bearbeitete Inhalte werden transparent dokumentiert und fachlich geprüft.
- Ob ein Impressum oder eine Datenschutzerklärung erforderlich ist, hängt vom konkreten Betreiber, Zweck, Hosting und den verarbeiteten Daten ab.
- Eine rein statische Offline-Seite verarbeitet normalerweise weniger Daten als eine veröffentlichte Seite mit Formularen, Analyse- oder Fremddiensten.
- Rechtliche Hinweise im Skript ersetzen keine aktuelle Rechtsberatung oder schulische Freigabe.

Pflichtauftrag · WEB-K17-A01 · Grundlagen - verbindlich

AUFTRAG

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Führe den vollständigen Tastatur-, Fokus-, Überschriften-, Link-, Bild-, Kontrast- und Zoomtest durch.
2. Erstelle ein Quellen- und Lizenzverzeichnis für alle fremden und KI-gestützten Inhalte.
3. Kennzeichne, welche Funktionen bei einer Veröffentlichung Daten übertragen oder speichern könnten.
4. Formuliere einen Entwurf für Anbieterkennzeichnung und Datenschutzhinweise mit deutlich markierten, noch zu prüfenden Angaben.
5. Behebe technische Fehler und dokumentiere rechtliche Prüffragen für die verantwortliche Stelle.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Ein Qualitäts- und Freigabeprotokoll mit technischen Nachweisen, Quellenliste, KI-Transparenz und klar gekennzeichneten rechtlichen Prüffragen.

DEIN ARBEITSTEMPO

PFLICHT · bis Ende der 1. Doppelstunde in Woche 14.

ZU HAUSE · fehlende Quellenbelege ergänzen.

VERTIEFUNG · Prüfraster um konkrete WCAG-Erfolgskriterien erweitern.

Gestufte Tutorhilfe

- Hilfe 1: Trenne nachweisbare technische Fehler von rechtlichen Prüffragen.
- Hilfe 2: Prüfe jede externe Ressource und jede Datenübertragung.
- Hilfe 3: Verwende keine erfundenen Personen- oder Kontaktdaten als scheinbar fertiges Impressum.

Prüfnachweis

- Tastatur und Fokus funktionieren vollständig.
- Quellen, Lizenzen und KI-Beiträge sind dokumentiert.
- Datentransfers und offene Rechtsfragen sind sichtbar.
- Keine ungeprüfte rechtliche Behauptung wird als sichere Freigabe dargestellt.

Vertiefung - weiterdenken und übertragen

VERTIEFUNG

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Ordne deine Befunde den vier Grundprinzipien der Barrierefreiheit zu.
- V2.** Entwickle eine Freigabecheckliste für eine spätere GitHub-Pages- oder Schulserver-Veröffentlichung.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Erstelle die Qualitäts-, Quellen- und Freigabeseite für den Lernagenten. Noch offene rechtliche Angaben bleiben ausdrücklich als Prüfauftrag markiert.

HALTE FEST

Prüfbereich · Nachweis · Befund · Änderung · Quelle/Lizenz · offene Freigabe

Mini-Check

SELBSTPRÜFUNG

Welche Prüfungen kannst du selbst sicher nachweisen und welche Fragen müssen vor einer Veröffentlichung aktuell geklärt werden?

Modul 18: Abschlussprojekt: Oberfläche eines persönlichen Lernagenten ↔ Inhalt

DEIN KAPITELSTART

Ziel: Du kannst eine statische, responsive und zugängliche Lernagenten-Oberfläche planen, umsetzen, testen, dokumentieren und fachlich verteidigen.

Situation und jetzt tun

Alle Einzelbausteine werden zu einem nachvollziehbaren Produkt verbunden. Die Website simuliert die Oberfläche eines Lernagenten; echte dynamische Verarbeitung folgt erst in einem späteren Kurs.

Formuliere ein Mini-Pflichtenheft: Zielgruppe, Zweck, vier Seiten beziehungsweise Bereiche, Pflichtfunktionen, Designregeln und Qualitätsnachweise.

