

Übungsblatt 2: Die Hypertext Markup Language

Vorstellung in den Tutorien am 8./9. November 2017

Hinweis zur Programmierung: Benutzen Sie zum Programmieren die Entwicklungsumgebung Cloud9. Verwenden Sie **noch kein** CSS oder JavaScript in den Aufgaben, wo es nicht explizit verlangt wird, damit bei den Vorstellungen im Tutorium die Übersichtlichkeit behalten wird. Zum Lösen der Aufgaben reicht pures HTML – es ist nicht erforderlich, dass die Webseiten gut aussehen.

Hinweise zur Präsentation Für die Durchführung der Übungen ist Ihre aktive Teilnahme erforderlich und erwünscht. Zu diesem Zweck sind die Aufgaben derart strukturiert, dass Sie die Lösungen in einem Kurzvortrag in der Übung präsentieren können. Planen Sie bei der Vorbereitung Zeit für Fragen und Diskussionen ein. Benutzen Sie für die Darstellung Ihrer Ergebnisse ihren eigenen Laptop oder den Computer des Tutors und sparen Sie nicht mit begleitenden Erläuterungen auf der Tonspur. Damit Sie ihre Lösungen immer digital zur Verfügung haben, sichern Sie diese unbedingt vor jeder Übung in der OwnCloud der TUB, wo jeder TUB-Student automatisch einen Account besitzt: <https://tubcloud.tu-berlin.de/>

Hinweise zur Klassifikation Generell können Sie sich frei aussuchen, welche Aufgaben Sie bearbeiten wollen um Punkte zu sammeln. Damit Sie unterscheiden können, welche Aufgaben wichtig für diese Veranstaltung sind und welche ergänzend sind, wird eine Klassifikation eingeführt. Die Aufgaben werden anhand der Hintergrundfarben der Überschriften unterschieden:

- **Wichtig:** die Themen dieser Aufgaben sind **vollständig** klausurrelevant. Manche Aufgaben sind thematisch ähnlich, deshalb müssen nicht immer alle wichtigen Aufgaben vollständig bearbeitet werden.
- **Unterstützend:** die Themen dieser Aufgaben sind **prinzipiell** klausurrelevant, jedoch dringt die Aufgabe tiefer in das Thema ein als für die Abschlussprüfung benötigt. Diese Aufgaben liefern ergänzende Informationen für ein grundlegendes Verständnis, was es erleichtert die Thematik vollständig zu verstehen.
- **Zusatz:** die Themen dieser Aufgaben sind **nicht** klausurrelevant. Sie gehen auf andere Aspekte des Themengebiets ein, die in der Veranstaltung (meist aus Zeitgründen) nicht behandelt werden. Sie sind rein informativ.

2.1 Statistiken über Webseiten (2 Punkte)

Auf Folie 10 von Kapitel 2 sehen Sie, dass Webseiten eine Vielzahl von Referenzen auf andere Dokumente/Ressourcen enthalten. Diese erweitern eine Webseite um bestimmte Aspekte, z.B. liefern CSS-Dateien zusätzliche Informationen über das Design einer Webseite.

Seit einigen Jahren verfolgt und analysiert das Projekt *HTTP Archive*¹ Websites im Internet. Fertigen Sie eine **Präsentation** an, in der Sie auf die wichtigsten Diagramme der allgemeinen Trends der TOP 1000 Websites seit November 2010 eingehen. Präsentieren und erklären Sie außerdem die aktuellen Statistiken der TOP 1000 Websites. Werten Sie danach noch eine eigene Website ihrer Wahl aus.

Lesen Sie außerdem unter dem *About*-Reiter in den FAQs nach, wo die Begrenzungen der Analyse von *HTTP Archive* liegen?

2.2 Die erste Webseite (4 Punkte)

In dieser Übung sollen Sie eine (erste) kleine Webseite erstellen. Lesen Sie sich dafür die Kurzeinführung zu HTML von Mozilla² durch und nutzen Sie ergänzend Kapitel 2 bis Folie 30.

