

Miniprojekt

Einführung in die Informatik für Geisteswissenschaftler
Sommer 2024

Abgabe bis zum 09.07.2024, 23:59 Uhr

Miniprojekt Webshop

40 Punkte

Erstellt eine Folge von dynamischen HTML-Seiten, mit denen ein fiktiver Web-Shop realisiert wird. Was Ihr in eurem Shop anbietet – Bücher, Reisen, Hotelbuchungen oder was auch immer –, bleibt dabei Euch überlassen. Verwendet dazu die im Rahmen der Vorlesung und der Übungen vorgestellten Konzepte.

Nehmt eine strikte Trennung zwischen Form und Inhalt vor. Verwendet HTML 5, um die Struktur Eurer Seiten zu beschreiben, und gestaltet alle Seiten optisch ansprechend mit Hilfe von CSS. Sämtliche Layout-Angaben sollen in einer separaten CSS-Datei vorgenommen werden, die in alle Eure Seiten eingebunden wird, um ein einheitliches Aussehen sicherzustellen.

Die Einstiegsseite in den Web-Shop kann als statische HTML-Seite angelegt werden. Auf dieser Seite sollte der Besucher neben einem Bild als Aufmacher einen kurzen Text vorfinden, der erläutert, was ihn auf den folgenden Seiten erwartet. Mit Hilfe eines Formulars soll der Besucher auf dieser ersten Seite eine Kategorie aus einer Liste auswählen können. Dies kann beispielsweise bei Büchern ein Themengebiet, bei Reisen oder einer Hotelbuchung das Zielland sein. Es soll auch die Möglichkeit bestehen, dass der Besucher sich alle im Shop angebotenen Produkte anzeigen lässt.

Durch Abschicken des Formulars wird die zweite Seite mit der Produktübersicht geladen. Auf ihr sollen einige Informationen zu den einzelnen Produkten der ausgewählten Kategorie angezeigt werden. Hat der Besucher keine Kategorie ausgewählt, sollen sämtliche Produkte angezeigt werden, zu denen Daten gespeichert sind. Diese Seite soll zu wesentlichen Teilen dynamisch mit JavaScript erzeugt werden (zur Verwaltung der Daten der einzelnen Produkte s.u.). Zu jedem Produkt sollen der Name und der Preis sowie, falls für das jeweilige Produkt vorhanden, eine Kurzbeschreibung und ein Foto angezeigt werden. Der Benutzer soll hier eines der Produkte auswählen können. Es reicht aus, wenn der Benutzer immer nur ein Produkt kaufen kann.

Auf der dritten, ebenfalls dynamisch zu erzeugenden Seite mit Produktdetails werden ausführliche Informationen zu dem ausgewählten Produkt angezeigt (die mit * markierten Informationen sind optional und sollen nur dann angezeigt werden, wenn sie für das jeweilige Produkt gespeichert sind):

- Produktname
- Kategorie
- Eindeutige Produktnummer
- Preis
- Kurzbeschreibung (*)
- Ausführliche Beschreibung (*)
- Foto (*)
- Verweis auf WWW-Seite des Anbieters (*)

Auf dieser Seite soll der Benutzer die Möglichkeit haben, seine Kontaktdaten (Name und vollständige Adresse) anzugeben. Er soll die Bestellung dann abschließen können.

Miniprojekt

Einführung in die Informatik für Geisteswissenschaftler Sommer 2022

Zum Abschluss wird dem Benutzer eine Bestellbestätigung mit den von ihm angegebenen Kontaktdaten sowie Daten zum gekauften Produkt angezeigt.

Euer Webshop soll so weit wie möglich objektorientiert programmiert sein. Insbesondere sollten sowohl für die Produkte als auch für den Produktkatalog Klassen mit geeigneten Eigenschaften und Methoden definiert sein. Verseht sämtlichen JavaScript-Code (auch in den HTML-Seiten) mit sinnvollen Kommentaren; HTML-Code muss nicht mit Kommentaren versehen werden. Korrekte Abgaben, die nicht ausreichend kommentiert sind, können keine volle Punktzahl erreichen.

