

Web-Entwicklung, VU

jQuery

Agenda

- Fragen
- Ass 2 beurteilt
- Freiwillige Übung
 - 2 Lösungswege
- Ausgabe Ass 3
- data-* attributes
- Events
- JavaScript - Bootstrap 4

Organisatorisches

9: XSLT/Projekt: XML to HTML, 24.5: Fällt aus, aber wer mag, es gibt eine freiwillige Fragestunde im kleinen Seminarraum.

weil parallel:

Fachspezifische digitale Methoden (Die Anordnung der Dinge. Informationsvisualisierung in den Kultur- und Geisteswissenschaften) [Windhager F](#)

10: XSLT/XML to HTML (Chrisi) 7.6

11: XSLT/Projekt (Chrisi/Chris) 14.6

Abgabe: Assignment 3

Anwesenheit freigestellt weil LV Windhager

12: XSLT + Fragestunde (Chrisi/Chris) 21.6

13: Präsentation der Projekte (Chrisi/Chris) 28.6

Abgabe: Projekt

Freiwillige Übung: Aufgabenstellung

- 3 Kategorien/Reiter, die per Klick ein `` mit 3 `` anzeigen.
- anderen `` bleiben verborgen und nur das ausgewählte `` angezeigt.
- Geklickte Reiter werden immer hervorgehoben (class = „active“)
- Jedes `` kann ausgewählt werden mit einem Kästchen.
- Ausgewählt `` werden im WebStorage gespeichert.
- Show Data gibt alle ausgewählten Kategorien in der Konsole aus.
- Clear löscht alles im WebStorage.

<https://moodle.uni-graz.at/mod/assign/view.php?id=111240>

Freiwillige Übung

ex_jQuery.js

https://moodle.uni-graz.at/pluginfile.php/359883/mod_resource/content/1/ex_jQuery.js

ex_JavaScript.js

https://moodle.uni-graz.at/pluginfile.php/359888/mod_resource/content/1/ex_javascript.js

ex_Christopher.js

https://moodle.uni-graz.at/pluginfile.php/359891/mod_resource/content/1/ex_Christopher.js

```
function hideDataUls() {  
    var dataUls = document.querySelectorAll(".data");  
    var i = 0;  
    while (i < dataUls.length) {  
        dataUls[i].classList.add("hidden");  
        i++;  
    }  
}  
  
function showOnlyCorrespondingData(e) {  
    var dataUls = document.querySelectorAll(".data");  
    var menuLi = document.querySelectorAll(".menu li");  
    var compareText = e.target.textContent;  
    console.log("The corresponding class is: " + compareText);  
    var i = 0;  
    while (i < dataUls.length) {  
        if (dataUls[i].classList.contains(compareText)) {  
            dataUls[i].classList.remove("hidden");  
        }  
        i++;  
    }  
}
```

```
$('#button.kat').click(function () {  
    $(this).addClass("active");  
    $(this).siblings("button").removeClass("active");  
    $('ul.' + this.id).removeClass("d-none");  
    $('ul.' + this.id).siblings('ul').addClass('d-none');  
});
```

```
function supplyMenuWithEventListener() {  
  console.log("--- Calling function  
supplyMenuWithEventListener --- \n");
```

```
  var menuLi = document.querySelectorAll(".menu li");  
  var i = 0;  
  while (i < menuLi.length) {  
    menuLi[i].addEventListener('click', function(Event) {  
      hideDataUls();  
      showOnlyCorrespondingData(Event);  
    });  
    i++;  
  }  
}
```

```
$('#button#store').click(function () {}
```

Assignment 3

Abgabe 14.06

Fragen zu deinem Code beim Abgabegespräch; Plagiatskontrolle;

Projektspezifische Aufgabe (5 Punkte)

Implementierte eine einfache Funktionalität für dein Projekt mit JavaScript/jQuery.
(Autocomplete, Highlight, einfachen Viewer einbauen, ...)

<https://jqueryui.com> / <https://getbootstrap.com/docs/4.0/getting-started/javascript> / <https://d3js.org>

Annotations Tool (15 Punkte)

Fragestunde bzw. Hilfestellungen:

24.5 Teil der LV

30.5, Mi bzw. Termin nach Bedarf

per Mail was vereinbaren, einfach vorbei kommen

Ass 3: Einfaches Annotations Tool [15 Pkt.]

- Datensätze in deinem Projekt sollen mittels Checkbox oder Button auswählbar sein. Bei Klick auf eine Checkbox/Button taucht ein Dialog-Fenster auf. Das Dialog-Fenster ist ein Formular mit den Eingabefeldern Titel, Beschreibung und Datum. Weiters gibt es einen 'Abbrechen' und einen 'Hinzufügen' Button. **[3 Pkt.]**
- 'Hinzufügen' ist nur aktiv, wenn Titel mindestens 1 Zeichen enthält.
'Abbrechen' schließt das Dialog-Fenster. Titel und Beschreibung können beliebige Strings enthalten, Datum wird mittels eingebauten <https://jqueryui.com/datepicker> ausgewählt. **[3 Pkt.]**
- Titel, Datum und die Beschreibung werden zusammen mit der ID des gewählten Objektes im localStorage des Browsers gespeichert, wenn man auf 'Hinzufügen' klickt. Danach schließt sich das Dialog-Fenster. **[3 Pkt.]**
- In einem weiteren HTML Dokument (annotation.html), das über ein <a> erreichbar ist, werden alle so gespeicherten Annotationen als Liste dargestellt. Beim Aufrufen der Seite wird immer der aktuelle Bestand im localStorage als Liste angezeigt. **[3 Pkt.]**
- Es gibt zwei Buttons zum Sortieren der Liste nach Titel und nach Datum. **[2 Pkt.]**
- Ein Button der alle Einträge löscht. **[1 Pkt.]**
- [Optional: Man kann einzelne Einträge löschen [1 Pkt.] | Man kann einzelne Einträge edieren [3 Pkt.]]

Spezifikationen

Klick auf Kategorie macht Inhalte sichtbar

| Annotationstool | |
|--|-------------------------------------|
| Superhelden | Clear |
| Show Annotationen | |
| Batman | <input checked="" type="checkbox"/> |
| Spiderman | <input checked="" type="checkbox"/> |
| Superman | <input type="checkbox"/> |
| Titel | |
| Annotation zu Spiderman | |
| Beschreibung | |
| Wurde ja von einer radioaktiven Spinne gebissen. | |
| Datum | |
| 2018-05-14 | |
| <input type="button" value="Hinzufügen"/> | |
| <input type="button" value="Abbrechen"/> | |

Erst wenn in Titel etwas steht, wird 'Hinzufügen' Button sichtbar.

Datum nur der Form
YYYY-MM-DD
Erlaubt

Spezifikationen

Annotations-Asnicht in annotation.html

Titel: Beschreibung, Datum, URL mit # zum Datensatz

Annotations Tool

Sortieren Titel Sortieren Datum

- A Batman: Test test test , 2015-16-02, file:///home/pollin/Downloads/ex/index.html#Batman
- Spiderman-Titel: Lorem Ipsum dolor..., 1991-01-01, file:///home/pollin/Downloads/ex/index.html#Spiderman
- Superheld: Lorem Ipsum dolor, 1990-01-01, file:///home/pollin/Downloads/ex/index.html#Spiderman
- Test: Test test test , 2018-16-06, file:///home/pollin/Downloads/ex/index.html#Superman

Hints

Dialog box/popup window in Bootstrap

<https://jqueryui.com/dialog>

https://www.w3schools.com/bootstrap4/bootstrap_modal.asp

localStorage/webStorage

https://www.w3schools.com/html/html5_webstorage.asp

data()

The data() method attaches data to, or gets data from, selected elements.

<https://api.jquery.com/jquery.data> | https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_misc_data_set_get

sort()

https://www.w3schools.com/jsref/jsref_sort.asp

change()

https://www.w3schools.com/jquery/event_change.asp

Hints

- Daten auswählen
- *this* zeigt auf *@id* des aktuellen Elements
- url wird aus der url erzeugt, mit Pfad + #
- id muss man zwischenspeichern -> *jQuery.data()*
- Dialogfenster erzeugen
- datepicker
- *addData()*
- check title on *change()*
- localStorage
- localStorage sortieren
 - *sort(function(a, b)*

data-* attribute

Wird verwendet um zusätzliche, “private” Daten auf einer Webseite zu speichern, die in HTML-Elemente als benutzerdefinierte Attribute, eingebunden werden.

data-* attribute können einfach mit JavaScript verarbeitet werden.

Bestehen aus 2 Teilen:

- data-
- und einem beliebigen String-Wert

https://www.w3schools.com/tags/att_global_data.asp

Beispiel: <http://gams.uni-graz.at/o:ufbas.1563>

Event

| Event | Description |
|-------------|--|
| onchange | An HTML element has been changed |
| onclick | The user clicks an HTML element |
| onmouseover | The user moves the mouse over an HTML element |
| onmouseout | The user moves the mouse away from an HTML element |
| onkeydown | The user pushes a keyboard key |
| onload | The browser has finished loading the page |

https://www.w3schools.com/jsref/event_onmouseover.asp

Bootstrap - JavaScript

Bootstrap 4 Collapse

https://www.w3schools.com/bootstrap4/bootstrap_collapse.asp

Bootstrap 4 JS Collapse

https://www.w3schools.com/bootstrap4/bootstrap_ref_js_collapse.asp

Web-Entwicklung, VU **Einführung**

Wer sind wir?

Christian Steiner

Translation (Eng/BKS), EuroMACHS

Cantus Network

<https://gams.uni-graz.at/cantus>

Jahrrechnungen der Stadt Basel

<https://gams.uni-graz.at/srbas>

Cooking Recipes of the Middle Ages



Christopher Pollin

Geschichte, EuroMACHS

Stefan Zweig digital

<https://glossa.uni-graz.at/context:szd>

Urfehdebücher Stadt Basel

<https://gams.uni-graz.at/ufbas>

MEDEA

<https://glossa.uni-graz.at/context:medea>

Wer seit ihr?

