

Prüfungsformalitäten

Umfang des Portfolios

Die Portfolioprüfung beinhaltet die beiden Vorlesungen „Webprogrammierung“ und „Verteilte Systeme“ aus dem dritten und vierten Semester. Das Prüfungsportfolio setzt sich deshalb aus folgenden Bestandteilen zusammen, die wie folgt verrechnet werden:

Artefakt	Gewichtung
3. Semester: Webprogrammierung	
Lernkontrollen in Moodle	2
Dieses Aufgabenblatt	4
4. Semester: Verteilte Systeme	
Lernkontrollen in Moodle	1
Neues Aufgabenblatt	3

Bearbeitung in Eigenleistung

Grundsätzlich sind alle Artefakte in Eigenleistung zu erstellen. Die Artefakte müssen den persönlichen Wissensstand widerspiegeln und soweit wie möglich einen individuellen Lösungsweg erkennen lassen. Zwar ist es durchaus erlaubt, sich in Form von Rückfragen oder Internetrecherche Hilfe zu holen. Die abgegebenen Lösungen müssen aber dennoch selbst ausgearbeitet worden sein, um gewertet werden zu können. Fremde Quellen müssen dabei entsprechend wissenschaftlicher Standards kenntlich gemacht werden.

Export der Lernkontrollen in Moodle

Da alle Prüfungsartefakte archiviert werden müssen, erhalten alle Studierenden nach Abschluss des Semesters ein PDF-Dokument mit den exportierten Lernkontrollen aus Moodle. Dieses Dokument muss dem Portfolio vor Abgabe hinzugefügt werden. Wir bitten daher, mit der Abgabe des Portfolios zu warten, bis das PDF-Dokument verschickt wurde.

Abgabe

Die Abgabe erfolgt in Moodle bis zum Ende des Semesters bis zum Tag vor Beginn der nächsten Praxisphase. Abweichend davon kann eventuell über Moodle ein anderer Stichtag, der in allen Kursen einheitlich gehandhabt wird, kommuniziert werden.

Aufgabenstellung

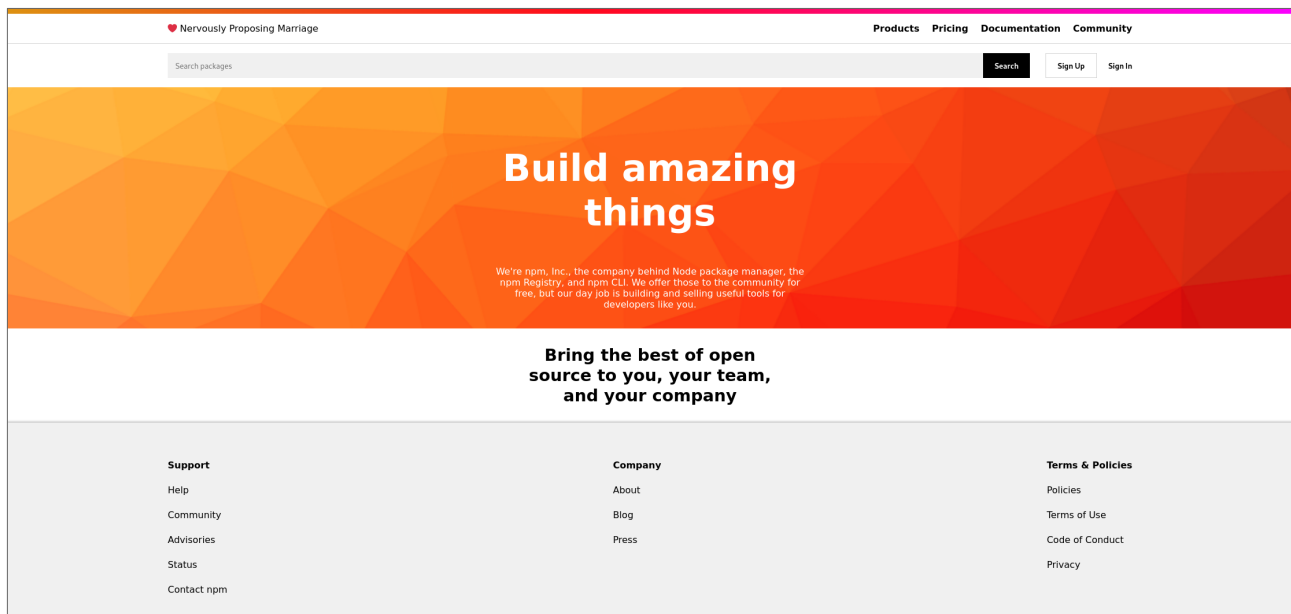
Aufgabe 1: Grundlagen des Web

(10 + 10 + 5 Punkte)

