

Content Management Systems

Robin Clemens

Stand August 2018

© FOM Hochschule für Oekonomie und Management gemeinnützige Gesellschaft mbH (FOM), Leimkugelstraße 6, 45141 Essen

Dieses Werk ist urheberrechtlich geschützt und nur für den persönlichen Gebrauch im Rahmen der Veranstaltungen der FOM bestimmt.

Die durch die Urheberschaft begründeten Rechte (u.a. Vervielfältigung, Verbreitung, Übersetzung, Nachdruck) bleiben dem Urheber vorbehalten.

Das Werk oder Teile daraus dürfen nicht ohne schriftliche Genehmigung der FOM reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Konzeption und Umsetzung: Prof. Dr. Jörg Muschiol

Mit freundlicher Unterstützung von
Robin Clemens

Rückfragen und Anregungen: muschiol@fom.de



Kurzvita

Founder & Geschäftsführer bitloft GmbH:

Gründung Januar 2013

Co-Founder Matchbird GmbH:

Gründung September 2013

Bachelor of Science:

Abschluss März 2013 an der FOM Hochschule

Thema der Bachelor Thesis:

„Konzeption von intelligenten Systemen mit Evaluierung der User Experience und Usability innerhalb einer Webanwendung“

An der Hochschule seit 2013

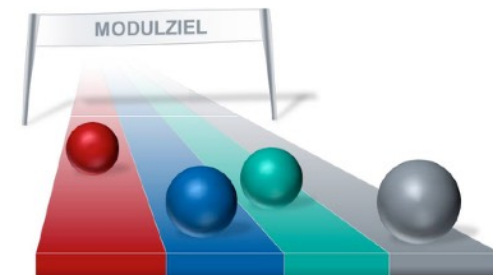
Multimedia und Design

Content-Management-Systeme

E-Business und Mobile Computing

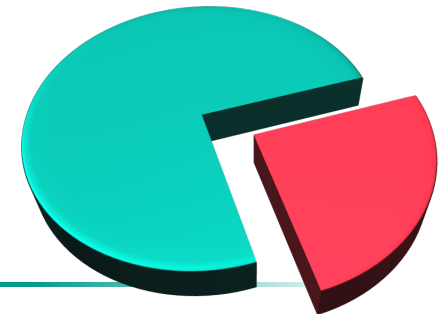
Die Studierenden können nach erfolgreichem Abschluss des Moduls:

- Die grundlegenden und erweiterten Funktionsweisen von CMS beschreiben und erklären, die Vor- und Nachteile der unterschiedlichen Plattformen und Systeme nennen, beschreiben und beurteilen.
- Verschiedene CM-Systeme bedienen und in Teilen technisch modifizieren und optimieren.
- ECMS Systeme beschreiben und für den betrieblichen Einsatz konzeptionieren.
- Einen Überblick über die grundlegenden Möglichkeiten der Programmierung von Webanwendungen und deren Architekturen geben.
- Sie verstehen die clientseitigen Basissprachen HTML, CSS und JavaScript und können Programmcode erklären.
- Die Funktionsweise von serverseitigen Programmiersprachen können sie anhand von Programmierbeispiele in den serverseitigen Scriptsprachen z.B. PHP oder JSP oder ASP.Net verstehen und erläutern.
- Ebenso können sie die jeweiligen Vor- und Nachteile der einzelnen Technologien und Plattformen für Webprojekten erklären und bewerten.



Die Modulnote setzt sich folgendermaßen zusammen:

- **Klausur 90 Minuten (100% der Modulnote)**
- **Transferaufgabe in der Klausur (etwa 10% des Klausurumfangs):**



Content Management Systems

Modul-/Veranstaltungsgliederung

	Begrüßung und Einleitung
1	Grundlagen Content Management Systeme
1.1	Historie, Konzepte und Aufbau
1.2	Konzepte Webpublishing
1.3	WCMS Architekturen und Systeme
1.4	Architektur
1.4.1	Statisches Publishing
1.4.2	Dynamisches Publishing
1.4.3	Staging
1.5	Content Lifecycle
1.5.1	Creation
1.5.2	Repository
1.5.3	Delivery
2	Enterprise Content Management
2.1	Dokumentenmanagement
2.2	Collaboration
2.3	Workflowsysteme
2.4	Portale