Wer hat eine Webseit gemacht?

Mit WYSIWYG-Editor?

Oder selbst gecodet?

Hat schon jemand CSS Erfahrung?

Hat jemand schon JavaScript Erfahrung?

THE TWO STATES OF EVERY PROGRAMMER



**I GOT
A JOB**

memegenerator.net



"WHY IS IT WHEN MY DESIGN LEAVES ME IT LOOKS LIKE JENNIFER ANISTON BUT IT LOOKS LIKE ROSEANNE BARR WHEN DEVELOPMENT GETS DONE WITH IT?"

**OH, SO YOU THINK I'M
CODING THE WRONG WAY**



**A user interface is like a joke.
If you have to explain it,
it's not that good.**

Digital Synopses.com

TYPES GOOGLE

Administratives

Anwesenheit

3 x Fehlen - kein Entschuldigungsmail notwendig
ab 4. (begründetem) Fehlen → Kontaktaufnahme, Ersatzleistung;

Aufwand

4 Ects ~ 4 mal 25 h = 100 h

Kontakt

Wenn es für alle interessant sein könnte: Moodle-Forum (Code, Hilfe)
sonst vorbeikommen,

Oder Mail: christopher.pollin@uni-graz.at
christian.steiner@uni-graz.at

Administratives

Mitarbeit ist erwünscht, Fragen stellen, LV mitgestalten :)

Stoffumfang

Folien + Ressourcen in Moodle (Links, Tutorials, Videos etc.)

Themen in LV behandelt, aber selbständiges Studium

Negative Note

nicht ausreichend Punkte, alle Assignments müssen <50% sein.

ab 1. Leistung muss benotet werden

Nulltoleranz-Politik für Plagiate

Moodle <https://moodle.uni-graz.at/course/view.php?id=12550>

Editor

Oxygen (Lizenzen bekommt ihr von uns), Notepad++, ... (VSCode, Atom, Sublime Text)

Zeitplan

1: Einführung (Chrisi/Chris) 8.3

2: HTML (Chris) 15.3

Ausgabe: Assignment 1

3: Cascading Stylesheets - CSS (Chris) 22.3

4: CSS-Frameworks I (Chris) 12.4

Abgabe: Assignment 1

5: CSS-Frameworks II (Chris) 19.4

Ausgabe: Assignment 2

6: JavaScript (Chrisi) 26.4

7: Das DOM und jQuery I (Chrisi) 3.5

Abgabe Assignment 2

8: jQuery II (Chrisi) 17.5

Ausgabe: Assignment 3

9: XSLT/Projekt : Ajax und JSON (Chrisi) 24.5

10: XSLT/Projekt : XML to HTML (Chrisi/Chris) 7.6

Abgabe: Assignment 3

11: XSLT/Projekt (Chrisi/Chris) 14.6

12: Werkstatt aka Fragestunde (Chrisi/Chris) 21.6

13: Präsentation der Projekte (Chrisi/Chris) 28.6

Abgabe: Projekt

Abgabegespräch: Woche 02 - 06.Juli

Benotung

Abgabegespräch = Theorie

Übungsteil = Assignment, Projekt

Notenschlüssel

20 Pkt **Assignment 1**

Moodle-Abgabe
HTML, Struktur der Webseite

105 - 110 Pkt = 1

94 - 104 Pkt = 2

20 Pkt **Assignment 2**

Moodle-Abgabe
CSS, Bootstrap:
Design und Darstellung der Webseite

77 - 93 Pkt = 3

56 - 76 Pkt = 4

0 - 55 Pkt = 5

20 Pkt **Assignment 3**

Moodle-Abgabe
Javascript, jQuery, Bootstrap-Javascript:
Funktionalität der Webseite

Alle Teilleistungen müssen
positiv (< 50%) sein!

30 Pkt **Projekt**

Aufbauend auf Assignments

20 Pkt **Abgabegespräch**

Projekt vorstellen, erklären, Feedback und
theoretischer Stoff (30 - 45 min)

Abgabe

Auf Moodle zu den eingetragenen Deadlines.

Harte Deadline: Keine Abgabe oder verspätete = 0 Punkte

Abgabe als 1 ZIP-File; andere Abgabeformen → :*(

Kurz-Präsentation (5-10 min) des Projektes am 28.06 verpflichtend.

- Was waren deine Daten?

- Was hast du wie dargestellt?

- Welche Funktion war dafür notwendig?

Assignment - Projekt - Abgabegespräch

Assignment 1

GeWi - Daten bzw. Thema für die zu erstellende Webseite finden.
HTML Struktur modellieren.

Assignment 2

Die HTML-Struktur aus Ass1 mit CSS und Bootstrap responsive stylen

Assignment 3

Passende Funktionalität für die Daten mit JavaScript etc. implementieren.

Projekt

Die Daten mit XSLT zu einer fertigen Webseite bringen

Abgabegespräch

Plagiatskontrolle, Fragen zum Code, Theorie, Feedback

Literatur und Lernunterlagen

Tutorials:

<https://www.w3schools.com/html/default.asp>

<https://www.w3schools.com/css/default.asp>

<https://www.w3schools.com/js/default.asp>

<https://www.w3schools.com/bootstrap/default.asp>

<https://wiki.selfhtml.org/wiki/HTML/Tutorials/HTML-Einstieg>

Online-Testumgebung:

<https://codepen.io>

<https://jsfiddle.net>

<https://jsbin.com/?html,output>

Forum:

<https://stackoverflow.com>

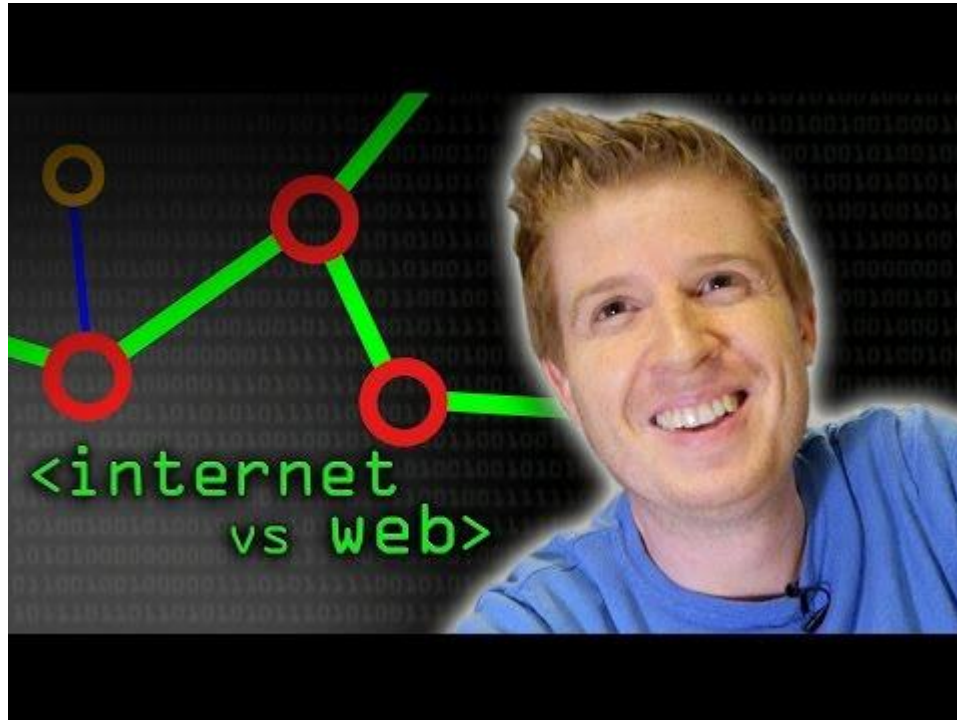
Videos (Channels/Playlists):

[SGML HTML XML \(Computerphile\)](#)

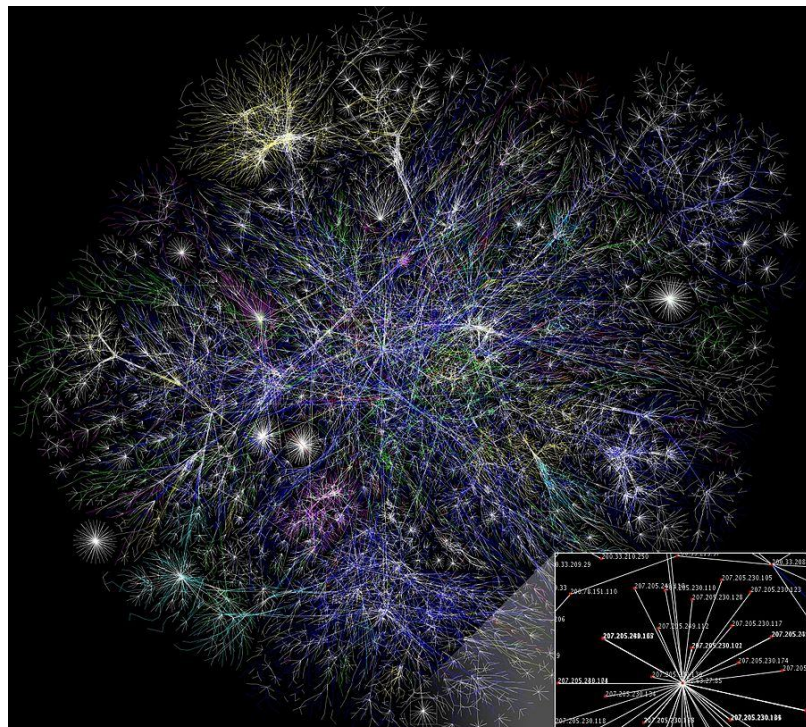
[CSS HTML Web Programming Basics \(ColorCode\)](#)