MÖGLICHER STARTPROMPT

Ich entwickle das Abschlussprojekt einer statischen Lernagenten-Website. Agiere als prüfender Lerncoach: Frage zuerst nach Zielgruppe, Inhalt, Semantik, Layout, Responsivität, Barrierefreiheit und Tests. Gib nur Hinweise und fordere für jede Entscheidung eine Begründung. Schreibe das Projekt nicht für mich.

Warum brauchst du das später?

Das Abschlussprojekt zeigt nicht nur Quellcode, sondern den gesamten Entwicklungsweg. Es bildet zugleich die Oberfläche, die später mit JavaScript oder PHP dynamisch erweitert werden kann.

Denkmodell und Fachwissen

- Das Projekt beginnt mit einem kleinen Pflichtenheft und einer Seiten- beziehungsweise Komponentenstruktur.
- HTML trägt Inhalt und Semantik, CSS Gestaltung und Layout; dynamisches Verhalten wird nur sichtbar vorbereitet.
- Mindestens Flexbox und Grid werden jeweils an einer passenden Stelle eingesetzt.
- Die mobile Grundansicht, mindestens eine größere Ansicht und alle Medien werden geprüft.
- Quellen, Lizenzen, KI-Nutzung und offene rechtliche Fragen werden dokumentiert.
- Abgabe umfasst Projektordner, Testprotokoll, kurze Dokumentation und persönliche Reflexion.
- Die Verteidigung prüft, ob Entscheidungen und eigener Quellcode erklärt werden können.

Pflichtauftrag · WEB-K18-A01 · Grundlagen - verbindlich**AUFTRAG**

Bearbeite den folgenden Pflichtauftrag vollständig. Verändere nur Quellcode, den du anschließend erklären und im Browser prüfen kannst.

1. Erstelle ein Mini-Pflichtenheft und eine Skizze für deinen Lernagenten.
2. Setze mindestens Start, Themenauswahl, Lernaufgabe, Hilfebereich und Lernfortschritt statisch um.
3. Verwende semantisches HTML, ein konsistentes Designsystem, Flexbox, Grid und responsive Regeln.
4. Integriere eine Datentabelle oder Galerie, passende Medien und eine zugängliche Navigation.
5. Führe die vollständige Qualitätscheckliste durch und überarbeite mindestens drei dokumentierte Befunde.
6. Erstelle eine Projektdokumentation und bereite eine kurze Demonstration mit fachlicher Verteidigung vor.

DEINE BEARBEITUNG

Dokumentiere hier Planung, wichtige Quellcodeausschnitte, Begründungen und Testergebnisse. Deine vollständigen Projektdateien speicherst du im vorgesehenen Modulordner.

Erwartetes Lernprodukt

Ein vollständiger Offline-Projektordner mit responsiver Lernagenten-Website, Pflichtenheft, Test- und Quellenprotokoll, Dokumentation und Reflexion.

DEIN ARBEITSTEMPO

PFLICHT · Doppelstunden 2 und 3 in Woche 14: zuerst Planung und Prototyp, danach Test, Überarbeitung und Verteidigung.

ZU HAUSE · keine reguläre Zusatzarbeit; nur individuell vereinbarte Fertigstellung.

VERTIEFUNG · zusätzliche Komponente oder zweite responsive Variante, ohne dynamische Programmierung vorwegzunehmen.

Gestufte Tutorhilfe

- Hilfe 1: Sichere zuerst die Pflichtbereiche, bevor du Effekte ergänzt.
- Hilfe 2: Prüfe jeden Abschnitt mit der Frage: Inhalt, Semantik, Gestaltung oder Verhalten?
- Hilfe 3: Nutze deine Modul-Checklisten und behebe jeweils nur einen Befund.
- Hilfe 4: Für die Verteidigung markierst du drei Stellen, an denen du eine bewusste Entscheidung erklären kannst.

Prüfnachweis

- Alle Pflichtbereiche funktionieren vollständig offline.
- Semantik, Navigation, Flexbox, Grid und responsive Regeln sind sinnvoll eingesetzt.
- Tastatur, Fokus, Kontrast, Bilder, Links und Breiten wurden geprüft.
- Quellen, Lizenzen und KI-Unterstützung sind transparent.
- Die Schülerin oder der Schüler kann ausgewählten Quellcode und Entscheidungen erklären.

