

## Praktikum 3 (Tabellen, Multimedia, Formulare)

Im Rahmen des Praktikums entwickeln wir eine Web-Anwendung, welche wir Schritt für Schritt mit weiteren Anforderungen, Funktionen und Technologien erweitern.

### Meilenstein 1: "HTML"

Sobald Sie Praktikum 2 und 3 vollständig bearbeitet haben, haben Sie Meilenstein 1 erreicht! Sprechen Sie uns (Sven Jörges, Andreas Harrer, Darius Schippritt oder Yahya Hasan Bek) im Praktikum an. Wir schauen uns dann gemeinsam Ihren Stand in einer kleinen Abnahme an. Dabei muss Ihre Gruppe, in welcher Sie die Praktika bearbeitet haben, vollständig anwesend sein. Für diesen Meilenstein sind insgesamt 2 Bonuspunkte erreichbar. Es folgen noch zwei weitere Meilensteine.

### Hinweis zum entstehenden Code:

- Zur Verwaltung und zum kollaborativen Bearbeiten Ihres Quellcodes empfehlen wir Ihnen die Nutzung von Git (z.B. in Verbindung mit dem [GitLab-Server des FB4](#), Login per FH-Account). Eine kurze Einführung in Git finden Sie [hier](#).
- Falls Sie Git nicht verwenden, so legen Sie Ihren Quellcode pro Praktikumsaufgabe in separaten Verzeichnissen (z.B. "praktikum2", "praktikum3") ab, damit die einzelnen Entwicklungsschritte bei den Abnahmen ersichtlich sind.

## Aufgabe 1: Bilder

---

Ergänzen Sie die Web-Anwendung folgendermaßen um Bilder:

1. Erweitern Sie die Seite mit der *Liste der Kategorien* ( `list.html` ), so dass zu jeder verfügbaren Kategorie ein Bild (Logo) angezeigt wird. Sowohl das Bild als auch der Name jeder Kategorie sollen dabei ein Hyperlink sein, welcher auf die Seite *Tutorials zu einer Kategorie* ( `tutorials.html` ) verlinkt.
2. Erweitern Sie die Seite *Tutorials zu einer Kategorie* ( `tutorials.html` ), so dass zu jedem enthaltenen Tutorial ein Bild (Logo) angezeigt wird. Setzen Sie semantische HTML-Elemente ein, um das Bild als *Abbildung* auszuzeichnen.

3. Erweitern Sie den Kopfbereich *auf allen Seiten*, so dass über dem Titel der Web-Anwendung zusätzlich ein Logo für die Web-Anwendung angezeigt wird.

Sämtliche Bilder können Sie frei wählen.