[Grundlagen der Webentwicklung \(auf Deutsch\)](#)

World Wide Web (WWW) vs. Internet

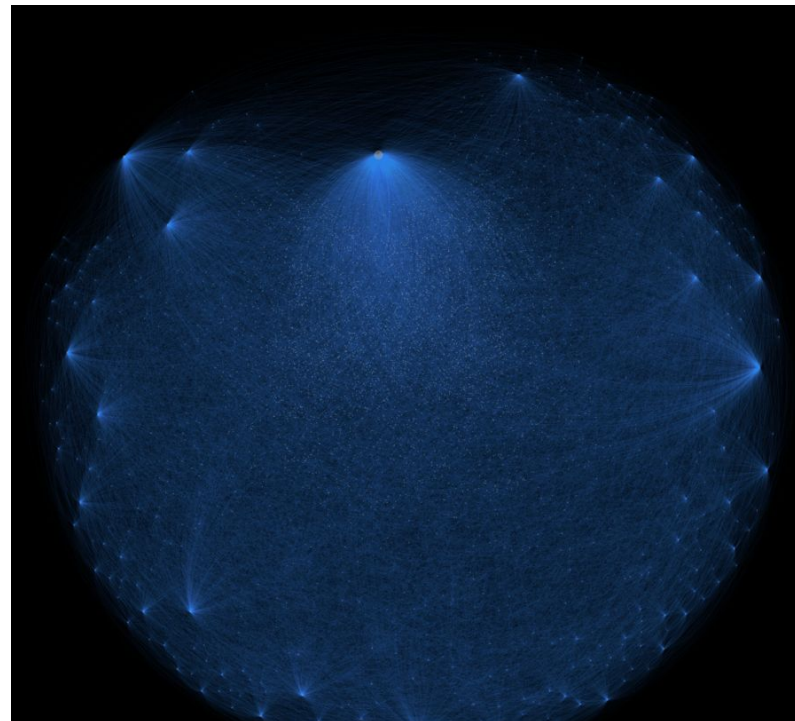


Internet



Visualisierung eines Teils des Internets

Web



Visualisierung des world wide web common crawls aus dem Jahr 2012
(44 mio. domains)

Internet: Die Infrastruktur und das Netz der Rechnernetze

Internet - die technische Infrastruktur!

Gewaltiges **Netzwerk** von millionen von **unabhängigen** Computern, das **dezentral** angelegt ist.

Jeder Rechner ist eindeutig adressierbar - IP

<https://whatismyipaddress.com>

143.50.30.45 = <http://gams.uni-graz.at>

Jeder Computer kann mit jedem anderen mittels

Protokollen (HTTP, FTP, etc.) kommunizieren.

Offene Systemarchitektur, jeder kann es erweitern, aber es gibt W3C

Endgeräte = Host ●

Server überprüft/antwortet -

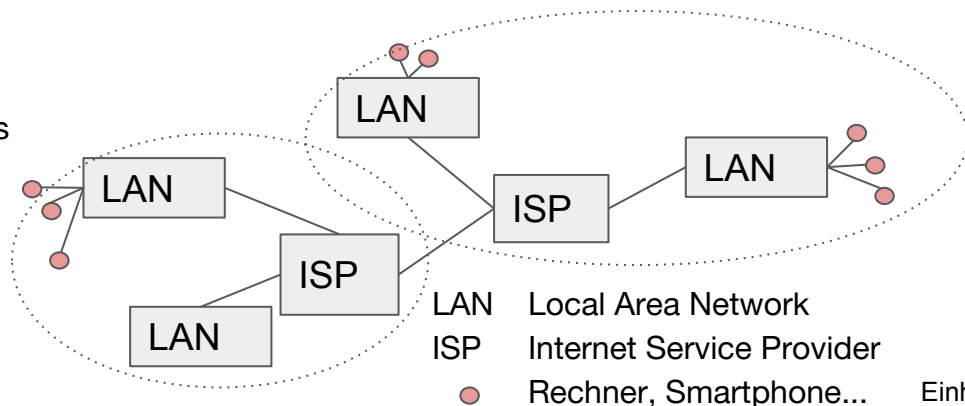
Client fordert an/ kontaktiert

Ist ein weltweiter Zusammenschluss weltweiter Netze zu einem **virtuellem Netz**.

Internetworking

bezeichnet die Verbindung von zwei oder mehreren Computernetzen

Kommunikationsprotokolle regeln Datenaustausch über physikalische Netzwerke hinweg.



Internet Protokoll Stack - OSI-Referenzmodell

| OSI-Schicht | TCP/IP-Schicht | Beispiel |
|--------------------|----------------|--|
| Anwendungen (7) | Anwendungen | HTTP, UDS, FTP, SMTP, POP, Telnet, OPC UA |
| Darstellung (6) | | |
| Sitzung (5) | | |
| | | SOCKS |
| Transport (4) | Transport | TCP, UDP, SCTP |
| Vermittlung (3) | Internet | IP (IPv4, IPv6), ICMP (über IP) |
| Sicherung (2) | Netzzugang | Ethernet, Token Bus, Token Ring, FDDI, IPoAC |
| Bitübertragung (1) | | |

World Wide Web (WWW): Ein Internetdienst

Ein gigantisches Hypermediasystem mit weltweit verteilten Ressourcen

Ab 1989 von Tim Berners-Lee am CERN entwickelt, Gesamtkonzept 1991 **zur freien Verfügung!** veröffentlicht. (kein Patent, keine Lizenzen)

Erste Webseite aller Zeiten am 6. August 1991 veröffentlicht: <http://info.cern.ch/hypertext/WWW/TheProject.htm>

Hypermedia-Dokumente sind mit Hyperlinks verbunden und bilden ein Netzwerk.

Anwendungen - wie das WWW - können über das Internet genutzt werden dank der Internetprotokolle auf der Anwendungsschicht des TCP/IP-Protokollstapels.

Ein Weg um auf Information - in Form von Hypermedia Dokumenten (Hyperlinks → Netzwerk) - zuzugreifen.

Dokumentenstruktur (HTML) vs. Dokumentendarstellung (CSS)

Web-Technologien

| | | |
|---------------------------|-----------------------------|--|
| URL | Uniform Resource Locator | - “Ort” eines Dokuments? |
| HTML | Hypertext Markup Language | - Struktur der Dokumente |
| CSS | Cascading Style Sheets | - Darstellung, Layout der Dokumente |
| HTTP | Hypertext Transfer Protocol | - Übertragung der Dokumente |
| Web-Programmierung | | <ul style="list-style-type: none">- Frontend: was der User sieht- Backend: im Hintergrund (Datenbanken) |

URI - URL - URN

Zur weltweit eindeutigen Identifikation und Adressierung von Dokumenten im Web dienen...

URI - Uniform Resource Identifier

URN - Uniform Resource Name

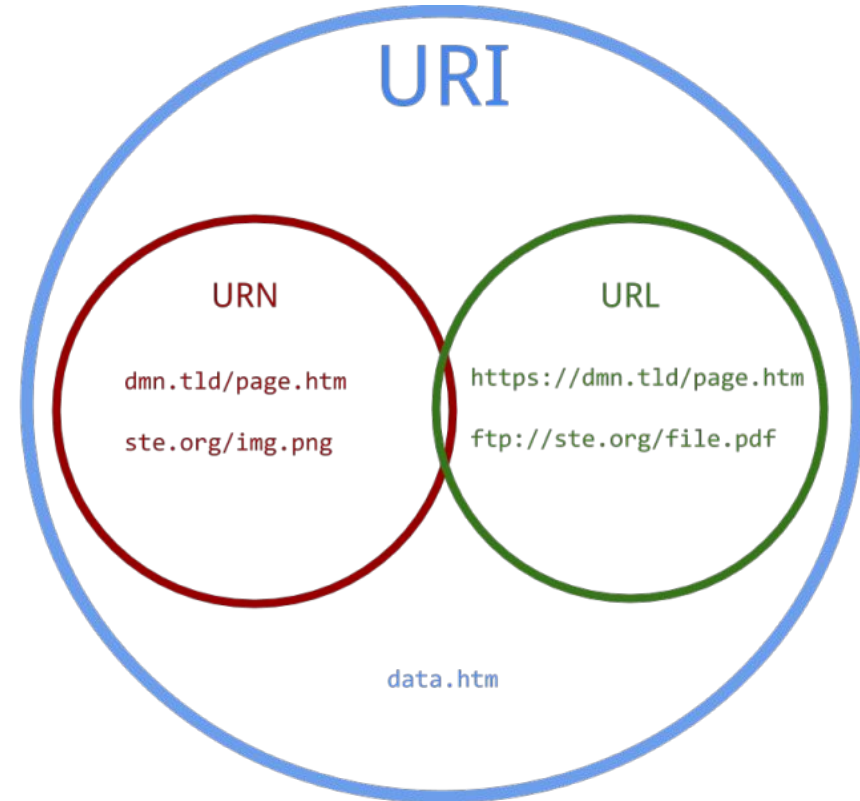
URL - Uniform Resource Locator

Verbreitetste Ausprägung der URIs sind URLs

<https://gams.uni-graz.at/context:ufbas>

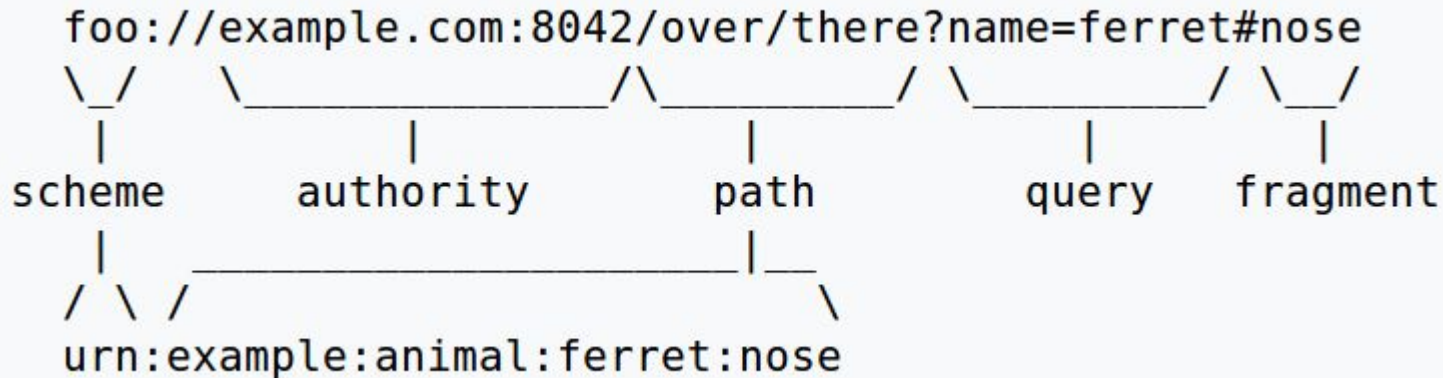
Bestandteile einer URL

- Name des Zugriffsprotokolls: https, ftp, smb ...
- Adresse des Servers: gams.uni-graz.at
- Name des Dokuments: index.html oder in der GAMS ein Datenstrom: context:ufbas