Als Abgabe erwarten wir eine Archiv-Datei im ZIP-Format mit folgenden Bestandteilen:

- die vier HTML-Seiten (Einstiegsseite, Produktübersicht, Produktdetails, Bestellbestätigung)
- alle in Eure Seiten eingebundenen Bilder,
- eine CSS-Datei, über die das Layout festgelegt wird,
- eine JavaScript-Datei, in der die Klasse für die Produkt-Objekte definiert wird,
- eine JavaScript-Datei, in der die Klasse für den Produktkatalog definiert wird,
- eine JavaScript-Datei, die den Produktkatalog befüllt.

In der letztgenannten Datei sollen mindestens neun Produkte in vier verschiedenen Kategorien beschrieben werden. Dabei soll mindestens ein Produkt ohne jegliche optionale Informationen sowie mindestens eines mit sämtlichen optionalen Informationen in den Katalog eingefügt werden.

Bewertungskriterien

Für alle HTML-Seiten werden wir jeweils bewerten, ob der HTML-Code korrekt und mit Strukturelementen ausgezeichnet ist und dass das Layout ausschließlich per CSS erfolgt. Außerdem bewerten wir bei der Einstiegsseite, ob das Formular die gemäß Aufgabenstellung geforderten Funktionen korrekt umsetzt.

Bei der Produktübersicht bewerten wir, ob alle Produkte der gewählten Kategorie angezeigt werden bzw. alle Produkte des Shops, falls der Besucher auf der ersten Seite keine Kategorie gewählt hat. Außerdem bewerten wir, dass alle geforderten Daten angezeigt werden und dabei berücksichtigt wird, dass optionale Daten nicht bei allen Produkten vorhanden sein müssen. Schließlich bewerten wir, dass auch hier das Formular die geforderten Funktionen korrekt umsetzt.

Bei der Seite mit den Produktdetails bewerten wir, ob das zuvor gewählte Produkt mit Hilfe der Produktnummer eindeutig bestimmt wird. Außerdem bewerten wir, ob alle geforderten Daten zu dem gewählten Produkt angezeigt und auch hier wieder optionale Daten berücksichtigt werden. Schließlich wird auch hier wieder bewertet, dass das Formular die geforderten Funktionen korrekt umsetzt.

Bei der Bestätigungsseite bewerten wir wieder, ob das gewählte Produkt mit Hilfe der Produktnummer eindeutig bestimmt wird. Außerdem bewerten wir, ob die geforderten Produkt- und Kundendaten angezeigt werden.

Miniprojekt

Einführung in die Informatik für Geisteswissenschaftler Sommer 2022

Wir bewerten zusätzlich, ob die Startseite erkennbar ist und die CSS-Datei korrekt ist. Die JavaScript-Dateien bewerten wir dahingehend, dass weitgehend objektorientiert programmiert wurde, insbesondere dass in separaten Dateien jeweils eine Klasse für die Produktobjekte sowie den Produktkatalog erstellt wurde sowie dass die Produkte als Objekte dem Katalog durch eine geeignete Methode hinzugefügt und durch geeignete Methoden in die Seiten ausgegeben werden. Schließlich bewerten wir, dass der JavaScript-Code in diesen Dateien verständlich dokumentiert ist.

Vorgeschlagenes Vorgehen

Das Vorgehen bei der Programmierung kann selbstverständlich frei gewählt werden. Wir empfehlen jedoch, das Miniprojekt der folgenden Beschreibung entsprechend zu bearbeiten. Die ersten Schritte können bereits mit den Inhalten der ersten Vorlesungen bearbeitet werden, da dafür lediglich HTML und CSS benötigt werden. Auch eine Planung, was in dem Webshop verkauft werden soll, sollte frühzeitig erfolgen.

Im Folgenden wird eine Folge von Schritten beschrieben, wie Ihr dieses Miniprojekt bearbeiten könnt. Diese Schritte sind so aufgebaut, dass jeweils Teilergebnisse der Lösung entstehen, die meist im Browser betrachtet und auch getestet werden können. Das hier vorgestellte Vorgehen ist wie gesagt nur ein Vorschlag, von dem Ihr abweichen könnt; Ihr könnt auch anders an die Aufgabe herangehen.

Schritt 1: Vorbereitungen

Erstellt ein Verzeichnis für Euer Projekt und erzeugt darin ein Dokument mit einem elementaren HTML-Rahmen. Achtet darauf, dass dieses Dokument korrektes HTML 5 mit allen notwendigen Angaben, insbesondere einen Dokumententyp und eine Zeichenkodierung, enthält. Mindestens die Tags `html`, `head`, `title`, `body`, `header` und `footer` sollten enthalten sein. Es bietet sich an, hier auch bereits ein Stylesheet einzubinden.

