

Webentwicklung



DHBW SS2018 – Elias Henrich

Elias Henrich

B. Sc. Medizintechnik

M. Sc. Informationstechnik

Senior Backend Entwickler (PHP) bei
netz98.de

Kontakt: elias@eliashenrich.de

Einführung

- Desktop- vs. Web-Apps
- Das WWW - Aufbau und Funktion des Internets
- HTTP-Protokoll
- Unsere Entwicklungsumgebung unter Windows, MacOS oder Linux

Themen

HTML



- vom Jahr 1991 bis HTML5
- HTML-Tags
- Das Document Object Model
- Interaktivität durch Formulare
- Bilder- und Grafikformate
- Styling mit HTML?
- Neue Features mit HTML5

Quelle: The Evolution of Web Data

Themen

Styling mit CSS

- CSS1 bis 3
- Eigenschaften von HTML-Objekten
- Das Box-Model
- Selektoren & Pseudoklassen
- Responsive Design
- CSS3: Transitions
- Das Twitter Bootstrap-Framework



Quelle: The Evolution of Web Data

Themen

JavaScript



- Basics & wichtige Sprachbestandteile
- Dynamische Inhalte durch AJAX
- JS mit HTML und CSS
- JSON und das REST-Protokoll
- Die jQuery-Bibliothek

Themen

Optional



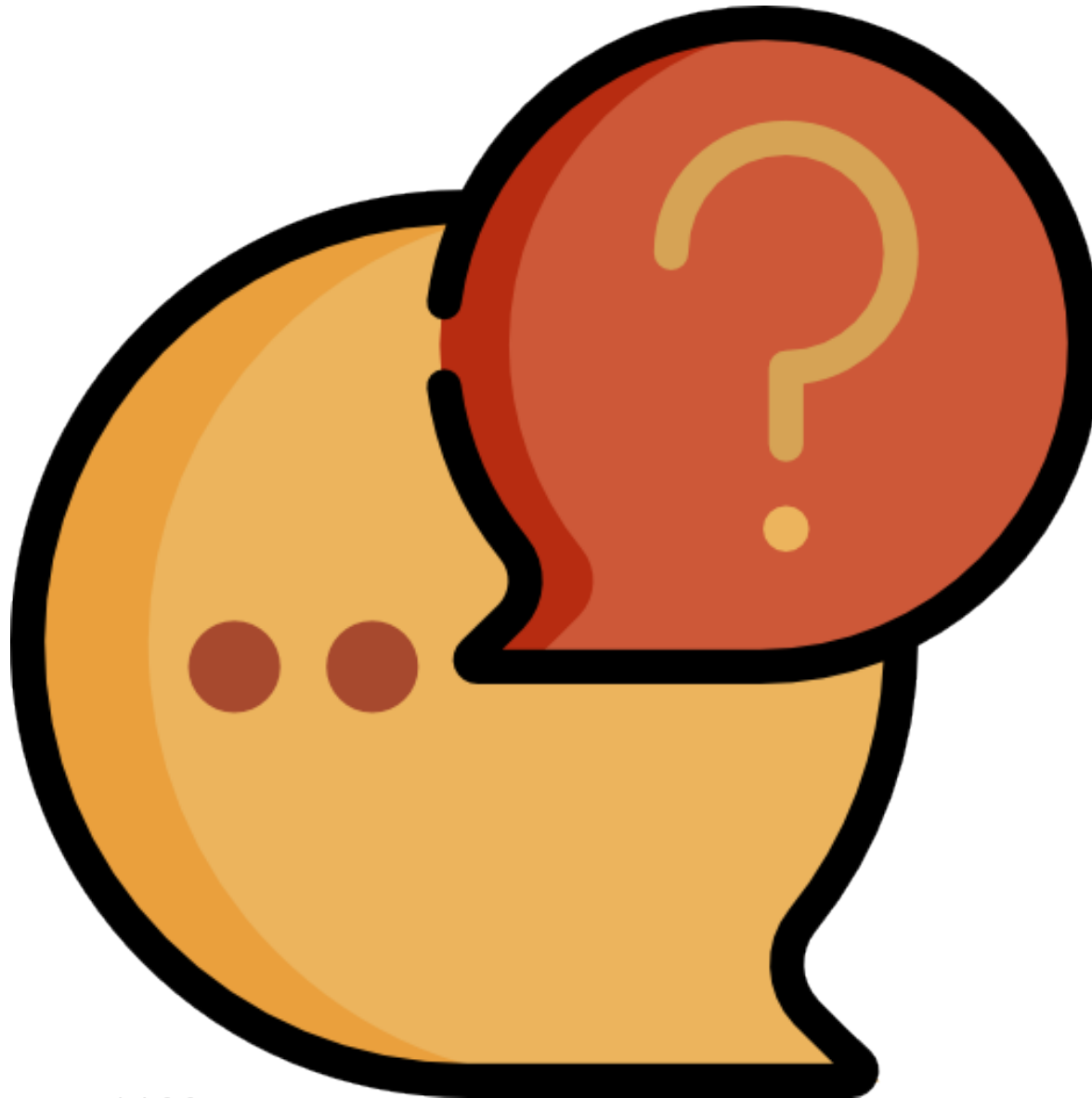
- robuste Anwendungen durch Web-Security
- Backend-Programmierung mit PHP
- von der Web- zur Mobile-App
- Karten mit Leaflet
- und viele mehr...