Aufbau einer URI

URI besteht aus fünf Teilen wobei nur “scheme” und “path” verpflichtend sind:



[http://gams.uni-graz.at/archive/objects/query:ufbas.volltext/methods/sdef:Query/get?params=\\$1|wyn](http://gams.uni-graz.at/archive/objects/query:ufbas.volltext/methods/sdef:Query/get?params=$1|wyn)

<http://gams.uni-graz.at/o:ufbas.1563#Eintrag-2>

URI vs. URL vs. URN

URL: `ftp://ftp.is.co.za/rfc/rfc1808.txt`

URL: `http://www.ietf.org/rfc/rfc2396.txt`

URL: `ldap://[2001:db8::7]/c=GB?objectClass?one`

URL?: `mailto:John.Doe@example.com`

URL?: `news:comp.infosystems.www.servers.unix`

URL: `telnet://192.0.2.16:80/`

URN (keine URL): `urn:oasis:names:specification:docbook:dtd:xml:4.1.2`

URN (keine URL): `tel:+1-816-555-1212`

Strenge Aufteilung wurde aufgegeben, da z.B. das mailto scheme keiner der beiden Klassen zugeordnet werden kann

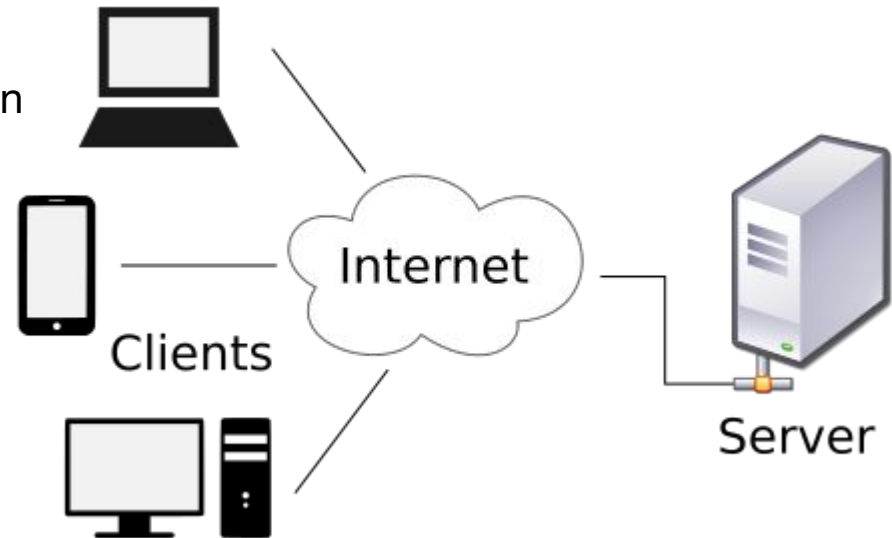
Client - Server Architektur

Benutzer fordert über den Browser (Client) ein Dokument im Web an

Browser kontaktiert den in der URL spezifizierten Server und fordert gewünschtes Dokument an

Server greift auf sein lokales Filesystem zu und sendet (in URL) spezifizierte Datei als Anfrage an den Browser

| | | | |
|---------|---|----------------------|-----------------|
| Browser | { | empfängt | Dokument |
| | | interpretiert | |
| | | stellt dar | |
| | | analysiert | |
| | | weist zu | HTML Struktur |
| | | | CSS-Anweisungen |



HTTP: Kommunikation zwischen Server und Client

Interaktion zwischen Browser und Server erfolgt
mittels **H**yper**T**ext **T**ransfer **P**rotocol

Protokoll auf Anwendungsschicht des
Internetprotokoll-Stack

Interaktion erfolgt lediglich in Form eines einfachen
Frage/Antwort-Verfahrens

Client macht einen **Request**

Server gibt **Response**

Protokoll ist zustandslos

Server 'weiß' nicht ob Client schon einmal
gefragt hat.

Es entstehen einfache Kommunikationszyklen.



Client Request

GET Anfordern von Server-Daten

POST Datenübertragung an Server

Weitere Methoden:

HEAD

PUT

DELETE

TRACE

OPTIONS

CONNECT

PATCH

Server Response

Trying 143.50.30.45
Connected to gams.uni-graz.at

Verbindungsaufbau
zum Server

GET / http/1.1
Host: gams.uni-graz.at

HTTP-Anfrage

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close
```

Serverantwort: Header

```
<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

Serverantwort: Body

Verbindungsende

HTML

HTML (Hypertext Markup Language) ist eine Auszeichnungssprache (so wie XML auch)

Mit HTML wird die Struktur des Dokumentes beschrieben (Überschriften, Absätze, Listen, Tabellen etc.)

Es können Hyperlinks und Multimedia-Komponenten eingebunden werden

Layout ist nur sehr begrenzt mit HTML beeinflussbar (→ CSS)

Die Kennzeichnungselemente werden Tags genannt.

Browser interpretieren Dokument auf Basis des HTML Markups

HTML

Alle HTML Dokumente müssen mit einer *document type declaration* beginnen: **<!DOCTYPE html>**

Das HTML Dokument selbst beginnt mit **<html>** und endet mit **</html>**

Der sichtbare Inhalt eines HTML Dokuments steht zwischen **<body>** und **</body>**, Metadaten in **<head>**

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Überschrift 1</h1>
    <p>Absatz</p>
  </body>
</html>
```

Valid ✓

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titel</title>
  </head>
  <body>
    <h1>Überschrift 1</h1>
    <p>Absatz</p>
  </body>
```

Empfohlen! 😊

```
<!DOCTYPE html>
<title>Titel</title>
<h1>Überschrift 1</h1>
<p>Absatz</p>
```

Valid ✓ (nicht empfohlen!)

CSS

Während HTML die Dokumentstruktur beschreibt, sorgt CSS (**C**ascading **S**tyle **S**heets) für die Beschreibung der Darstellung (Layout) dieser Struktur.

CSS ermöglicht:

Beschreibung für Bildschirm (PC, Tablet, Handy etc.) und Druck

Einheitliches Layout in allen zugehörigen HTML-Dokumenten

Entwurf eines Dokuments nach den Wünschen und Bedürfnissen des Autors oder Nutzers

Nutzung und Koexistenz von kompletten Stylesheet-Hierarchien, welche vom Autor, Browser-Hersteller oder Nutzer erstellt werden können

CSS - Syntax



Pures CSS:

<https://codepen.io/Beaugust/pen/JxLnK>

28 Webseiten zur Layout Inspiration:

<https://www.creativeblog.com/web-design/examples-css-912710>

Frontend and Backend



Web Programmierung

JavaScript

https://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst

Einzige standardisierte und weit verbreitete Technologie für clientseitige Skripte in Webseiten

JQuery

Freie JavaScript Library für Zugriff und Manipulation

DOM - Document Object Model

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_method

W3C Schnittstelle für Zugriff und Manipulation auf HTML-Dokument und seine Teilbereiche

DOM modelliert Dokument entsprechend seiner Struktur als Strukturbaum

z.B. kann man mit JavaScript zugreifen

AJAX - Asynchronous JavaScript And XML

https://www.w3schools.com/js/tryit.asp?filename=tryjs_ajax_first

Daten vom Webserver lesen nachdem Seite schon geladen wurde

Update ohne Seite erneut zu laden

Daten senden

Keine Programmiersprache

Umsetzung von

Animationen, 3D-Grafik, Games, mobile Apps und und und

Ein erstes JavaScript Beispiel

```
<!DOCTYPE html>
<html>
<body>

<h2>My First JavaScript</h2>

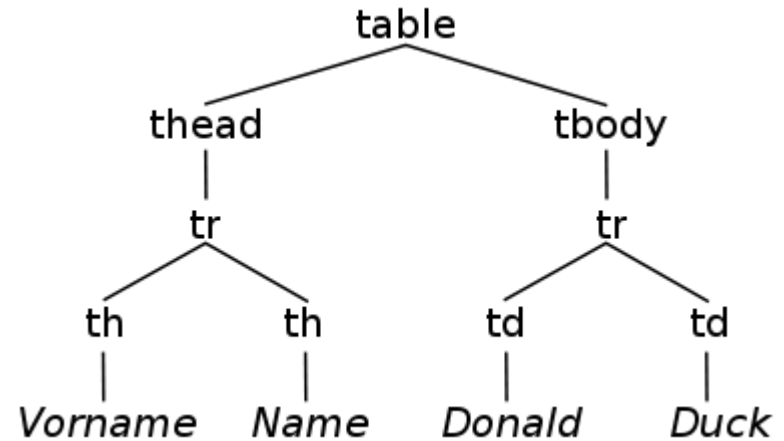
<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```

DOM