Vertiefung - weiterdenken und übertragen**VERTIEFUNG**

Wähle mindestens eine Aufgabe, wenn dein Pflichtteil vollständig geprüft ist.

- V1.** Entwickle eine zweite Darstellung für eine andere Zielgruppe.
- V2.** Markiere Schnittstellen, an denen später JavaScript oder PHP Verhalten ergänzen könnte, ohne es bereits zu implementieren.

Transfer zum Learning Agent

AGENTENBAUSTEIN

Führe alle Agentenbausteine zu einer konsistenten statischen Oberfläche zusammen. Sie wird im nächsten Skript zur Ausgangsbasis dynamischer Programmierung.

HALTE FEST

Pflichtenheft · Dateistruktur · Designsystem · Layoutentscheidungen · Tests · Quellen/KI · Reflexion

Mini-Check

SELBSTPRÜFUNG

Welche drei Entscheidungen belegen deine Kompetenz am besten? Was bleibt bewusst statisch und wie könnte es später dynamisch erweitert werden?

Verbindliche Prüfungsübersicht [← Inhalt](#)

Die folgende Übersicht grenzt das prüfungsrelevante Pflichtwissen von möglichen Vertiefungen ab. Konkrete Aufgabenstellungen dürfen unbekannte Inhalte verwenden, prüfen aber die hier genannten Kompetenzen.

Kompetenzbereich	Verbindlich nachweisen	Nicht automatisch Pflicht
Web und Dateien	Browser, Website, Webseite, index.html, Dateinamen, relative und absolute Pfade, systematischer Test	GitHub-Befehle, Hosting-Konfiguration
HTML	Grundgerüst, UTF-8, Sprache, Überschriften, Absätze, Listen, Links, Bilder, semantische Seitenbereiche	komplexe Formulare, dynamische Templates
CSS-Grundlagen	Einbindung, Regelaufbau, Element- und Klassenselektoren, Kaskade, Vererbung, Farben und Typografie	komplexe Spezifitätsberechnungen, Präprozessoren
Box-Modell	Content, Padding, Border, Margin, border-box, relative Größen, max-width	komplexe mathematische CSS-Funktionen
Layout	Flex-Container und Items, Achsen, gap, wrap; Grid-Container, Zeilen,	Subgrid und experimentelle

Kompetenzbereich	Verbindlich nachweisen	Nicht automatisch Pflicht
	Spalten, fr; begründete Auswahl	Layoutfunktionen
Responsive Design	Viewport, mobile first, flexible Medien, Media Query, Tests bei mehreren Breiten	viele gerätespezifische Breakpoints
Daten und Medien	semantische Tabelle; Bilder, alt, figure/figcaption, Formate, Quellen und Lizenzen	aufwendige Bildbearbeitung, Video-Schnitt
Qualität	Validierung, Tastatur, Fokus, Überschriften, Links, Kontrast, Zoom, Testprotokoll	vollständiges professionelles Accessibility-Audit
Transfer	Quellcode erklären, Fehler finden, Entscheidung begründen, unbekannte Situation übertragen	dynamische Programmierung mit PHP oder JavaScript

Mögliche Formate für Leistungsnachweise

- Kurzcheck: Begriffe erklären, Pfade bilden, Quellcode lesen und kleine Fehler begründen.
- Praxischeck: Eine bereitgestellte Seite vervollständigen, gestalten und bei vorgegebenen Breiten testen.
- Klassenarbeit: Analyse- und Implementierungsanteil mit transparenten Punkten für Struktur, CSS, Layout, Test und Begründung.
- Projektleistung: Pflichtenheft, Zwischenstände, formatives KI- oder Peer-Feedback, Überarbeitung, Demonstration, Verteidigung und Reflexion.

BEWERTUNGSGRUNDSATZ

Ein optisch ähnliches Ergebnis allein genügt nicht. Bewertet werden ebenso semantische Struktur, nachvollziehbarer Quellcode, responsive Funktion, systematisches Testen und die Fähigkeit, eigene Entscheidungen zu erklären.