Erstellen Sie in Cloud9 einen neuen Workspace mit einem **Blank**-Template, der in Ihrem eigenen Dashboard liegt. Dazu muss bei der Einstellung *Team* die voreingestellte Option *webtech* auf *Don't set a team for this workspace* gewählt werden³:

¹<http://httparchive.org>

²https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics

³Alle zu Testzwecken bereits angelegten Workspaces im Team-Arbeitsbereich bitte wieder löschen.

Create a new workspace

Workspace name

my-first-html-website

Description

Make a short description of your workspace

Team

Don't set a team for this workspace

Hosted workspace

Clone workspace

Remote SSH workspace

Salesforce

☐ Private

This is a workspace for your eyes only

☒ Public

This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

e.g. ajaxorg/ace or git@github.com:ajaxorg/ace.git

Choose a template

HTML5

Node.js

PHP, Apache & ...

Python

Django

Ruby

C++

Wordpress


Rails Tutorial

Blank


Harvard's CS50

Create workspace

Konfigurieren Sie nun zuerst die lokale Versionierung des Projekts mit Git, indem Sie `git init` im Terminal ausführen, was mit einer erfolgreichen Initialisierung eines leeren Repositories bestätigt wird. Erstellen Sie danach eine neue HTML-Datei namens `FirstHTMLPage.html` im Workspace. In die Datei schreiben Sie das HTML-Grundgerüst mit `<!DOCTYPE html>`, `<html>`, `<head>`, `<meta charset="utf-8">` und `<body>`-Tags rein.



3



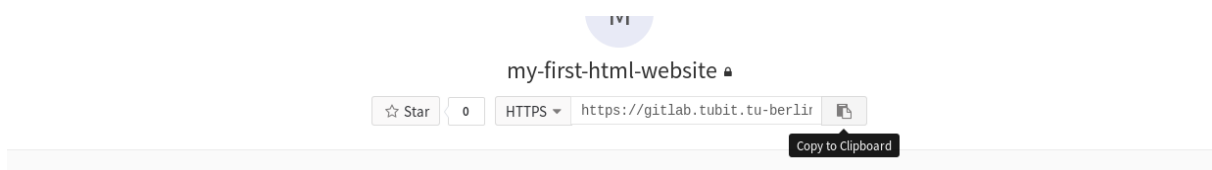
Die in dieser Aufgabe zu erstellende Webseite ist dabei Ihnen überlassen. Legen Sie allerdings nur eine reine HTML-Seite und noch **keine** CSS-/JavaScript-Dateien an. (Bei Unklarheiten über die möglichen HTML-Elemente und dessen Attribute sind Informationen in der MDN HTML Element Reference zu finden.) Die minimalen Anforderungen an die Webseite sind :

- das `<title>`-Element verwenden
- einen Text mit mehreren Abschnitten und einer Überschrift einbringen
- ein verkleinertes Bild einbinden, welches in einem Unterordner namens `images` gespeichert ist
- eine ungeordnete und geordnete Liste einfügen
- einen Link sowohl auf eine externe (nicht von Ihnen erstellte), als auch auf eine interne (von Ihnen neu erstellte) Webseite einbinden
- das Bild soll auf sich selbst verlinken, so dass beim Klick auf das Bild nur noch das Bild selbst in Original-Größe angezeigt wird

Sie sollten jetzt mindestens zwei HTML-Dateien und eine Bild-Datei in Ihrem Workspace haben. Mit Klicks auf *Preview – Live Preview File* können Sie die Ergebnisse ihrer Programmierung live verfolgen und so die Fehler korrigieren. (Falls das nicht funktioniert, muss die Webseite mit *Run* und durch Klick auf die im Terminal erscheinende *URL – Open* geöffnet werden ⁴ .)