```
<table>
  <thead>
    <tr>
      <th>Vorname</th>
      <th>Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Donald</td>
      <td>Duck</td>
    </tr>
  </tbody>
</table>
```



Zusammenfassung

- Was ist das Internet und was ist das Web?
- Was ist ein Webdokument?
- Wie wird Struktur (HTML), Darstellung (CSS), Funktion (Webprogrammierung) und ihre Übertragung (HTTP) für das Web umgesetzt?
- Wie funktioniert HTTP?
- Welche Aufgabe haben Internetprotokolle?
- Was ist das OSI Referenzmodell?
- Was ist das Client - Server Paradigma?
- Welche Funktionen beinhalten “Frontend” und “Backend”?
- Was sind URIs? Wie sind sie aufgebaut? Welche Funktion haben sie?

Web-Entwicklung, VU

HTML

Was wissen wir schon?

HTML – Hypertext Markup Language – ist die einfache Markupsprache zur Beschreibung der Dokumente des WWW



Basiskomponente des WWW

Gibt weder Format noch Layout eines Dokumentes vor, sondern die Struktur der Inhalte

Einfache Beschaffenheit von HTML war und ist ein Grund für den Erfolg des WWW

Baut wie XML auch ursprünglich auf SGML - Standard Generalized Markup Language auf

...a long time ago

GML - IBM Generalized Markup Language: 1969



SGML - Standard Generalized Markup Language: 1986



HTML - Hypertext Markup Language: 1992

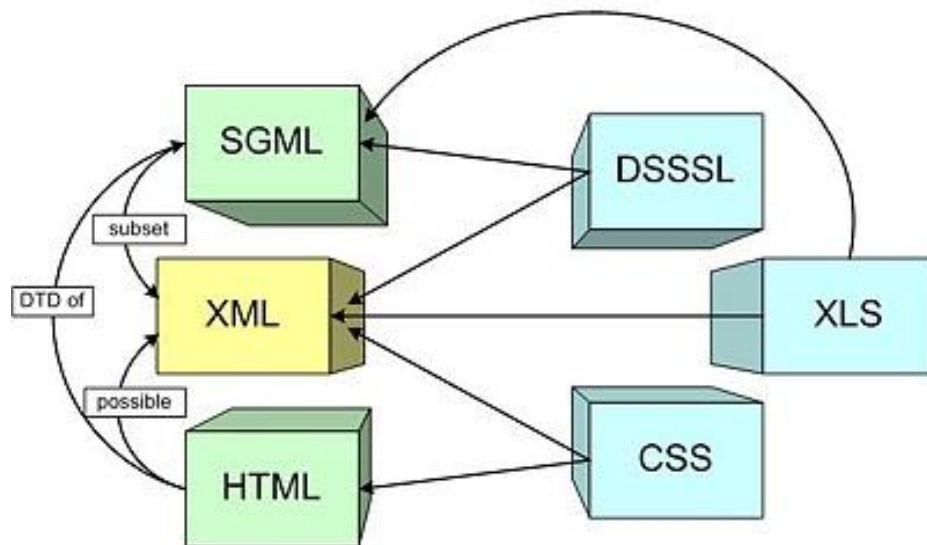


XML - Extensible Markup Language: 1998



SGML - XML - HTML

Relation HTML/XML/SGML



CSS Cascading Style Sheet
 DSSSL Document Style Semantics and Specification Language
 DTD Document Type Declaration
 XLS Extensible Style Sheet Language

HTML ist sozusagen in SGML formuliert (so wie TEI in XML)

Einheit 2

Trennung von Darstellung und Struktur: Strikt?

SGML sieht eigentlich eine strikte Trennung von Layout und Struktur vor

HTML weicht dies auf, weil das für die schnelle Realisierung des WWW damals günstig erschien

Heute ganz schlechter Stil → CSS benutzen

HTML5 streicht viele dieser Elemente:

`<basefont>` `` `<big>` `<center>` `<strike>` `<tt>`

→

CSS!

HTML - Struktur (Wiederholung)

Alle HTML Dokumente müssen mit einer *document type declaration* beginnen: **<!DOCTYPE html>**

Das HTML Dokument selbst beginnt mit **<html>** und endet mit **</html>**

Der sichtbare Inhalt eines HTML Dokuments steht zwischen **<body>** und **</body>**, Metadaten in **<head>**

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Überschrift 1</h1>
    <p>Absatz</p>
  </body>
</html>
```

Valid ✓

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titel</title>
  </head>
  <body>
    <h1>Überschrift 1</h1>
    <p>Absatz</p>
  </body>
```

Empfohlen! 😊

```
<!DOCTYPE html>
<title>Titel</title>
<h1>Überschrift 1</h1>
<p>Absatz</p>
```

Valid ✓ (nicht empfohlen!)

Guter Stil - Coding Conventions

Alle Tags schließen - Browser sind sehr nachsichtig

XML Syntax benutzen hilft (Oxygen hilft dabei) → `
` vs. `
`

Einheitlich Kleinbuchstaben für Tags und Attributnamen verwenden `<P Class="bLa">`

Werte von Attributen in Anführungszeichen `<p title=Ein Absatz>`

Keine Leerzeichen in Attributzuweisungen `<p class = "info">` vs. `<p class="info">`

Keine unnötigen Leerzeichen und Leerzeilen

- keine leeren Zeilen ohne triftigen Grund

- Leerzeilen nur zwischen größeren logischen Blöcken

- nicht jedes Element einrücken

- Einrückung nicht mit Tabulator-Taste, sondern mit 2 Leerzeichen

Guter Stil - Coding Conventions

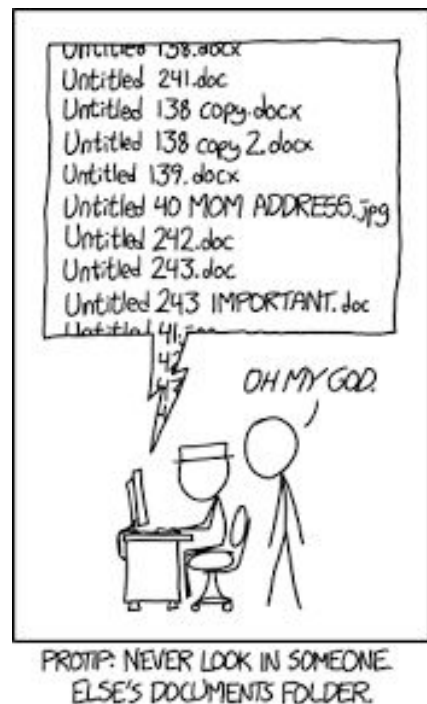
Dateinamen

Dateinamen in Kleinbuchstaben mit Unterstrichen (html_syntax.html)

Dateiendung für HTML: .html (bevorzugt) oder .htm

Dateiendung für CSS: .css

Dateiendung für JavaScript: .js



HTML - Struktur

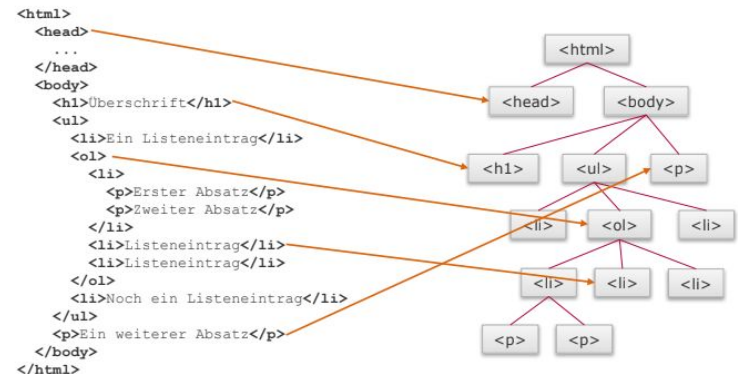
Ein sinnvolles HTML-Dokument besteht aus einem Head und einem Body

Der Header enthält Informationen über das Dokument, die nicht zum eigentlichen Inhalt gehören, z.B. Titel, Autor, Keywords, Kodierung, Sprache, ... (Metadaten)

Der Body enthält eigentlichen Inhalt des Dokuments, z.B. Überschriften, Absätze, Listen, Tabellen, Formulare, Links zu Bildern usw.

Derzeit ca. 120 Tags in HTML5

Immer eine Baumstruktur



HTML Head

Metadaten: Titel, Character Set, CSS Styles, Links, Scripts ...

Suchmaschinenoptimierung

Tags: <title>, <style>, <link>, <meta>, <script>, und <base>

<title>: Title im Browser Tab, in den Favoriten, in Suchergebnissen

<style>: CSS für eine einzige HTML Seite

<link>: externe CSS Stylesheets können damit geladen werden

<meta>: Character Set, Seitenbeschreibung, Keywords, Autoren ...

Spezialfall Viewport - um responsive zu sein immer diese Angabe im Head machen:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

HTML Head

<script>: für Client-seitiges JavaScript

```
<script>
function V1 {
    document.getElementById("V1").innerHTML = "Ein Test";
}
</script>
```

oder

```
<script src="/lib/1.0/plugins/matchHeight/matchHeight.js"></script>
```

<base>: hiermit kann eine Basis URL für alle relativen URLs auf einer Seite festgelegt werden (wird eher selten verwendet)

HTML Body

`<body>...</body>`

Unterscheidung zwischen Block- und Inline-Elementen

Block-Elemente nehmen die volle Breite ein (`<div>`, `<p>`, `<h1>`, `<main>` ...)

Inline-Elemente befinden sich im Fließtext (``, `<a>`, ``, `<i>`, `<input>` ...)

Kann in CSS verändert werden: `{display: block;}`, `{display: inline;}`

Container Elemente: `<div>` (Block), `` (Inline)