Sofern Ihr auf allen Euren Seiten einen einheitlichen Seitenkopf und -abschluss verwenden wollt, solltet Ihr hier auch den Header und den Footer geeignet füllen.

Speichert diese Datei unter dem Namen `vorlage.html` in Eurem Verzeichnis.

Schritt 2: Erstellung der Einstiegsseite

Erstellt eine Kopie der Datei `vorlage.html` und baut diese so zur Einstiegsseite des Web-Shops aus, dass sie alle in der Aufgabenstellung geforderten Elemente enthält. Gebt der Datei einen geeigneten Namen wie beispielsweise `index.html`, so dass erkennbar ist, dass es sich bei dieser Datei um die Einstiegsseite handelt.

Schritt 3: Erstellung eines ersten Stylesheets

Erstellt eine erste Version Eures Stylesheets und testet diese in Verbindung mit der in Schritt 2 erstellten Einstiegsseite. Gestaltet die Seite so, dass sie ansprechend aussieht.

Miniprojekt

Einführung in die Informatik für Geisteswissenschaftler
Sommer 2022

Schritt 4: Vorbereitung der Produktdaten

Erstellt auf Papier, in einer Textdatei oder in einer Tabellenkalkulation eine Tabelle der von Eurem Web-Shop angebotenen Produkte mit allen Daten, die zu diesen gespeichert werden sollen. Die ersten Spalten sollten die Pflichtangaben (Name des Produkts, Kategorie, eindeutige Produktnummer, Preis) zu allen Produkten enthalten; die weiteren Spalten sollten die optionalen Angaben enthalten, so dass hier einzelne Tabelleneinträge leer sein werden. Achtet darauf, dass Ihr auf jeden Fall mindestens ein Produkt im Katalog habt, das über keine der optionalen Angaben verfügt.

Schritt 5: Erstellung der Klasse für die Produktobjekte

Die Produkte sollen als einzelne Objekte angelegt werden. Erstellt dazu eine JavaScript-Datei, die den gleichen Namen trägt wie die Klasse, also beispielsweise buch.js, wenn Ihr Bücher verkaufen wollt. In ihr definiert Ihr zunächst eine Klasse für ein Produkt-Objekt, deren Konstruktor sämtliche möglichen Angaben als Parameter übergeben bekommt und diese in geeigneten Eigenschaften speichert (also sowohl die Pflichtangaben als auch die optionalen Angaben). Die optionalen Angaben sollten dabei mit sinnvollen Default-Werten vorbelegt werden.

Ergänzt die Klasse um eine geeignet benannte Methode (beispielsweise kurzansicht()), die keine Parameter übergeben bekommt und lediglich die Pflichtangaben des Objekts als HTML-Zeichenkette per Rückgabewert zurückgibt. Diese Zeichenkette soll später in der Produktübersicht angezeigt werden. Testet die Funktion des Konstruktors und der weiteren Methode, indem Ihr eine Kopie der Datei vorlage.html erstellt. Im Kopf dieser Seite muss zunächst die soeben erstellte JavaScript-Datei eingebunden werden. Dann sollten in der Datei zwei oder mehr Objekte erzeugt und der Rückgabewert der Methode für diese Objekte per document.write() ausgegeben werden.

Schritt 6: Erstellung der Klasse für den Produktkatalog

Erstellt eine weitere JavaScript-Datei, in der Ihr eine Klasse für Euren Produkt-Katalog definiert. Am einfachsten ist es, ein Array zur Speicherung der Objekte zu verwenden. Der Konstruktor sollte lediglich einen leeren Katalog erzeugen, also eine Eigenschaft anlegen, die ein leeres Array enthält. Ergänzt diese Katalog-Klasse um eine geeignet benannte Methode, die dem Katalog ein als Parameter übergebenes Objekt hinzufügt.

Testet die Funktion auch dieses Konstruktors und der Methode, indem Ihr die in Schritt 5 erstellte Datei zunächst so modifiziert, dass die in diesem Schritt neu erstellte JavaScript-Datei zusätzlich eingebunden wird. Dann soll die Datei weiter modifiziert werden, sodass ein Katalog-Objekt erzeugt wird, dem dann die erzeugten Produkt-Objekte hinzugefügt werden.