Abschlussreflexion

1. Welche drei verpflichtenden Kompetenzen beherrschst du sicher? Nenne jeweils einen Nachweis.
2. Bei welchem Fehler hilft dir inzwischen eine feste Prüfstrategie? Beschreibe sie.
3. Welcher Teil deines Lernagenten ist besonders gut auf eine spätere dynamische Erweiterung vorbereitet?
4. Welche Unterstützung hast du genutzt und wie hast du die Ergebnisse selbst geprüft?

Quellen und Arbeitsgrundlagen [↩ Inhalt](#)

- Bildungsplan Berufskolleg Wirtschaftsinformatik, 13. März 2025: BPE 4 Webentwicklung.
- Wochenplan BKWI1, Schuljahr 2026/2027, sechs Wochenstunden.
- Repository web-project-static: <https://github.com/ChristineJanischek/web-project-static> (Ideen- und Materialquelle für die Lehrkraft; für den Einstieg nicht erforderlich).
- Repository web-project-dynamic: <https://github.com/ChristineJanischek/web-project-dynamic> (Ausblick und Materialquelle; für den Einstieg nicht erforderlich).
- Materialpaket Webentwicklung 2025 mit Informationsblättern, Aufgaben, Lösungen und Quellcodeprojekten:
<https://edublog.emotionalspirit.de/index.php/2025/10/07/webentwicklung/> (didaktisch und technisch aktualisiert verwendet).
- Vorhandene Lernskripte zu Tabellenkalkulation, Geschäftsprozessen und Git/GitHub.
- Tierfotografien der eingebetteten UI-Mockups: Christine Janischek, Repository web-project-dynamic, Ordner src/02_PHP/images; Nutzung gemäß Repository-Lizenz NC School-Only 1.0.

Lizenz und Nutzungsbedingungen

Christine Janischek NC School-Only 1.0 - Lernskriptfassung

Diese Lizenz gilt für das vorliegende Lernskript, soweit einzelne Bestandteile nicht ausdrücklich anders gekennzeichnet sind.

Copyright und Website

Copyright (c) Christine Janischek
Website: <https://emotionalspirit.de>

1. Erlaubte Nutzung

Dieses Lernskript darf kostenfrei genutzt, vervielfältigt, bearbeitet und weitergegeben werden, sofern sämtliche nachfolgenden Bedingungen eingehalten werden.

2. Namensnennung

Bei jeder Nutzung und Weitergabe ist sichtbar anzugeben:
„Christine Janischek – <https://emotionalspirit.de>“

3. Nichtkommerzielle Nutzung

Eine kommerzielle Nutzung ist nicht gestattet.

4. Staatliches Schulwesen

Die Nutzung ist ausschließlich im Zusammenhang mit staatlichen Schulsystemen gestattet.

5. Bearbeitungen

Änderungen sind als solche kenntlich zu machen. Dieser Lizenzhinweis muss bei jeder Weitergabe vollständig erhalten bleiben.

6. Andere Nutzungen

Jede Nutzung außerhalb dieser Bedingungen bedarf der vorherigen ausdrücklichen schriftlichen Erlaubnis der Rechteinhaberin.

7. Keine Markenrechte

Rechte an Marken, Logos oder Kennzeichen werden nicht eingeräumt.

8. Fremdmaterialien

Abbildungen, Quelltexte, Marken oder sonstige Inhalte Dritter können eigenen Bedingungen unterliegen. Diese Lizenz gilt nur für Bestandteile, an denen Christine Janischek die erforderlichen Rechte besitzt.

9. Haftungsausschluss

Das Werk wird ohne Gewähr bereitgestellt. Soweit gesetzlich zulässig, wird keine Haftung für Richtigkeit, Vollständigkeit, Eignung oder Rechtsmängel übernommen.

10. Erlöschen der Nutzungsrechte

Bei einem Verstoß gegen diese Bedingungen erlöschen die eingeräumten Nutzungsrechte automatisch.

Hinweis: Diese individuell formulierte Lizenz ist keine Creative-Commons- und keine Open-Source-Lizenz.