Organisatorisches

Prüfungsleistung

- Bearbeitung eines vorgegebenen oder abgesprochenen Themas
 - ca. 2-3 Wochen Vorbereitung
 - Code- & Handout-Abgabe in Git-Repository
 - 10-minütiger Vortrag mit Code-Demo und Q&A
- i. d. R. keine weiteren Hausaufgaben,
dennoch Nachbereitung wichtig!

Fragen zu Inhalt oder Prüfungsleistung?



Wieso Web-Apps?

Summary of key survey findings:

Web Technologies are Critical to Business

- 93 percent say web technologies are critical to their strategy for desktop and mobile.
- 75 percent increased web technology investments last year; 76 percent plan to increase next year.
- Web technology use is driven primarily by ease of support for multiple browsers and screen sizes (68 percent), development efficiency (62 percent), quality of applications (57 percent), and time to market (54 percent).

Desktop Applications are Far from Obsolete, but Legacy Browsers are Definitely on the Way Out

- 80 percent say desktop applications are “absolutely essential.”

Desktop-Anwendungen

- auf dem Rechner installiert, laufen nur dort
- nahezu unlimitierte Datenmengen
- optimale Auslastung bei intensiven Arbeiten
(z.B. Nvidia Cuda)
- direkter Zugriff auf Gerätehardware
(z.B. spezielle Industriekomponenten)
- offline verfügbar