3	Weitere Publishing Systeme
3.1	Blogs
3.2	Shopsysteme
3.3	Newsletter
4	Siteplanung
4.1	Projektablauf und Vorgehensweise
4.2	Umsetzungskriterien und Akteure
4.3	Mehrsprachigkeit
4.4	Barrierefreiheit
4.5	Workflow und Berechtigungskonzepte
4.6	RSS / Atom
4.7	Mashups
4.8	Rich Snippets
4.9	Externe Elemente (bspw. Disqus, Github, ReCaptcha)
4.10	Content Delivery Networks für Inhalte, Fonts, etc.
5	Technische Grundlage der Webtechnologien
5.1	Protokolle (FTP, DNS, HTTP)
5.2	Sichere Protokolle (HTTPS, spdy, HTTP/2)
5.3	Kommunikation

5.3.1	Client (Browser)
5.3.2	Server (Webserver)
5.3.3	Konfiguration (Webserver)
5.4	Aufbau und Funktionsweise Domains /NICs
5.5	Mehrschichtenarchitekturen für Webanwendungen
6	Aufbau und grundlegende Funktionsweise der Webprogrammierung
6.1	Klassifikationen der Sprachen und deren Interaktionen
6.1.1	Auszeichnungssprache (HTML)
6.1.2	Stylesheet-Sprache (CSS)
6.1.3	Prototyp-basierte Sprache (JavaScript)
6.1.4	Skriptsprache (PHP, Perl)
6.1.5	OO-Sprache (Java, PHP, JS)
6.2	HTML
6.2.1	HTML-Syntax
6.2.2	Versionshistorie
6.2.3	Doctype, Aufbau
6.2.4	Versionen
6.2.5	HTML5
6.2.6	Shadow DOM

6.3	CSS
6.3.1	Selektoren
6.3.2	Attribute
6.3.3	BoxModell
6.3.4	CSS3
6.3.5	Animationen, Retuschen, etc.
6.4	JavaScript
6.4.1	Sprachgrundlagen
6.4.2	Eventhandling
6.4.3	Document Object Models (DOM)
6.4.4	Sicherheit (Client- und Serverseitige)
6.4.5	Frameworks
6.4.6	JSON / JSONP
6.5	XML
6.5.1	Anwendungsbeispiele
6.5.2	Physikalischer und logischer Aufbau
6.5.3	Parser
6.5.4	Schemata und Validierung

6.5.5	Web Service Basisarchitektur
6.5.6	Web Service Description Language (WSDL)
6.5.7	Simple Object Access Protocol (SOAP)
6.5.8	XML-RPC
7	Praktische Demonstration einer webbasierten Programmiersprache z.B. PHP
7.1	Laufzeitumgebungen
7.2	Allgemeine Syntax
7.3	Ausgabe
7.4	Skriptsprachenorientierte Programmierung
7.4.1	Typen und Datenstrukturen
7.4.2	Operatoren
7.4.3	Kontrollstrukturen
7.4.4	Funktionen
7.5	Request verarbeiten
8	Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress
8.1	Einrichtung von Wordpress
8.2	Backend und Frontend
8.3	Erstellen von Seiten, Blogbeiträgen und Inhalten

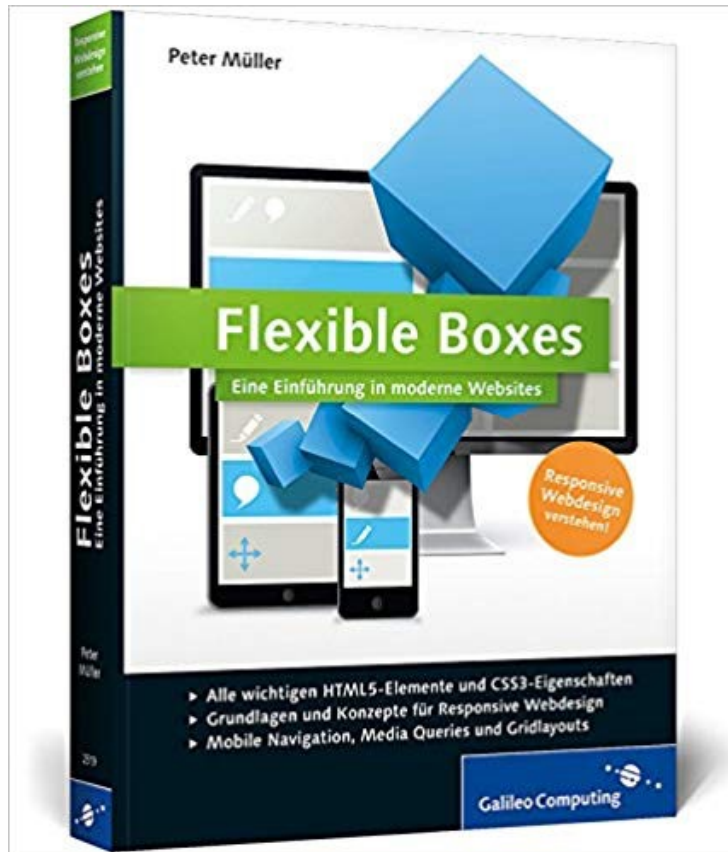
8.4	Verwalten von Assets
8.5	Themen Anpassungen
8.6	Kontaktformulare
8.7	Erstellen und Einbindung von HTML Elemente mit CSS
8.8	Plugin Funktionen