Anhang A: Kompetenz-, Aufgaben- und Materialmatrix [↔ Inhalt](#)

Die Matrix ist die verbindliche Prüfkette des Skripts. Jede bewertbare Aufgabe besitzt eine stabile ID. Die Dateinamen in den Paketen verwenden dieselbe ID.

Planbezug	Kompetenz	Aufgaben-ID	Material	Prüfnachweis
BPE 4.1	Grundgerüst, Projektstruktur und Pfade	WEB-K01-A01 bis WEB-K03-A01	K01-K03	Bestandsaufnahme, Pfadtest, Browsercheck
BPE 4.1	Semantik, Text, Navigation und Validierung	WEB-K04-A01 bis WEB-K06-A01	K04-K06	Strukturtest, Linktest, Tastaturtest
BPE 4.1	CSS, Box-Modell, Flexbox und Grid	WEB-K07-A01 bis WEB-K12-A01	K07-K12	CSS-Test, Layoutkarte, Breitentest
BPE 4.1	Responsive Design und Medienabfragen	WEB-K13-A01, WEB-K13-A02	K13	360/768/1280 px, Praxischeck
BPE 4.1	Tabellen und Multimedia	WEB-K14-A01 bis WEB-K16-A01	K14-K16	Semantik-, Alttext-, Fokus- und

Planbezug	Kompetenz	Aufgaben-ID	Material	Prüfnachweis
				Medientest
BPE 4.2	Corporate Design und Zielgruppe	WEB-K08-A01, WEB-K18-A01	K08, K18	Designblatt, Zielgruppenbegründung
BPE 4.3	Barrierefreiheit und rechtliche Einordnung	WEB-K06-A01, WEB-K15-A01 bis WEB-K18-A01	K06, K15-K18	Freigabe-, Quellen- und Lizenzprotokoll
BPE 4.1-4.3	Entwurf → Mockup → Umsetzung → Vergleich	WEB-K01-A01, WEB-K05-A01, WEB-K13-A01, WEB-K16-A01, WEB-K18-A01	Mockup-Werkstatt	Raster, Sollansichten, Soll-Ist-Protokoll

Anhang B: Verbindliche Mockup-Routine [↩ Inhalt](#)

Die Mockups sind keine Dekoration und keine pixelgenaue Kopiervorlage. Sie machen Anforderungen sichtbar. Für die vier vorgegebenen Ansichten und für den eigenen Abschlussentwurf gilt dieselbe Routine.

1. Wahrnehmen: Bereiche, Inhalte, Aktionen, Hierarchie und sichtbare Zustände markieren.
2. Ableiten: semantische HTML-Elemente, Layoutsystem, Abstände, Farben, Breakpoints und Bedienzustände begründen.
3. Planen: Desktop- und Mobilraster in der Offline-Vorlage skizzieren; Inhaltsreihenfolge ohne CSS festlegen.
4. Umsetzen: zuerst semantisches HTML, danach CSS; Flexbox und Grid nach Aufgabe des Bereichs auswählen.
5. Prüfen: bei etwa 360, 768 und 1280 Pixel, 200 % Zoom, Tastatur und reduzierter Bewegung testen.
6. Vergleichen: Abweichungen vom Mockup als Fehler, bewusste Verbesserung oder technisch notwendige Anpassung einordnen.
7. Dokumentieren: Screenshot oder Skizze, Testprotokoll und zwei begründete Entscheidungen abgeben.

Abnahmekriterien: vollständige Inhalte, klare Hierarchie, semantische Struktur, kein horizontaler Bildlauf, sichtbarer Tastaturfokus, ausreichende Kontraste, verständliche Navigation und begründete Abweichungen.

Anhang C: Zeitmodell [↩ Inhalt](#)

Der Bildungsplan weist 40 Unterrichtsstunden aus. Der Pflichtlernweg umfasst deshalb 20 Doppelstunden. Die im Wochenplan zusätzlich sichtbare 21. Doppelstunde in Woche 12 dient ausschließlich Leistungsnachweis, Nachschreiben oder organisatorischem Puffer und erweitert den Pflichtstoff nicht.