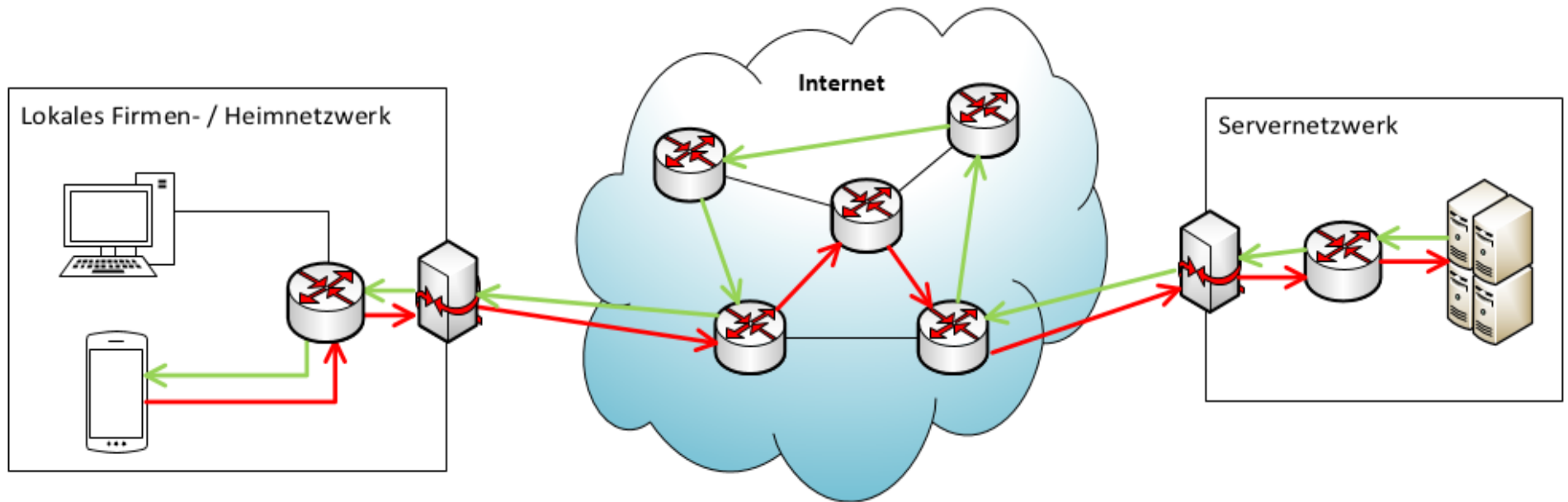
Web-Anwendungen

- Programmcode auf Webserver
- Ausführung im Client-Browser (Desktop-App)
- unabhängig vom Gerätetyp (Desktop, Tablet, Smartphone)
- keine Installation notwendig
- Entwicklung unabhängig vom Betriebssystem

Abgrenzung

- HTML5 löst viele Probleme der Web-Apps
- Je nach Anforderung überwiegen Vorteile beider Methoden
- daher: genaues Abwägen vor Programmentwurf notwendig!
- Alternative: Hybride Apps vereinen durch nativen Container beide Welten

Das Internet: Aufbau und Funktion



- Routing & Laufzeiten können variieren
- Datenpakete kommen evtl. in falscher Reihenfolge an

Das Internet: Aufbau und Funktion

- viele autonome Systeme (z.B. Firmen- und Ländernetze)
- dezentrale miteinander verknüpft
- offene, nicht proprietäre Protokolle als Grundlage
- Am wichtigsten:

HTTP

HyperText Transfer Protocol

HTTPS

Erweiterung von HTTP mit SSL

Übung: HTTP-Header

Nutzen Sie die Developer Tools, um folgende Infos über **<http://eliashenrich.de>** zu sammeln:

- 1) Wie bauen sich HTTP-Anfrage und -Antwort auf?
- 2) Welche HTTP-Version wird genutzt?
- 3) Wie viele Elemente werden abgerufen? Wozu?
- 4) Wird eine Anfrage geblockt? (AdBlocker!)
- 5) Gibt es Unterschiede zwischen erster und zweiter Anfrage?
- 6) Wie ist das Verhältnis von Kopf- zu Nutzdaten?
Sehen Sie hier Probleme?

Übung: HTTP-Header

Chrome: Rechtsklick → *Untersuchen* → *Network*

Firefox: Rechtsklick → *Element untersuchen* → *Netzwerkanalyse*

Besonderheit in Chrome:

▼ Request Headers

⚠ Provisional headers are shown

Referer: http://www.eliashenrich.de/
User-Agent: Mozilla/5.0 (Windows NT

?



Übung: HTTP-Header

1) Wie bauen sich HTTP-Anfrage und -Antwort auf?

5.1 Request-Line

The Request-Line begins with a method token, followed by the Request-URI and the protocol version, and ending with CRLF. The elements are separated by SP characters. No CR or LF is allowed except in the final CRLF sequence.

```
Request-Line = Method SP Request-URI SP HTTP-Version CRLF
```

5.1.1 Method

The Method token indicates the method to be performed on the resource identified by the Request-URI. The method is case-sensitive.

```
Method = "OPTIONS"           ; Section 9.2
      | "GET"                 ; Section 9.3
      | "HEAD"               ; Section 9.4
      | "POST"               ; Section 9.5
      | "PUT"                ; Section 9.6
      | "DELETE"             ; Section 9.7
      | "TRACE"              ; Section 9.8
      | "CONNECT"            ; Section 9.9
      | extension-method
extension-method = token
```

Übung: HTTP-Header

1) Wie baut sich eine HTTP-Anfrage auf?

GET

ist die gebräuchlichste Methode. Mit ihr wird eine Ressource (zum Beispiel eine Datei) unter Angabe eines [URI](#) vom Server angefordert. Als Argumente in dem URI können also auch Inhalte zum Server übertragen werden, allerdings soll laut Standard eine GET-Anfrage nur Daten abrufen und sonst keine Auswirkungen haben (wie Datenänderungen auf dem Server oder ausloggen). Die Länge des URIs ist je nach eingesetztem Server begrenzt und sollte aus Gründen der Abwärtskompatibilität nicht länger als 255 [Bytes](#) sein. (siehe [unten](#))

Übung: HTTP-Header

1) Wie bauen sich HTTP-Anfrage und -Antwort auf?