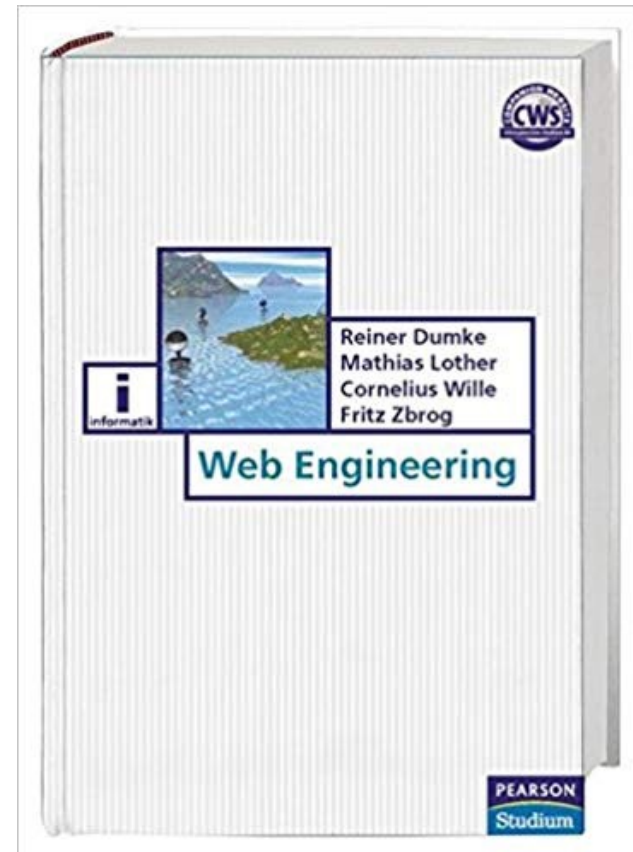
Peter Müller
Little Boxes, Teil 1. Webseiten gestalten mit CSS
ISBN: 978-3827247148 | 2011



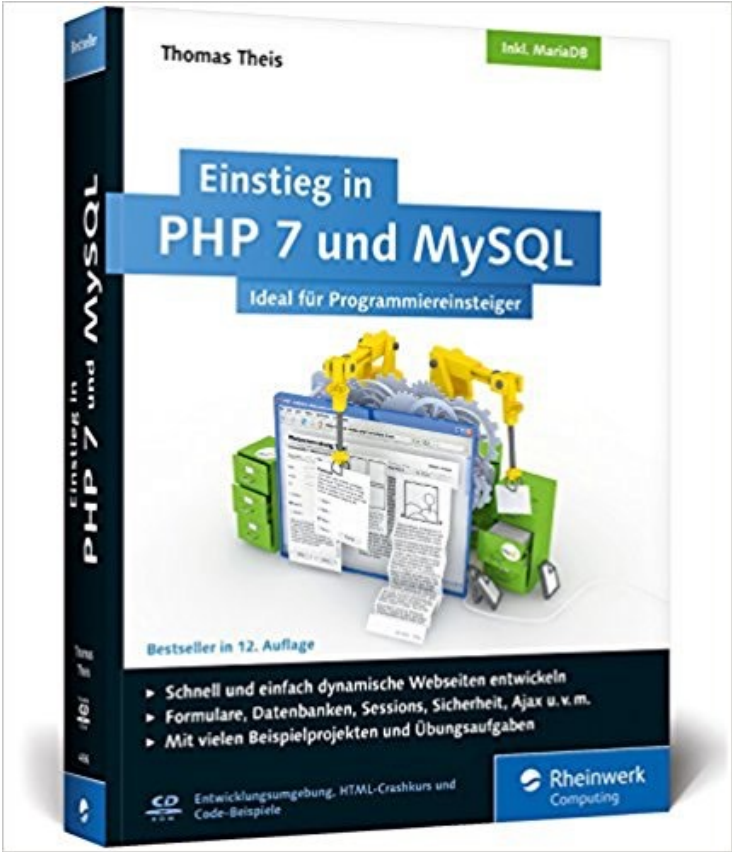
Brian P. Hogan
HTML5 & CSS3: Webentwicklung mit den Standards von morgen
ISBN: 978-3897213166 | 2011