Wenn alle Webseiten erstellt und korrekt sind, sollte der Zustand in Git versioniert werden. Schreiben Sie sich dazu die folgenden Schritte/Befehle auf und erklären Sie diese im Tutorium: Fügen Sie alle neuen/geänderten Dateien der Staging Area (auch *Index* genannt) hinzu und erstellen Sie eine neue Version. Vergeben Sie anschließend noch einen sinnvollen *Tag* für diese Version ⁵.

Zum Teilen und eventuell späteren Zusammenarbeiten mit anderen Kommilitonen soll der Sourcecode noch auf einen externen Git-Server der TU Berlin abgelegt werden. Dazu erstellen Sie unter <https://gitlab.tubit.tu-berlin.de> ein neues Projekt. Wichtig hierbei ist die Repository-URL, die Sie sich kopieren sollten:



Mit dem Befehl `git remote add origin URL` können Sie dann das externe Repository unter dem Namen *origin* hinzufügen. Ob das erfolgreich war, können Sie mit `git remote -v` überprüfen. Interessant ist an dieser Stelle, dass Sie ihr Git-Projekt theoretisch nicht nur mit einem Repository (standardmäßig *origin* genannt) synchronisieren können, sondern mit beliebig vielen.

Laden Sie nun im letzten Schritt noch alle Daten und Tags in das Repository hoch.

⁴Um die Anzeige im Preview-Fenster zu ermöglichen, müssen im Chrome eventuell noch die Cookies von `*.c9.io` erlaubt werden, z.B. durch das in der URL-Leiste befindliche Cookie-Symbol.

⁵Für Unklarheiten beim Arbeiten mit Git lesen Sie das Basic-Kapitel der Dokumentation

Zum besseren Verständnis: Speichern Sie abschließend das Projekt auf Ihrer lokalen Festplatte, indem Sie es aus Cloud9 herunterladen und entpacken. Wie Sie gesehen haben, sind Webseiten im Grunde nur Textdateien mit Anweisungen zur Gliederung und Darstellung von Inhalten (sogenannte Markup-Sprachen). Diese Dateien werden von einem Programm, dem Browser oder auch User Agent genannt, interpretiert und entsprechend dargestellt. Um ihre eigene Webseite anzuzeigen, reicht es daher die HTML-Datei im Browser zu öffnen.

Damit andere Personen Ihre Webseite betrachten können, müssen Sie diese öffentlich anbieten. Dazu muss der Computer als Server fungieren. Entfernte Computer die eine Webseite von einem Server abrufen, nennt man auch Clients. Damit ein Server eine Webseite aber überhaupt anbieten kann, muss er mit den Clients kommunizieren und sie verstehen – das erfolgt über das standardisierte Protokoll *HTTP*, welches Sie später in der Veranstaltung noch kennenlernen werden.

2.3 JavaScript (2 Punkte)

In dieser Aufgabe sollen Sie kurz den Stoff der Vorlesung wiederholen, indem Sie Kapitel 2 des frei verfügbaren Buches *Eloquent JavaScript*⁶ lesen. Zum Testen können Sie die JavaScript-Konsole von Chrome benutzen, die Sie in der letzten Übung kennengelernt haben. Da Sie nun aber teilweise mehrzeiligen Code schreiben müssen, ist es meist komfortabler stattdessen eine neue JavaScript-Datei, z.B. `ha21.js`, in Cloud9 anzulegen und diese mit *Run* (oder im Terminal mit `node ha21.js`) zu starten.

Führen Sie alle Aufgaben am Ende des Kapitels durch. Im Tutorium erklären Sie ihren Kommilitonen kurz die wichtigsten Punkte des Kapitels und wie Sie die Aufgaben gelöst haben, wozu unterstützend eine Präsentation oder Live-Demo genutzt werden kann.

⁶http://eloquentjavascript.net/02_program_structure.html