Request headers:

```
Host: eliashenrich.de
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: de,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: _pk_id.4.12d3=46db16ddc9aefe88.1528141939.1.1528141939.1528141939.
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

Übung: HTTP-Header

1) Wie bauen sich HTTP-Anfrage und -Antwort auf?

Response headers:

```
HTTP/1.1 200 OK
Date: Tue, 05 Jun 2018 17:36:46 GMT
Server: Apache
Last-Modified: Sun, 14 May 2017 21:01:56 GMT
ETag: "26886d5-5b3-54f823faf3e72"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 721
Keep-Alive: timeout=2, max=1000
Connection: Keep-Alive
Content-Type: text/html
```

Übung: HTTP-Header

5) Gibt es Unterschiede zwischen erster und zweiter Anfrage?

An HTTP cookie (web cookie, browser cookie) is a small piece of data that a server sends to the user's web browser, that may store it and send it back together with the next request to the same server. Typically, it's used to know if two requests came from the same browser allowing to keep a user logged-in, for example. It remembers stateful information for the stateless HTTP protocol.

Fazit: HTTP

- Grundlage jeder Webanwendung
- jedes Seitenelement benötigt eine eigene HTTP-Request
- HTTP1.1: Eine Verbindung = mehrere Übertragungen
- GET-Methode am verbreitetsten
- User kann Teile der Anwendung blockieren
- zustandslos
 - durch Cookies (+ andere Methoden) zustandsbehaftet

Entwicklungsumgebung

- Webbrowser (empfohlen: Chrome oder Firefox)