```
<div style="background-color:green;padding:100px;">
  <h1>HTML</h2>
  <p>HTML is da <span style="font-weight:bold">wahnsinn</span>.</p>
</div>
```

Semantische Elemente

`<div>` und `` eindeutig nicht semantisch

`<p>` eindeutig semantisch

Neue semantische Elemente in HTML 5:

`<header>` - Kopfzeile

`<nav>` - Navigationsmenü oben

`<main>` - Hauptinhalt

`<section>` - Irgendein thematischer Abschnitt (Einleitung, Kontaktinfos...)

`<article>` - Ein Artikel der für sich alleine Sinn macht

`<aside>` - z.B. Sidebar

`<footer>` - Fußzeile

`<details>` - Ein- und Ausblenden von Text

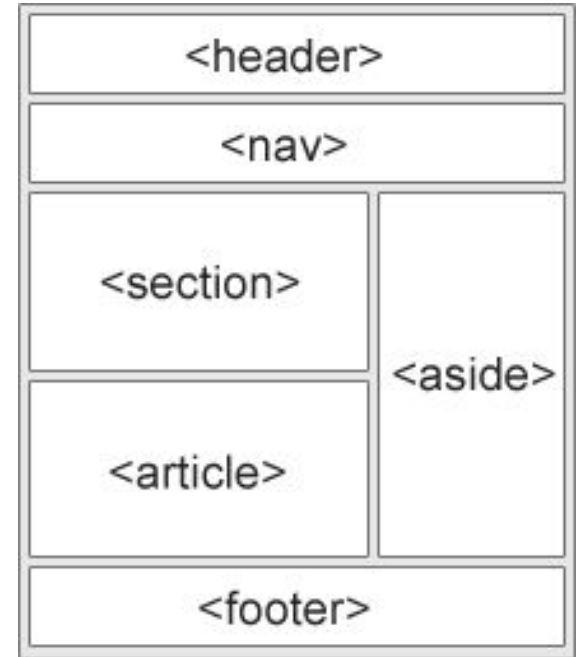
`<summary>` - Die Überschrift für das `<details>` Element

`<figure>` - Fotos, Diagramme, Illustrationen etc.

`<figcaption>` - Bildunterschrift für `<figure>`

`<mark>` - Text Highlighting (z.B. Suchergebnisse)

`<time>` - wichtig für Suchmaschinen, z.B. damit Google einem automatisch seine Flugzeiten etc. anzeigt



Wichtige Elemente

`<h1>-<h6>` Heading

`
` Break

`<a>` Anchor

`` Image

`<table>` Table

`<tr>` Table Row

`<th>` Table Heading

`<tr>` Table Row

`<td>` Table Data

`` Unordered List

`` List Item

`` Ordered List

`` List Item

Periodic table of HTML elements

| | | | | | | | | | | | | | | | | | | |
|---------|---------|------------|------|------|------|------|-------|------|--------|----------|----------|----------|----------|----------|----------|--|--|--------|
| html | | | | | | | | | | | | | | | | | | output |
| base | header | | | | | | | | strong | area | param | caption | th | input | progress | | | |
| head | h1-h6 | | | | | | | | sub | audio | source | col | thead | keygen | select | | | |
| style | hgroup | dt | ol | bdi | data | mark | ruby | sup | map | canvas | colgroup | tr | label | details | | | | |
| title | nav | figcaption | p | bdo | dfn | q | s | time | track | noscript | table | button | legend | dialog | | | | |
| body | section | figure | pre | br | em | rp | samp | u | video | script | tbody | datalist | meter | menu | | | | |
| address | dd | hr | ul | cite | i | rt | small | var | embed | del | td | fieldset | optgroup | menuitem | | | | |
| article | div | li | abbr | code | kbd | rtc | span | wbr | object | ins | tfoot | form | option | summary | | | | |
| footer | dl | main | b | | | | | | | | | | | | | | | |

content

element

shadow

template

base metadata sectioning text-content text-semantics media
embedded script edits table forms interactive web-components

<https://www.madebymike.com.au/demos/html5-periodic-table>

HTML Attribute

Die am häufigsten verwendeten Attribute:

href: `UNI Graz`

src: ``

alt: ``

class: `<p class="logos">`

id: `<p id="logo_1">`

lang: `<html lang="de-AT">`

style: `<p style="color:green">`

title: `<p title="Ein Tooltip">`

Entities

Entitäten in HTML = häufig verwendete Zeichenketten

Format: &Kennung;

Sonderzeichen, die für Markup reserviert sind: < – < und > – >

Non-Breaking Space: = geschütztes Leerzeichen

Prof. Dr. Pollin

99 Luftballons

& - &

€ - €

“ - "

© - ©

Layout

Möglichkeiten für ein Gesamt-Layout Design:

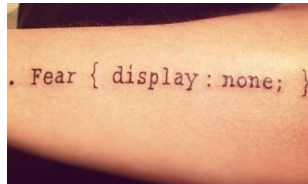
HTML Tabellen: schlechteste Methode, früher benutzt aber heute sinnlos und chaotisch

CSS Framework: Schnell zu benutzen, Responsive, große Auswahl (Bootstrap, Bulma, PureCSS, W3.CSS ...)

CSS Float Property: float und clear properties auch schnell zu erlernen, schwierig alles responsive zu halten

CSS Flexbox: neues Layout Modul in CSS3, extra für Responsive Design entwickelt, nicht so leicht zu lernen

CSS Grid: noch neueres Layout Modul in CSS, ebenfalls voll responsive, nicht so leicht zu verstehen



Fazit: Frameworks benutzen oder Flexbox oder Grid lernen ODER beides lernen und kombinieren für die Super-CSS-Geeks

Validierung

Validator: <https://validator.w3.org>

Welcher Browser unterstützt was: <https://caniuse.com>

Wie komme ich zu Daten?

<https://www.opendatasoft.com/a-comprehensive-list-of-all-open-data-portals-around-the-world/#/austria>

<https://www.opendataportal.at/>

<https://www.forbes.com/sites/bernardmarr/2016/02/12/big-data-35-brilliant-and-free-data-sources-for-2016/#63cee9dab54d>

<https://openhumanitiesdata.metajnl.com/>

<http://www.digitale-edition.de/>

http://humanities-data-centre.org/?page_id=220

<http://gams.uni-graz.at>

Web-Entwicklung, VU

JavaScript, JSON, WebStorage

jQuery

Agenda

- Fragen?
 - **17.5** Ausgabe: Assignment 3
 - außerordentliche Fragestunde: 28 - 30 Mai oder 4. - 6. Juni
 - **7.6** Abgabe: Assignment 3
- JSON
- JavaScript
 - WebStorage
 - Array
 - JSON
 - Dynamisch HTML-Elemente erzeugen
- jQuery
 - Aufbau, Syntax
 - Live-Demo

Verflichte FOR-Schleife!!!

https://www.w3schools.com/js/js_loop_for.asp

```
for (BEGIN; END; ITERATION) {  
    code block to be executed  
}
```

```
for (i = 0; i < 5; i++) {  
    text += "The number is " + i + "<br>";  
}
```

FOR/IN Schleife

```
var person = {fname:"John", lname:"Doe", age:25};  
var text = "";  
var x;  
  
for (x in person) {  
    text += person[x];  
}
```


JSON - JavaScript Object Notation

- Austausch von Daten zwischen Browsern und Servern
- JSON ist nur String, der in JavaScript zu Objekten konvertiert wird.
- Selbstbeschreibend, sprachunabhängig
- Macht die Verarbeitung relativ einfach, kein kompliziertes Parsen und Übersetzen.
 - *“Lightwight data-interchange format”*

JSON vs XML

```
{  
  "Herausgeber": "Xema",  
  "Nummer": "1234-5678-9012-3456",  
  "Deckung": 2e+6,  
  "Waehrung": "EURO",  
  "Inhaber":  
    {  
      "Name": "Mustermann",  
      "Vorname": "Max",  
      "maennlich": true,  
      "Hobbys": ["Reiten", "Golfen", "Lesen"],  
      "Alter": 42,  
      "Kinder": [],  
      "Partner": null  
    }  
}
```

<Kreditkarte

```
Herausgeber="Xema"  
Nummer="1234-5678-9012-3456"  
Deckung="2e+6"  
Waehrung="EURO">
```

<Inhaber

```
Name="Mustermann"  
Vorname="Max"  
maennlich="true"  
Alter="42"  
Partner="null">
```

<Hobbys>

```
<Hobby>Reiten</Hobby>  
<Hobby>Golfen</Hobby>  
<Hobby>Lesen</Hobby>
```

</Hobbys>

<Kinder />

</Inhaber>

</Kreditkarte>

JSON Syntax leitet sich vom JavaScript- Objekt ab

```
var person = { "name":"John", "age":31, "city":"New York" };  
person.name;           // returns John  
person["name"];        // returns John
```

Aus JavaScript Objekt JSON erzeugen mit JSON.stringify()

```
var myObj = { "name":"John", "age":31, "city":"New York" };  
var myJSON = JSON.stringify(myObj);  
console.log(myJSON);
```

JSON erhalten und parsen mit JSON.parse()

```
var myJSON = '{ "name":"John", "age":31, "city":"New York" }';  
var myObj = JSON.parse(myJSON);  
console.log(myObj.name);
```

WebStorage und Data Attributes

Beispiel auf

<http://glossa.uni-graz.at/context:szd>

HTML Web Storage (https://www.w3schools.com/html/html5_webstorage.asp)

Damit können Daten lokal beim Client im Browser gespeichert werden.

vorher nur cookies

Web Storage ist sicherer, mehr Speicherplatz und beeinflusst nicht die Performance der Seite

webStorage.js

Pause

jQuery

“jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.”

<http://jquery.com>

“With jQuery you select (query) HTML elements and perform "actions" on them.”

hellojquery.html

//jQuery ist ein selbständiges JavaScript File, wird auch so eingebunden

// hier von einem CDN (Content Delivery Network).

<script

src="<https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js>">

</script>

<script>

\$(document).ready(function(){

\$("#p").click(function(){

\$(this).hide();

});

});

</script>

jQuery Syntax