## Aufgabe 2: Erweiterung der Detailseite zu einem Tutorial

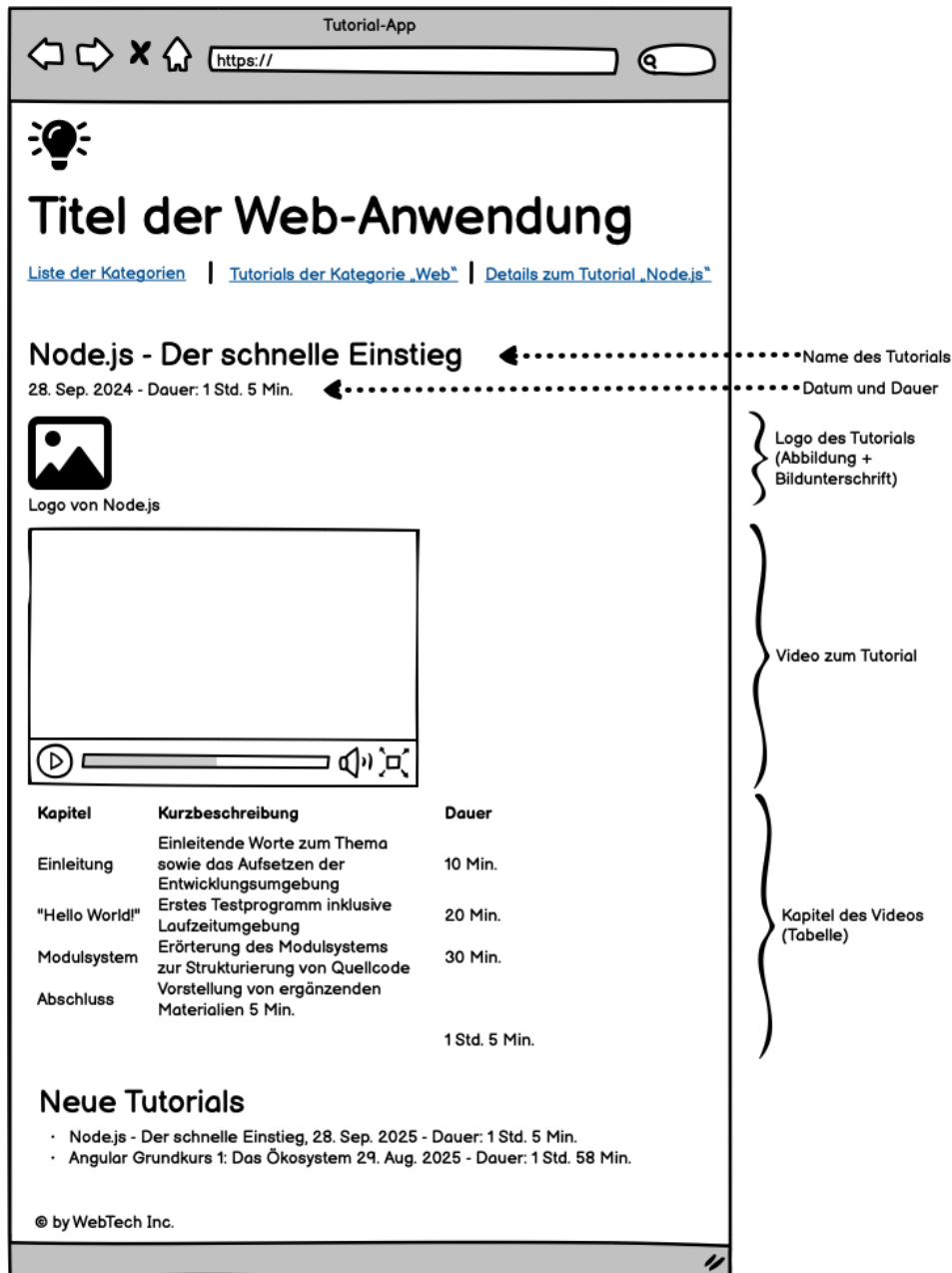
---

Erweitern Sie die *Detailseite zu einem Tutorial* ( `tutorial.html` ) so, dass diese ein Beispiel für ein *Video-Tutorial* ist. Ergänzen Sie dazu folgende Elemente:

1. Zusätzlich zu den bisherigen Informationen (Titel, Datum) soll die Seite nun auch ein Bild (Logo) für das Tutorial anzeigen. Setzen Sie semantische HTML-Elemente ein, um das Bild als *Abbildung mit Bildunterschrift* auszuzeichnen. Das Bild können Sie frei wählen.
2. In Praktikum 2 haben Sie auf der Seite einen Hyperlink zum Tutorial eingebaut (Aufgabe 1, Punkt 3). Ersetzen Sie diesen Hyperlink nun durch ein *Video* zum Tutorial. Betten Sie das Video so in die Seite ein, dass es direkt auf der Seite abgespielt werden kann. Arbeiten Sie hier zunächst mit einer lokal vorliegenden Video-Datei (diese können Sie frei wählen). Nutzen Sie lediglich HTML zur Lösung der Aufgabe.
3. Ergänzen Sie die Seite um eine *Tabelle*, welche die Kapitel des Videos anzeigt. Die Tabelle soll dabei folgende Spalten enthalten:
  - Name des Kapitels (z.B. "Einleitung"),
  - Kurzbeschreibung des Kapitelinhalts (z.B. "Einleitende Worte zum Thema sowie zum Aufsetzen der Entwicklungsumgebung"),
  - Dauer des Kapitels in Minuten (z.B. "10 Min.").

Zudem soll die Gesamtdauer des Tutorials in der Fußzeile der Tabelle angezeigt werden (z.B. "2 Std. 35 Min."). Verwenden Sie *geeignete semantische HTML-Elemente* zur Strukturierung der Tabelle.