- Developer Tools
- Text-Editor, z.B.
 - Windows Notepad
 - Notepad++
 - Sublime Text 3
 - Microsoft Visual Studio Code
 - IntelliJ WebStorm bzw. PHPStorm
- GitHub
- evtl. Grafikeditor, z.B. Paint.Net, GIMP

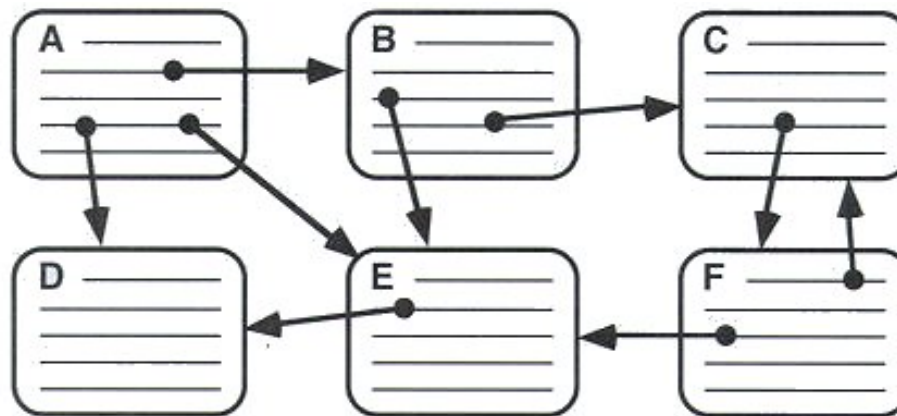


HTML: Einführung

Hypertext Markup Language

- Hypertext is text which is not constrained to be linear.
- Hypertext is text which contains links to other texts.
The term was coined by Ted Nelson around 1965.

Quelle: W3C



Quelle:

<http://webdoc.sub.gwdg.de/edoc/ia/eese/schreiber/Hsystem.jpg>

HTML: Einführung

Hypertext Markup Language

markup - definition and synonyms



NOUN



Pronunciation

/ˈmark,ʌp/

Word Forms



Contribute to our Open Dictionary

- 1 [COUNTABLE] the amount of money that someone who sells goods adds to the price of the goods in order to make a profit
a fixed markup of 20 percent



Synonyms and related words

Profit: profit, earnings, percentage...

Explore Thesaurus

- 2 [UNCOUNTABLE] COMPUTING instructions added to documents that tell a computer how to print or organize the information

HTML: Einführung

Hypertext Markup Language



„Sprache zur Beschreibung der Darstellung von Hypertexten“

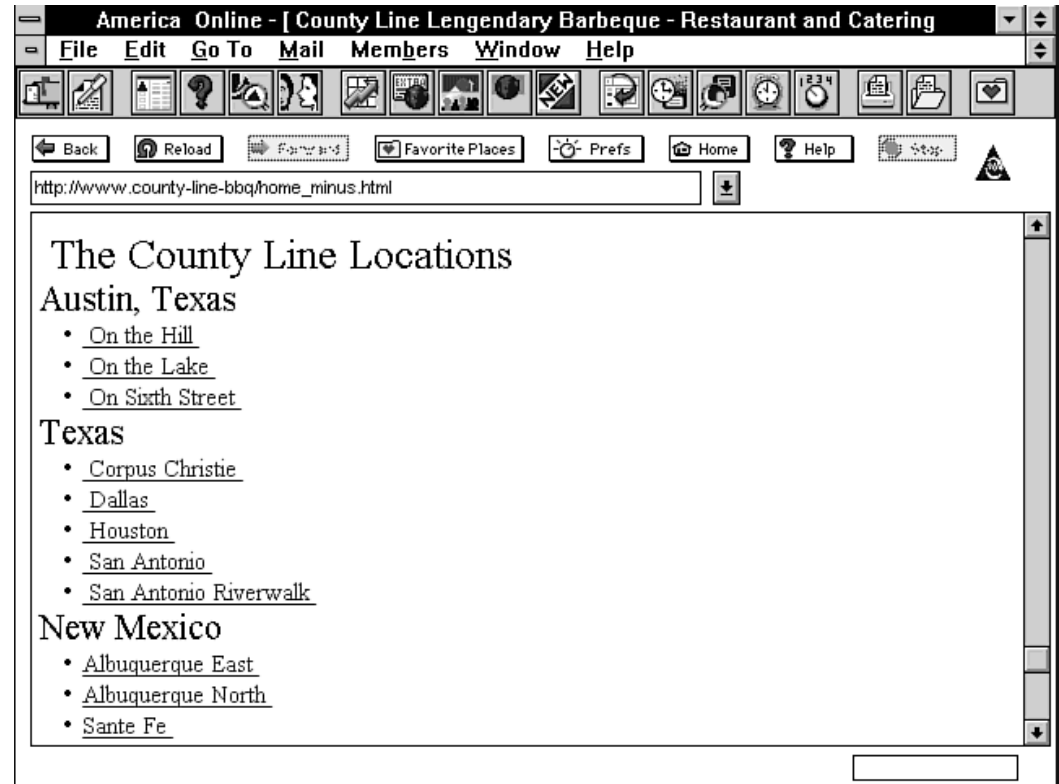


„HTML [ist] die Sprache, die die Struktur und Semantik der Inhalte eines Web-Dokuments beschreibt.“

Quelle: Mozilla Developer Network

HTML: Entwicklung

v1.0



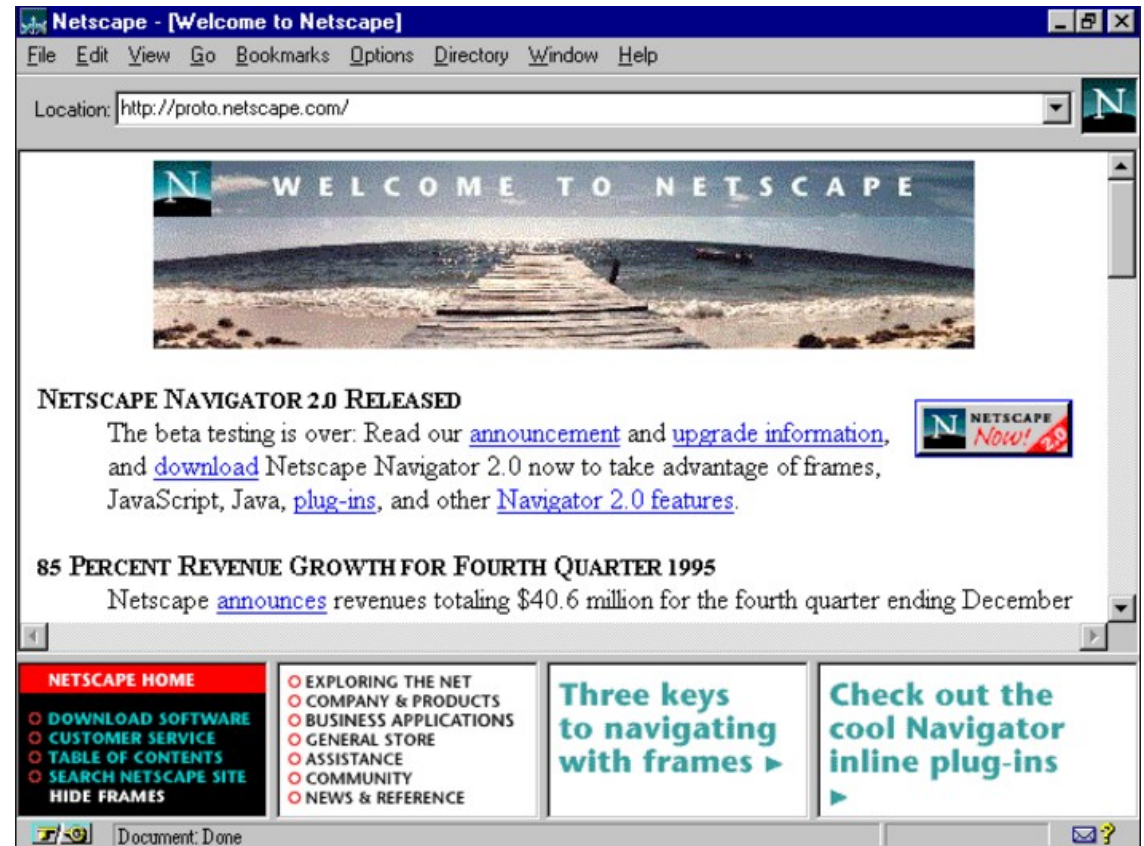