Die Ausgabe der Produkt-Objekte sollte dann durch eine Schleife erfolgen, die über den Katalog läuft. Durchläuft dazu das in dem Katalog-Objekt enthaltene Array mit den Produkten mit Hilfe einer for-Schleife und ruft für jedes Produkt die in Schritt 5 erstellte Methode kurzansicht() auf. Deren Rückgabewert übergibt Ihr der Methode document.write (), um ihn in das Dokument auszugeben.

Diese Test-Datei wird im weiteren Verlauf nicht mehr benötigt und kann nun gelöscht werden.

Miniprojekt

Einführung in die Informatik für Geisteswissenschaftler
Sommer 2022

Schritt 7: Erstellung des JavaScript-Codes für die Befüllung des Produktkatalogs

Erstellt eine weitere JavaScript-Datei, die eine geeignet benannte Funktion zur Befüllung des Produktkatalogs enthält. Diese Funktion sollte keine Parameter erhalten. In ihr sollte zunächst ein Katalog-Objekt als lokale Variable erzeugt werden. Anschließend sollte für jedes Eurer Produkte in einer weiteren lokalen Variablen ein Objekt mit den Pflichtangaben erzeugt, ggf. um die optionalen Angaben ergänzt und dem Katalog hinzugefügt werden. Die Funktion sollte dann abschließend das Katalog-Objekt als Rückgabewert liefern.

Schritt 8: Erstellung einer ersten Version der Produktübersicht

Erstellt eine Kopie der Datei `vorlage.html`. Schreibt in diese geeigneten HTML- und JavaScript-Code, so dass zunächst eine Liste aller Eurer Produkte mit den für jedes Produkt vorhandenen Daten (also ohne die optionalen Daten) ausgegeben wird. Dazu müssen zunächst die JavaScript-Dateien im Kopf Eurer Seite eingebunden werden. Im Rumpf der Seite sollte zunächst die Funktion zum Aufbau des Produktkatalogs aufgerufen werden. Der Rückgabewert dieser Funktion – das Objekt mit dem Produktkatalog – sollte in einer Variablen gespeichert werden.

Durchläuft das Array mit den Produkten mit einer `for`-Schleife und ruft für jedes Produkt die in Schritt 5 erstellte Methode `kurzansicht()` auf. Deren Rückgabewert übergibt Ihr der Methode `document.write()`, um sie in das Dokument auszugeben.

Schritt 9: Berücksichtigung der Produktkategorie in der Produktübersicht

Modifiziert die in Schritt 8 erstellte Produktübersicht so, dass nur die Angaben zu den Produkten ausgegeben werden, die zu einer bestimmten Kategorie gehören. Deklariert dazu eine Variable und weist ihr zunächst einen festen Wert zu, mit dem die Kategorie angegeben werden kann. Als Werte sollen hier die von Euch verwendeten Kategorien sowie die Zeichenkette „alle“ möglich sein. Je nach Wert der Variablen sollen dann die Produkte der entsprechenden Kategorie bzw. alle Produkte ausgegeben werden. Testet diese Datei mit verschiedenen Werten für die Variable, in der die Kategorie gespeichert ist.

Schritt 10: Kopplung der Produktübersicht an das Formular der Einstiegsseite

Modifiziert die in Schritt 2 erstellte Einstiegsseite so, dass sie ein Formular enthält, in dem der Besucher der Seite eine der Kategorien oder sämtliche Kategorien auswählen kann. Dieses Formular sollte die in Schritt 9 erstellte Produktübersicht laden. Das Formular sollte außerdem einen `on-submit`-Event-Handler erhalten, der eine im Kopf der Seite definierte Funktion aufruft. Diese Funktion sollte aus dem Formular ermitteln, welche Kategorie der Benutzer ausgewählt hat, und diesen Wert im Local Storage speichern.

Damit die vom Benutzer vorgenommene Eingabe in der Produktübersicht ausgewertet werden kann, muss diese ebenfalls modifiziert werden. Statt der Variablen für die Kategorie dort einen festen Wert zuzuweisen, muss nun der entsprechende Wert aus dem Local Storage ermittelt werden.