Die Seite soll nach der Erweiterung ungefähr so aussehen wie in folgender schematischen Darstellung:



## Aufgabe 3: Suchformular

Fügen Sie auf der Seite mit der *Liste der Kategorien* ( `list.html` ) ein Formular zum Suchen nach Tutorials hinzu:

1. Das Formular ermöglicht es, den Titel eines Tutorials einzugeben.
2. Über eine Schaltfläche mit der Beschriftung "Suchen!" kann das Formular abgesendet werden.
3. Wählen Sie eine geeignete *HTTP-Methode* für das Formular.
4. Realisieren Sie das Formular mit folgendem Eingabefeld und Eigenschaften:

Feld	Name für Formularwert	Gültige Werte
Titel	search	<ul style="list-style-type: none"> <li>◦ Erforderliche Eingabe</li> <li>◦ Minimale Länge: 4 Zeichen</li> <li>◦ Maximale Länge: 20 Zeichen</li> <li>◦ Muss mit einem Großbuchstaben beginnen</li> </ul>

5. Zur Unterstützung der Benutzer\*innen soll das Eingabefeld mit einem hilfreichen *Platzhalter* versehen werden.

## Aufgabe 4: Hinzufügen eines neuen Tutorials

---

Wir ergänzen die Web-Anwendung nun um eine weitere Seite, auf welcher Benutzer\*innen ein neues Tutorial hinzufügen können.

Gehen Sie wie folgt vor:

1. Erstellen Sie eine neue HTML-Seite `form.html`, die ein *Formular* zum Anlegen eines neuen Tutorials enthält.
2. Realisieren Sie das Formular, so dass folgende Daten eines Tutorials mit entsprechenden gültigen Werten eingegeben werden können:

Feld	Name für Formularwert	Gültige Werte
Titel	name	<ul style="list-style-type: none"> <li>◦ Erforderliche Eingabe</li> <li>◦ Minimale Länge: 4 Zeichen</li> <li>◦ Maximale Länge: 20 Zeichen</li> <li>◦ Muss mit einem Großbuchstaben beginnen</li> </ul>
Sprache	language	<ul style="list-style-type: none"> <li>◦ Auswahl aus einer festen Menge von Optionen</li> <li>◦ Es kann nur eine Option ausgewählt werden (keine Mehrfachauswahl)</li> <li>◦ Mögliche Optionen: <input type="text" value="-"/> , <input type="text" value="Deutsch"/> , <input type="text" value="Englisch"/> , <input type="text" value="Spanisch"/> , <input type="text" value="Französisch"/></li> <li>◦ Folgende Werte sollen jeweils beim Absenden des Formulars übertragen werden ( Gewählte Option → Wert ): <input type="text" value="Deutsch"/> → <input type="text" value="de"/> , <input type="text" value="Englisch"/> → <input type="text" value="en"/> , <input type="text" value="Spanisch"/> → <input type="text" value="es"/> , <input type="text" value="Französisch"/> → <input type="text" value="fr"/></li> </ul>
Kategorie(n)	categories	<ul style="list-style-type: none"> <li>◦ Erforderliche Eingabe</li> <li>◦ Format siehe unten 3.1</li> </ul>
Beschreibung	description	<ul style="list-style-type: none"> <li>◦ Mehrzeilige Eingabe</li> <li>◦ Minimale Länge: 40 Zeichen</li> </ul>
Dauer	duration	<ul style="list-style-type: none"> <li>◦ Erforderliche Eingabe</li> <li>◦ Zeitangabe der Form <input type="text" value="MM:SS"/></li> <li>◦ Minimale Dauer: 20 Sekunden</li> <li>◦ Maximale Dauer: 15 Minuten 59 Sekunden</li> </ul>
Veröffentlichung (Datum)	date	Erforderliche Eingabe
Inhalt	type + content	<ul style="list-style-type: none"> <li>◦ Erforderliche Eingabe</li> <li>◦ Details siehe unten 3.2</li> </ul>

Bei Eingabe ungültiger Werte soll das Formular nicht abgesendet werden können. Nutzen Sie die Möglichkeiten der Formularvalidierung, die HTML Ihnen bietet.

3. Berücksichtigen Sie folgende weitere Eigenschaften des Formulars:

1. Sehen Sie für die Eingabe der Kategorie(n) ein Textfeld vor. Es sollen 1-n Kategorien als *kommaseparierte Liste* eingegeben werden können. Eine Kategorie ist dabei eine Folge von 1-n *beliebigen Zeichen*. Es muss mindestens eine Kategorie angegeben werden. Die Eingabe soll nicht mit einem Komma enden. Das Textfeld soll nur dementsprechend gültige Eingaben zulassen.

*Beispiel einer gültigen Eingabe:* `Kommunikation, Projektmanagement`

2. Die Eingabe des Formularfeldes "Inhalt" soll aus zwei Elementen bestehen:

1. Eine Auswahl, ob es sich um ein *Text-Tutorial* oder um ein *Video-Tutorial* handelt. Die entsprechende Auswahl soll beim Absenden des Formulars unter dem Namen "type" übertragen werden.
2. Ein Textfeld, in welchem je nach Auswahl in 1. entweder eine URL zum Text oder ein YouTube-Embed-Code für ein Video eingegeben werden kann. Der eingegebene Wert soll beim Absenden des Formulars unter dem Namen "content" übertragen werden.