1980er: Tim Berners-Lee entwickelt HTML- und Browser-Konzept

→ 1991: 18 ersten HTML-Elemente werden vorgestellt

(wovon 11 heute noch in Gebrauch sind)

HTML: Entwicklung

v2.0



1995: Berners-Lee und andere erweitern die Sprache um zusätzliche Elemente (endlich Bilder!)

HTML: Entwicklung

1997: Standardisierung durch
mehrere Organisationen,
überwiegend Netscape und
Microsoft

v3.2

1998: Umsetzung vieler Ideen aus
v3.0
de facto Standard, teilweise noch heute

v4.01

Standardisierung noch im Gange,
dennoch bereits von allen
aktuellen Browsern unterstützt

v5

Übung: HTML-Textelemente

Hello World!

Dieser Text ist Teil eines Absatzes.

Er hat auch Zeilenumbrüche und Sonderzeichen, wie z.B. die Umlaute Ü, ö und ß.

Nutzen Sie die Dokumentation auf <https://www.w3schools.com/html/>, um folgende Aufgaben zu lösen:

Fügen Sie Ihrer Seite folgende Elemente hinzu:

1. einige **fett geschriebene Elemente**
2. einige *kursiv geschriebene Elemente*
3. eine Kombination aus **fett und kursiv**
4. eine unsortierte Auflistung

Beachten Sie: Nutzen Sie zur Lösung ausschließlich HTML-Tags und keine CSS-Eigenschaften.

Falls Ihnen Inhalte fehlen: <http://www.loremipsum.de>

HTML-Element

besteht aus Tag-Paar:

<	Beginn eines Starttags
a	Elementname (a wie anchor)
href="..."	Attributliste
>	Ende des Starttags

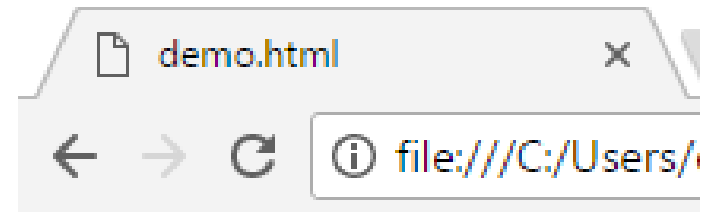
Inhalt des HTML-Element

</	Beginn des Endetags
a>	Elementname + schließendes Endetag

HTML-Element

besteht aus Tag-Paar, Attributen und Inhalt:

```
<a href="http://imgtfy.com" target="_blank">  
    Ein Hyperlink  
</a>
```

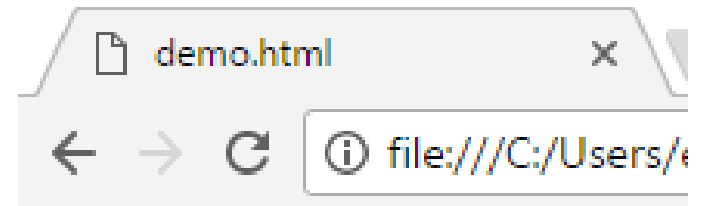


[Ein Hyperlink](http://imgtfy.com)

HTML-Element

Tags verschachteln

```
<p>  
  <a href="http://lmgtfy.com">  
    Ein  
      <strong>  
        Fetter  
      </strong>  
    Hyperlink  
  </a>  
</p>
```



[Ein Fetter Hyperlink](http://lmgtfy.com)

HTML-Elemente

Sonderfall: Empty Tags

dürfen keine Kindelemente haben, d.h. ihr Inhalt ist immer leer

Tag	Bedeutung	Verwendung
br	Break	Einfacher Zeilenumbruch
hr	Horizontal reference	Abtrennung durch horizontale Linie
img	Image	Anzuzeigendes Grafikelement
input	Input	Eingabefeld (Wert wird in Attribut <i>value</i> gehalten)

HTML-Absätze

Tag	Bedeutung	Verwendung
p	Paragraph	Absatz
br	Break	Zeilenumbruch
pre	Preformatted Text	Fixed Width-Font Leerschritte und Zeilenumbrüche werden aus Quelltext übernommen

Sonderfälle: (Inhaltliche) Zusammenfassen von Bereichen, jedoch mit dem Ziel diese ansprechen zu können:

Tag	Bedeutung	Verwendung
div	Division	Block, der inhaltlich oder optisch zusammenhängt.
span	Span	Markieren von inline-Elementen

HTML-Textelemente

Tag	Bedeutung	Verwendung
b oder strong	Bold oder Strong	Fettschrift
i	Italic	<i>Kursiv</i>
del	Deleted	Durchgestrich
sub und sup	Subscript / Superscript	Text _{Tiefgestellt} und Text ^{Hochgestellt}
h1 ... h6	Heading 1 ... 6	Überschriften, 1 = größte Stufe

HTML-Kommentare

```
<!-- Das ist ein Kommentar  
Auch über mehrere Zeilen -->  
<h1>Überschrift</h1>  
<!-- Noch mehr Kommentare -->
```

Achtung: Kommentare sind für jeden
im Quellcode lesbar!

HTML-Links

Externer Link:

```
<a href="http://google.de" target="_blank">  
    Ein Link  
</a>
```

Interner Link:

```
<a href="#textMarke1">  
    Springe zu 1  
</a>  
[...]  
<p id="textMarke1">  
    [...]  
</p>
```

HTML-Tabellen

```
<table>
  <thead>
    <tr>
      <th>Kopfzelle</th>
      <th>Noch eine</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Inhalt 1.A</td>
      <td>Inhalt 1.B</td>
    </tr>
    [...]
  </tbody>
</table>
```

Kopfzelle	Noch eine
Inhalt 1.A	Inhalt 1.B
Inhalt 2.A	Inhalt 2.B
Inhalt 3.A	Inhalt 3.B

HTML: Grafiken einbinden

```

```

Übung: HTML-Elemente 2

Erstellen Sie eine Webseite mit einigen relevanten HTML-Tags.

Folgende Elemente sollten enthalten sein:

- Inhaltsverzeichnis, das auf Abschnitte verweist
- Tabellen
- Codebeispiele
- Grafiken
- Saubere Formatierung des Quellcodes

Die wichtigsten HTML-Tags

Inhaltsverzeichnis

- [Absätze](#)
- [Textformatierung](#)
- [Kommentare](#)
- [Hyperlinks](#)
- [Grafiken einbinden](#)

Absätze

Folgende Tags bietet HTML zur Strukturierung von Texten zur Verfügung:

Absätze		
Tag	Bedeutung	Verwendung
p	<i>Paragraph</i>	Absatz
br	<i>Break</i>	Zeilenumbruch
pre	<i>Preformatted Text</i>	Fixed Width-Font Leerschritte und Zeilenumbruch werden aus Quelltext übernommen

Formatieren von Text

Folgende Tags bietet HTML zur Textformatierung zur Verfügung:

Textformatierung		
Tag	Bedeutung	Verwendung
strong	<i>Strong</i>	Verstärkte Darstellung
i	<i>Italic</i>	Kursivschrift