- a) Setzen Sie eine gültige HTTP-Anfrage an einen beliebigen Webserver ab, ohne einen Webbrowser hierfür zu benutzen. Verwenden Sie stattdessen ein Testwerkzeug wie Postman oder httpie, mit dem die Inhalte der HTTP-Anfrage exakt definiert und die gesamte Antwortnachricht angezeigt werden kann. Dokumentieren Sie das Vorgehen sowie die HTTP-Nachrichten mit entsprechenden Screenshots.
- b) Erklären Sie anhand der eben erstellten Screenshots die Funktionsweise von HTTP sowie die typischen Bestandteile einer HTTP-Nachricht. Gehen Sie dabei insbesondere auf den Aufbau der Nachrichten ein.
- c) Rufen Sie einen beliebigen Wikipedia-Artikel auf und klicken Sie im Inhaltsverzeichnis auf eine der Überschriften. Verwenden Sie dann die in der Adresszeile des Browsers sichtbare URL, um daran die wesentlichen Bestandteile einer URL zu erklären.

Aufgabe 2: HTML und CSS

(10 + 10 + 5 Punkte)



- a) Während der Vorlesung haben Sie die Übungsaufgabe „npmjs.com“ bearbeitet und die Startseite von npmjs.com nachgebaut. Kommentieren Sie nun den Quellcode der Musterlösung. Mit welchen Überlegungen wurde der HTML-Code so aufgebaut, wie er ist und welche Aufgabe erfüllen die einzelnen HTML-Elemente? Erklären Sie auch, wie sich die jeweiligen Anweisungen im Stylesheet auswirken.
- b) Erstellen Sie ein paar Screenshots einer aus Ihrer Sicht besonders gut gestalteten Webseite oder Webanwendung und erklären Sie daran typische Gestaltungsregeln für Webseiten und Anwendungen. Achten Sie dabei insbesondere auf Dinge wie die Anordnung der Elemente, Farbgebung, Schriftsatz, Größen und Abstände oder das allgemeine Erscheinungsbild (Flat Design vs. Skeuomorphismus) ein.
- c) Erklären Sie anhand entsprechender Beispiele zwei gängige UI-Patterns (im Sinne der Interaktion zwischen Benutzer*in und Webanwendung anstatt nur dem Erscheinungsbild) im Webdesign.

Aufgabe 3: Responsive Webdesign

(10 + 10 Punkte)

- a) Sie wurden beauftragt, die Webseite eines kleinen Hotels in Ihrem Ort neu zu gestalten. Insbesondere die Präsentation der unterschiedlichen Zimmerkategorien soll durch eine für alle Bildschirmarten optimierte Gestaltung aufgewertet werden. Entwerfen Sie verschiedene Mockups für kleine, mittlere, große Bildschirme zur Vorstellung der Zimmer und zeigen Sie anhand der Mockups, wie Kopfbereich, Menü, Beschreibung, Bilder, Formular für Buchungsanfragen, ... auf unterschiedlichen Endgeräten angeordnet werden.
- b) Setzen Sie das eben erarbeitete Layout als Prototyp um. Erstellen Sie hierzu eine statische HTML-Datei mit Dummy-Inhalten (z.B. Lorem ipsum Blindtext), die das fertige Erscheinungsbild auf den unterschiedlichen Gerätetypen demonstriert. Die Seite soll Mobile First entwickelt werden und auf allen Geräten ein ansprechendes Layout besitzen.

Aufgabe 4: JavaScript

(10 + 10 + 10 Punkte)

- a) Erklären Sie anhand praktischer Quellcodebeispiele die Gemeinsamkeiten und Unterschiede zwischen Java und JavaScript. Gehen Sie dabei vor allem die Definition und Verwendung von Funktionen bzw. Methoden sowie die Typisierung von Variablen, Listen und Objekten ein.
- b) Im Onlineskript befindet sich eine Stoppuhr als Beispiel für den Zugriff auf das Document Object Model mit JavaScript. Die gesamte Logik befindet in der Event-Listener-Funktion für das `load`-Ereignis. Schreiben Sie den Quellcode so um, dass die Logik in einer neuen Klasse gekapselt wird. Bei Auftreten des `load`-Ereignisses soll dann nur noch eine Objektinstanz der Klasse erzeugt und eine Methode namens `start()` aufgerufen werden, um die Anwendung zu starten. Die bisher lokalen Variablen sollen zu Objektattributen, die im Konstruktor mit Werten versorgt werden, und die Funktion `updateDisplay()` zu einer Methode umgewandelt werden.
- c) Entwickeln Sie eine einfache Single Page App auf Basis des in der Vorlesung vorgestellten SPA-Routers und der darauf aufbauenden Beispielarchitektur. Denken Sie sich hierfür einen einfachen Anwendungsfall mit zwei unterschiedlichen Bildschirmseiten aus, die nicht der Vorlage aus der Vorlesung entsprechen. Die Daten der Anwendung müssen nicht dauerhaft gespeichert werden. Wie immer sollte die Anwendung aber ein liebevoll gestaltetes, responsives Layout besitzen. HTML-Code und Stylesheet der einzelnen Seiten sollen darüber hinaus in getrennten Dateien definiert und zur Laufzeit dynamisch nachgeladen werden.