```
$(document).ready(function){  
    $("p").click(function(){  
        $(this).hide();  
    });  
});
```

Grundlegende Syntax: **\$(selector).action()**

- \$ um jQuery zu definieren
- DOM
- Prüft ob Dokument ganz geladen ist.
- jQuery Selector: wählt alle <p> aus
- Ist ein Event: “wenn darauf geklickt wird”
- “Dann rufe **function()** auf, die
\$(this).hide()
Für das selektierte Elemente die Funktion
hide() aufruft und
Das Element versteckt
(das CSS im Hintergrund manipuliert.)

jQuery User Interfaces

<https://jqueryui.com>

Autocomplete

<https://jqueryui.com/autocomplete>

Draggable

<https://jqueryui.com/draggable>

Color Animation

<https://jqueryui.com/animate>

jQuery_highlight_autocomplete.js

```
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script>
$( function() {
  var availableTags = [
    "green","red","yellow","rubin", "grey",
    "blue","black","orange"
  ];
  $( "#color" ).autocomplete({
    source: availableTags
  });
} );
</script>
```

Syntax

https://www.w3schools.com/jquery/jquery_syntax.asp

jQuery Selectors

https://www.w3schools.com/jquery/jquery_selectors.asp

Events

https://www.w3schools.com/jquery/jquery_events.asp

jQuery Effects - Hide and Show

https://www.w3schools.com/jquery/jquery_hide_show.asp

Zusammenfassung

Wie ist jQuery aufgebaut?

Was ist JSON und wofür wird es verwendet?

Wie unterscheidet sich JSON von XML?

Was sind die Stärken von jQuery?

Was ist der WebStorage?

Warum wird er verwendet?

Web-Entwicklung, VU

Bootstrap 4

Exkurs: XHTML Namespace

Oxygen erstellt automatisch den XHTML Namespace, auch wenn wir HTML als “neue Datei” auswählen

Bitte wieder löschen!

```
<html xmlns="http://www.w3.org/1999/xhtml">
```



```
<html>
```

Was wissen wir schon?

CSS Frameworks bzw. Front-End Frameworks sind Sammlungen von Elementen um Webdesign zu vereinfachen

Größtes Framework: Bootstrap

Basis: Gestaltungsraster mit Flexbox realisiert

Modular aufgebaut

Einbindung über CDN oder lokal/eigener Server

css und js files

Grid (Raster) System mit 5 Klassen

Bootstrap 4 Grid System Übersicht

`.col-` (extra small devices - Bildschirmbreite weniger als 576px)

`.col-sm-` (small devices - gleich oder größer als 576px)

`.col-md-` (medium devices - gleich oder größer als 768px)

`.col-lg-` (large devices - gleich oder größer als 992px)

`.col-xl-` (xlarge devices - gleich oder größer als 1200px)

Anzahl der möglichen Columns: 12

Eine Zeile: `class="row"`

Gleich breite Columns (Spalten): `class="col"`

Responsive Columns: `class="col-*-*" (zb class="col-sm-3")`

<https://getbootstrap.com/docs/4.1/layout/grid/>

Bootstrap 4 Typographie

Display Heading: größer als normale Headings `.display-1`, `.display-2`, `.display-3`, `.display-4`

`<small>`: Eine Unterüberschrift bzw. weniger wichtiger Text:

```
<h1>Überschrift 1 <small>Unterüberschrift</small></h1>
```

`<mark>`: gelber Hintergrund

`<abbr>`: gepunkteter Unterstrich `<abbr title="Zentrum für Informationsmodellierung">ZIM</abbr>`

`<blockquote class="blockquote">` und `<footer class="blockquote-footer">` für Zitate mit Quelle

Bootstrap 4 Typographie

`<code>` = einzeliger Programmcode

`<kbd>` = Keyboard

`<pre>` = mehrzeiliger Programmcode

Klassen:

`.small`

`.lead`

`.font-italic`

... (<https://getbootstrap.com/docs/4.1/content/typography/>)

Bootstrap 4 Tabellen

`.table`

`.table .table-borderless`

`.table .table-striped`

`.table .table-bordered`

`.table .table-hover`

`.table .table-dark`

Kombinationen:

`.table .table-dark .table-hover`

Bootstrap 4 Tabellen

Responsive Tabelle:

```
<div class="table-responsive">  
  <table class="table">  
    ...  
  </table>  
</div>
```

<https://getbootstrap.com/docs/4.1/content/tables/>

Bilder

`.rounded`

`.rounded-circle`

`.img-thumbnail`

Responsive:

`.img-fluid`

<https://getbootstrap.com/docs/4.1/content/images/>

Bootstrap Navbar

Der Klassiker:

https://drive.google.com/open?id=1i02sS0MGeWgCwt6zDog_fSPEwQ0nF9h7

Navbars sind komplex!

- Buttons
- Dropdowns
- Forms
- Inputs
- Colors
- Utilities

Übung

Bitte zu zweit (auf einem Laptop coden, auf dem anderen im Google Doc Fragen und Antworten sammeln)!

Gestaltet eine Bootstrap Website mit **Header**

Navbar (egal welche Art, also mit/ohne Dropdown, vertikal/horizontal, mit/ohne Suche, Farben frei wählen)

Main Content mit einer Tabelle und einem oder mehreren Bildern und einer Row mit 2 Columns (Breite selbst wählen!) und Dummy Text und

Footer

Ihr könnt Bootply oder was auch immer ihr findet verwenden oder lokal coden!

Ressourcen

<https://www.w3schools.com/bootstrap4/default.asp>

<https://getbootstrap.com/docs/4.1/getting-started/introduction/>

<https://www.bootply.com/>

<https://hackerthemes.com/bootstrap-cheatsheet/> !

<https://hackerthemes.com/bootstrap-buffer/> !

<https://bootbites.com/>

Fragen und Antworten, Tipps, Links

Wissen aus unserer “Cloud” bitte hier zusammenfassen oder einfach drauf los fragen (es gibt keine blöden Fragen;):

<https://docs.google.com/document/d/1V6PEGRtWszthyuLnXlKcCu-T9nll2ks8W15RbbgTC4g/edit?usp=sharing>

Mögliche Fehlerquellen

Bootstrap 3 Code kopiert, aber Bootstrap 4 eingebunden

Erster (fehlerhafter) Versuch ist noch im Cache (Strg+F5)

Auf CSS Hierarchien achten

Web-Entwicklung, VU

XSLT - Projekt

Agenda

- Ass 3
- XSLT, Xpath, XSLT-Processor
- XML to HTML
- Arbeit am Projekt

Ass3 - Fragen?

Projekt-Abgabe

- Abgabe der Ausgangsdaten
- XSLT

XSL Transformation

XSL (eXtensible Stylesheet Language) ist eine Programmiersprache um XML zu verarbeiten, zu transformieren und zu manipulieren.

“XSLT transforms an XML source-tree into an XML result-tree.”

XPath ist eine Sprache um in XML Dokumenten zu navigieren.

https://www.w3schools.com/xml/xsl_intro.asp

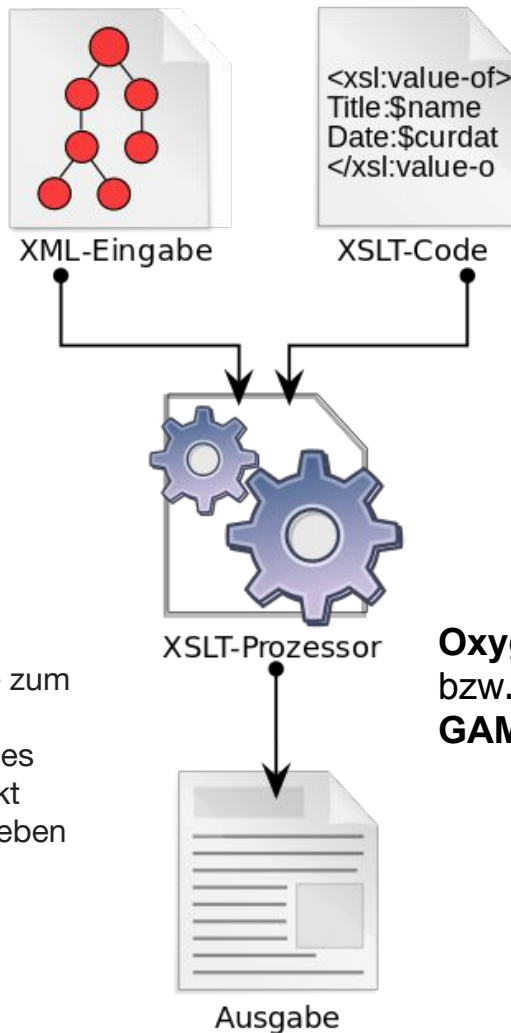
Eure Daten

bzw.

http://glossa.uni-graz.at/o:szd.werke/TEI_SOURCE

Ein **XSLT-Prozessor** ist eine Software zum Umwandeln von Dokumenten mittels eines XSLT-Stylesheets. Das Ergebnis ist ein neues Dokument, das (z. B. als [DOM-Objekt](#)) direkt weiterverarbeitet oder in eine Datei ausgegeben werden kann.

<https://de.wikipedia.org/wiki/XSLT-Prozessor>



Eure Abgabe für das Projekt

bzw.

<http://glossa.uni-graz.at/gamsdev/pollin/szd/trunk/www/szd-Werke.xsl>

Oxygen
bzw.
GAMS

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <biblFull xml:id="SZDBIB.365">
3   <fileDesc>
4     <titleStmt>
5       <title type="Titel">Proust : recherches sur la création intellectuelle</title>
6       <author ref="http://d-nb.info/gnd/12490310X" role="Verfasser">Abraham, Pierre</author>
7     </titleStmt>
8     <publicationStmt>
9       <publisher>Rieder</publisher>
10      <date>1930</date>
11      <pubPlace>Paris</pubPlace>
12    </publicationStmt>
13    <seriesStmt>
14      <title type="Reihe">Maîtres des littératures ; 7</title>
15    </seriesStmt>
16    <sourceDesc>
17      <msDesc>
18        <msIdentifier>
19          <settlement>London</settlement>
20          <repository ana="https://gams.uni-graz.at/o:szd.standorte#SZDSTA.18">Erben Stefan Zweigs</repository>
21          <idno>A.I.01</idno>
22          <altIdentifier corresp="https://gams.uni-graz.at/o:szd.glossar#Grossbuchstabe1">
23            <idno>B 3</idno>
24          </altIdentifier>
25        </msIdentifier>
26        <msContents>
27          <textLang>
28            <lang>fre</lang>
29          </textLang>
30        </msContents>
31        <physDesc>
32          <objectDesc>
33            <supportDesc>
34              <extent>
35                <measure type="Seiten">91, LX</measure>
36                <measure type="Format">8°</measure>
37              </extent>
38            </supportDesc>
39          </objectDesc>
40          <!-- ... -->
41 </biblFull>