Miniprojekt

Einführung in die Informatik für Geisteswissenschaftler
Sommer 2022

Schritt 11: Berücksichtigung optionaler Eigenschaften in der Kurzansicht

Ergänzt die Methode für die Kurzansicht Eurer Produkte um geeignete Abfragen, um auch die in der Aufgabenstellung für die Kurzansicht geforderten optionalen Eigenschaften zu berücksichtigen, also die Kurzbeschreibung und das Foto immer dann (und nur dann) mit in der erzeugten Zeichenkette zurückzugeben, wenn sie für das jeweilige Produkt vorhanden sind.

Schritt 12: Erstellung einer ersten Version der Seite mit den Produktdetails

Erstellt eine weitere Kopie der Datei `vorlage.html`. Modifiziert diese so, dass sämtliche zu einem Produkt gespeicherten Daten ausgegeben werden. Erweitert dazu die Klasse der Produkte (s. Schritt 5) um eine weitere Methode `detailansicht()`, die ähnlich vorgeht wie die Methode `kurzansicht()`, und wie in der Aufgabenstellung gefordert alle Pflichtdaten sowie die optionalen Daten berücksichtigt.

Auf der Seite mit den Produktdetails kann die Produktnummer als eindeutiges Unterscheidungsmerkmal der Produkte verwendet werden. Deklariert für diese Seite zunächst wiederum eine Variable und weist ihr einen festen Wert zu, mit dem die Produktnummer angegeben werden kann. Testet diese Datei mit verschiedenen Werten für die Variable, in der die Produktnummer gespeichert ist.

Schritt 13: Kopplung der Seite mit den Produktdetails an die Produktübersicht

Modifiziert die Produktübersicht so, dass die dort ausgegebene Liste der Produkte in ein Formular geschrieben wird. Versendet der Benutzer dieses Formular, soll die in Schritt 12 erstellte Seite mit den Produktdetails geladen werden. Die Seite mit den Produktdetails ist dahingehend zu modifizieren, dass die Variable für die Produktnummer aus dem Local Storage ermittelt wird, statt sie mit einem festen Wert zu belegen.

Um jedes einzelne Produkt auswählbar zu machen, ist in der Produktübersicht außerdem die in Schritt 5 erstellte Methode für die Kurzansicht zu ergänzen. Zu jedem Produkt sollte zusätzlich ein Radio-Button mit in die HTML-Zeichenkette geschrieben werden, der als Wert die Produktnummer enthält. Das Formular sollte ähnlich wie die Einstiegsseite mit einem Event-Handler versehen werden, der eine im Kopf der Seite definierte Funktion aufruft. Diese Funktion sollte den vom Benutzer gewählten Radio-Button ermitteln und dessen Wert im Local Storage ablegen.

Schritt 14: Ergänzung der Seite mit den Produktdetails um Kundendaten

Modifiziert die in Schritt 12 erstellte Datei zur Anzeige der Produktdetails so, dass dort zusätzlich ein Formular angezeigt wird, in das der Benutzer seine Kontaktdaten eintragen kann. Dieses Formular sollte die noch zu erstellende Bestellbestätigung laden. Das Formular sollte außerdem ähnlich wie die Einstiegsseite einen `onSubmit`-Event-Handler erhalten, der eine im Kopf der Seite definierte Funktion aufruft. Diese Funktion sollte alle für die Bestellbestätigung benötigten Daten aus dem Formular ermitteln und im Local Storage speichern.

Miniprojekt

Einführung in die Informatik für Geisteswissenschaftler
Sommer 2022

Schritt 15: Erstellung der Bestellbestätigung

Erstellt eine weitere Kopie der Datei `vorlage.html`. Dort soll nun die Bestellbestätigung angezeigt werden, die die Kurzansicht des Produkts sowie die auf der vorangegangenen Seite vom Benutzer eingegebenen Kontaktdaten enthält. Die benötigten Daten sind aus dem Local Storage auszu-lesen. Für die Ausgabe der Beschreibung des Produkts sollte eine neue Methode `bestaetigungsansicht()` erstellt werden, die praktisch identisch zur Methode `kurzansicht()` ist, jedoch keine Radio-Buttons enthält.

Schritt 16: Test, Layout, Optimierung, Dokumentation

Testet Euren Web-Shop ausführlich. Überarbeitet gegebenenfalls die einzelnen HTML-Dateien auch in optischer Hinsicht und modifiziert Euer Stylesheet so, dass alle Seiten ein einheitliches Erscheinungsbild besitzen. Überprüft auch, ob der JavaScript-Code genügend gut kommentiert ist.