**Hinweis:** Verwenden Sie hier zunächst unabhängig von der Auswahl in 1. ein einfaches Textfeld. In späteren Praktika werden wir das Formular so erweitern, dass hier je nach Auswahl dynamisch ein passendes Eingabefeld angezeigt wird.

3. Wählen Sie auch für die übrigen Daten jeweils geeignete Eingabeelemente.
4. Zur Unterstützung der Benutzer\*innen sollen alle Eingabefelder mit hilfreichen *Platzhaltern* versehen werden, wo möglich.
5. Verwenden Sie geeignete *semantische HTML-Elemente*, um die Eingabeelemente im Formular *logisch zu gruppieren*. Verwenden Sie das `label`-Element zur Beschriftung Ihrer Eingabeelemente.
4. Wählen Sie eine geeignete HTTP-Methode für das Formular.
5. Die neue HTML-Seite soll wie die anderen Seiten die *gemeinsamen Bereiche* aus Praktikum 2 besitzen.
6. Fügen Sie für die neu erstellte HTML-Seite einen Hyperlink im Navigationsmenü der Web-Anwendung hinzu (auf *allen* bisherigen Seiten!).

## Aufgabe 5: Formulare testen

---

Zum Testen der Formulare stehen Ihnen folgende zwei URLs zur Verfügung:

- Für das Suchformular: <https://labs.inf.fh-dortmund.de/tutorialService/testTutorialSearch.php>
- Für das Neuanlage-Formular: <https://labs.inf.fh-dortmund.de/tutorialService/testCreateTutorial.php>

Binden Sie Ihre Formulare an diese URLs an. Bei Aufruf dieser URLs über ein Formular werden jeweils die übergebenen Daten überprüft und die Ergebnisse der Überprüfungen dargestellt. Sorgen Sie dafür, dass die Überprüfungen keine Fehler melden.

## Allgemeine Hinweise:

---

- Validieren Sie Ihren HTML-Code mit dem [W3C Markup Validator](#). Der entstehende HTML-Code soll beim Check keine Fehler und Warnungen produzieren.
- Ausnahme: Warnungen bzgl. `date` - und `time` -Eingabefeldern können Sie ignorieren.

## Aufgabe 6: Verweis-Challenge (CTF)

---

Ihre Aufgabe ist es, eine HTML-Injection-Schwachstelle in der Kommentarfunktion einer Webanwendung zu finden und auszunutzen, um eine versteckte Flagge aufzudecken.

### Aufgabenstellung:

1. Nutze Sie Ihre HTML und HTTP-Header-Kenntnisse sowie die Entwicklertools Ihres Browsers, um die Schwachstelle zu identifizieren und die Flagge zu finden.
2. Wenn Sie die versteckte Flagge gefunden haben, tragen Sie diese auf der [CTFd-Plattform](#) im Feld 'Flag' ein.

**Link zur Challenge:** <https://ctfd.labs.inf.fh-dortmund.de/ctfd-challenge-6>

### Ziel:

Finden Sie versteckte Flagge und tragen Sie diese auf der [CTFd-Plattform](#) ein.

Viel Erfolg bei der Challenge!

### Hinweise zur Challenge:

Hier ein Auszug mit Informationen zu HTML-Injection vom [Open Worldwide Application Security Project \(OWASP\)](#):

„HTML-Injection ist eine Art von Injection-Schwachstelle, die auftritt, wenn ein Benutzer in der Lage ist, einen Eingabepunkt zu kontrollieren und beliebigen HTML-Code in eine verwundbare Webseite einzuschleusen. Diese Schwachstelle kann viele Folgen haben, wie

z. B. die Offenlegung der Sitzungs-Cookies eines Benutzers, die dazu verwendet werden könnten, sich als das Opfer auszugeben, oder, allgemeiner gesagt, kann sie es dem Angreifer ermöglichen, den Seiteninhalt, den die Opfer sehen, zu verändern.

Diese Sicherheitslücke tritt auf, wenn Benutzereingaben nicht ordnungsgemäß bereinigt und die Ausgabe nicht verschlüsselt wird. Eine Injection ermöglicht es dem Angreifer, eine schädliche HTML-Seite an ein Opfer zu senden. Der Zielbrowser ist nicht in der Lage, legitime Teile von schädlichen Teilen der Seite zu unterscheiden (ihnen zu vertrauen), und führt daher die gesamte Seite im Kontext des Opfers aus.“