```

http://gams.uni-graz.at/o:szd.bibliothek/TEI_SOURCE



▼ Proust : recherches sur la création in... | 1930



| | |
|--------------------|--|
| Verfasser/in | Abraham, Pierre 🔍 |
| Titel | <i>Proust : recherches sur la création intellectuelle</i> |
| Veröffentlichung | Paris : Rieder, 1930 |
| Reihe | Maîtres des littératures ; 7 |
| Sprache | Französisch |
| Umfang/Einband | 91, LX Seiten : illustriert. 8°, Broschur |
| Widmung ⓘ | Pour Stefan Zweig, en sympathie et en admiration Pierre Abraham. |
| Marginalie ⓘ | Anmerkungen von Stefan Zweig in Bleistift |
| Merkzeichen ⓘ | Unterstreichungen in Bleistift |
| Originalsignatur ⓘ | B 3 |
| Heutiger Standort | Erben Stefan Zweigs, London 🔍 A.I.01 |

<http://gams.uni-graz.at/o:szd.bibliothek>

Xpath

| Expression | Description |
|-----------------|---|
| <i>nodename</i> | Selects all nodes with the name " <i>nodename</i> " |
| / | Selects from the root node |
| // | Selects nodes in the document from the current node that match the selection no matter where they are |
| . | Selects the current node |
| .. | Selects the parent of the current node |
| @ | Selects attributes |

Guter Überblick über alle Möglichkeiten:

https://www.w3schools.com/xml/xpath_syntax.asp

XPath - TEI_SOURCE_SZDDatei

//biblFull

//biblFull/fileDesc/titleStmt/author

//biblFull/fileDesc/titleStmt/title[@type="Einheitssachtitel"]

//biblFull[.//term[@type='Objektyp']/text() = 'Notizbuch']

//item[@ana='Beilage']/ancestor::fileDesc//title

Xpath - Übung

Gib mir alle Signaturen von Werken (biblFull), die die Ordnungskategorie 'Romane/Erzählungen' und eine Beilage haben.

Lösung

//biblFull

[./item

 [@ana='Beilage']

]

[./term [@type='Ordnungskategorie'] = 'Romane/Erzählungen']

./idno

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
  xmlns="http://www.w3.org/1999/xhtml" xmlns:t="http://www.tei-c.org/ns/1.0">
  <xsl:output method="xml" doctype-system="about:legacy-compat" encoding="UTF-8" indent="no"/>
```

```
  <xsl:template match="/">
    <html lang="en">
      <head><title>Stefan Zweig digital</title></head>
      <body>
        <h1>Stefan Zweig digital - Beispiel</h1>
        <h2>Werke</h2>
        <ul class="list-group">
          <xsl:apply-templates select="//t:listBibl/t:biblFull"/>
        </ul>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="//t:listBibl/t:biblFull">
    <li class="list-group-item">
      <xsl:apply-templates/>
    </li>
  </xsl:template>
</xsl:stylesheet>
```


XSLT-Template: TEItoHTML.xsl

- Template für weitere Projekte.
- Bootstrap 4, CSS, JavaScript, jQuery eingebunden.

Übung

Eigene Daten mit dem TEItoHTML.xsl als index.html ausgeben.

XSLT-szd-Werke.xsl Datei

Clarissa

Some small text here.

Autor: Zweig Stefan

```
<a data-toggle="collapse" href="{concat( '#', generate-id())}" role="button" class="list-group-item  
list-group-item-action flex-column align-items-start">
```

```
<div class="d-flex w-100 justify-content-between">
```

```
<h5 class="mb-1">
```

```
<xsl:apply-templates select="t:fileDesc/t:titleStmt/t:title[1]"/>
```

```
</h5>
```

```
<small>
```

```
<xsl:text>Autor: </xsl:text>
```

```
<xsl:value-of select="t:fileDesc/t:titleStmt/t:author[1]"/>
```

```
</small>
```

```
</div>
```

```
<p class="collapse">Enter any content here.</p>
```

```
<small>Some small text here.</small>
```

```
</a>
```

href="#d1e147"

```
<div id="{generate-id()}" class="collapse">  
  <table class="table table-dark table-striped">  
    <tbody>  
      <xsl:apply-templates select="."/t:msIdentifier/*"/>  
    </tbody>  
  </table>  
</div>
```

id="d1e147"

| | |
|--|---------------------|
| Clarissa Some small text here. | Autor: Zweig Stefan |
| country | |
| Österreich | |
| settlement | |
| Salzburg | |
| repository | |
| Literaturarchiv Salzburg | |
| idno | |

Zusammenfassung

Was ist XSLT und für was wird es verwendet in der Webentwicklung?

Was ist XPath?

Was ist ein XSLT-Processor?

Was muss in einem, TEI/XML nach HTML Szenario mittels XSLT bedacht werden?

Web-Entwicklung, VU

Zusammenfassung und Tipps

Organisatorisches

Projektpräsentation 28.06.

5 min max!

Webseite präsentieren, Intention, Datenbasis, ...

Abgabe-Projekt + Abgabegespräch

Umfang:

Ausgangsdaten + XSLT als .zip

Doodle:

<https://doodle.com/poll/erebs4wawcbpph5n>

Juli-Termin:

02.07-06.07

Oktober-Termin

01.10-05.10

Feedback zur LV auf Moodle

Abgabegespräch

Stoffumfang sind ausgehend von Folien die behandelten Themengebiete + Selbststudium dieser Dinge im 'Wikipedia-Ausmaß' (Zusammenfassung jeweils auf letzter Folie)

Fragen zum JavaScript bzw XSLT Code

“An Stelle xy im Code steht folgendes...was macht das und warum”

“Gegeben ist function(): Was ist der output?”

Zusammenfassung

- Internet / WWW
- HTML
- CSS
- CSS Frameworks
- Bootstrap
- JavaScript
- jQuery
- JSON, WebStorage
- XSLT

Internet / WWW

- Internet: die technische Infrastruktur
 - Protokolle (HTTPS, FTP, TCP ...)
- WWW: Hypermediasystem mit weltweit verteilten Ressourcen
- Web-Technologien
 - Frontend
 - Backend
- URI - URL - URN
- Client Server Architektur
 - HTTP zur Kommunikation

HTML

Benutzt semantische Elemente wann immer möglich!

Validiert eure HTML Dokumente immer mal wieder: <https://html5.validator.nu/>

Benutzt die DevTools von Chrome / Inspector von Firefox! → Ctrl + Shift + I (Win)
oder Command + Option + I (Mac)

Arbeitet mit Copy&Paste → `<meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"> etc.`

Pflegt eure Meta Tags! Seitenbeschreibung, Keywords, Literatur, Autoren ...

Achtet auf die korrekte Verwendung von Block- bzw. Inline-Elementen

CSS

- Achtet auf die Hierarchie! →
1. Inline
 2. Interne
 3. Externe
 4. Unterste Position im <head> gewinnt

CSS Selektoren braucht ihr auch für jQuery:

https://www.w3schools.com/cssref/css_selectors.asp

Arbeitet mit den Attributen id und class

Benutzt ein CSS Framework!

Bootstrap

Arbeitet mit Copy&Paste: <https://getbootstrap.com/docs/4.1/getting-started/introduction/> → Starter Template

CDNs machen das Leben leichter!

Auf Abhängigkeiten achten und DAVOR einbinden! (“JavaScript dependencies (jQuery and Popper.js)”

Testet immer wieder auf Tablet und Handy bzw. die Funktion dazu in den DevTools

Für die CSS (und nicht SCSS) Anweisungen in den DevTools: settings → sources → enable css source maps

Nutzt die Community: Stackoverflow, Google → Recherche oft zeitsparender

Webprogrammierung

JavaScript - DOM

https://docs.google.com/presentation/d/1oYM1zKHTiccRPLYI_qGYHmI4Iq6bAPI5F-N3zTaryxk/edit#slide=id.g30ed2b0aa8_0_0

jQuery - JSON - jQuery

https://docs.google.com/presentation/d/19oNKra_1ly7MOffTdaGBbALSY1fncO0AngagemUiz2c/edit#slide=id.g32f45c9b3b_0_0

Client-Server

https://docs.google.com/presentation/d/1Ke3024UR3GzdMlsxHQuPN5VFHFrCOtQ6qPoKVUoWjo/edit#slide=id.g346b93f9cb_0_0

XSLT-Prozessor / XSLT- Workflow

https://docs.google.com/presentation/d/1jtzn_yn8XX68whFFTs0ok42OUUsAnQvysTh6OvyN4RIM/edit#slide=id.g37ef34eb3a_0_11

Ein Lösungsweg für Ass3