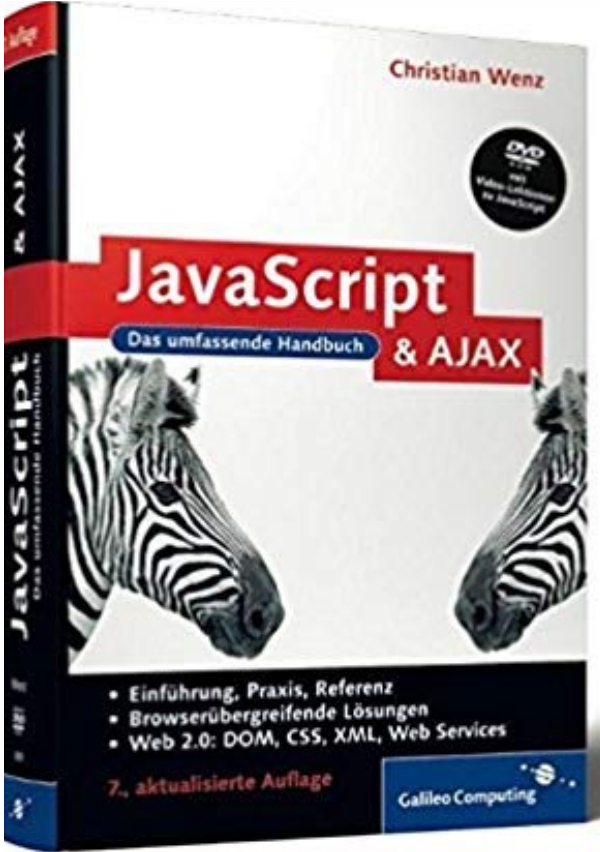
Peter Müller
Flexible Boxes: Eine Einführung in moderne Websites.
Responsive Webdesign verstehen
ISBN: 978-3836225199 | 2013



Mathias Lothar, Cornelius Wille, Fritz Zbrog, Reiner Dumke
Web Engineering
ISBN: 978-3827370808 | 2003



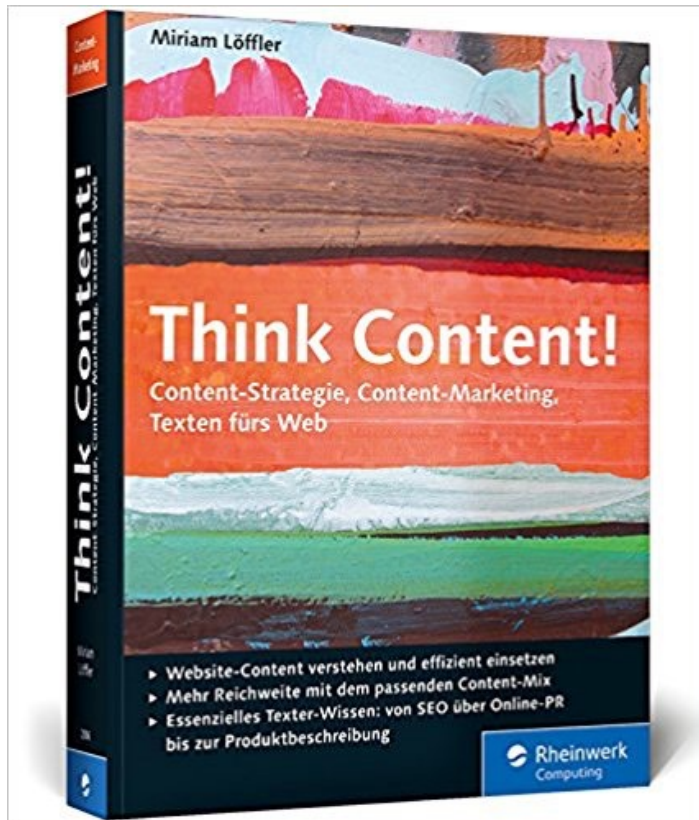
Thomas Theis
Einstieg in PHP 7 und MySQL: Für Programmieranfänger geeignet. So programmieren Sie dynamische Websites mit PHP und MySQL. Inkl. MariaDB Taschenbuch
ISBN: 978-3836244961 | 2017



Christian Wenz
JavaScript und AJAX: Das umfassende Handbuch
ISBN: 978-3898428590 | 2006



Jon Duckett
HTML and CSS: Design and Build Websites
ISBN: 978-1118008188 | 2011



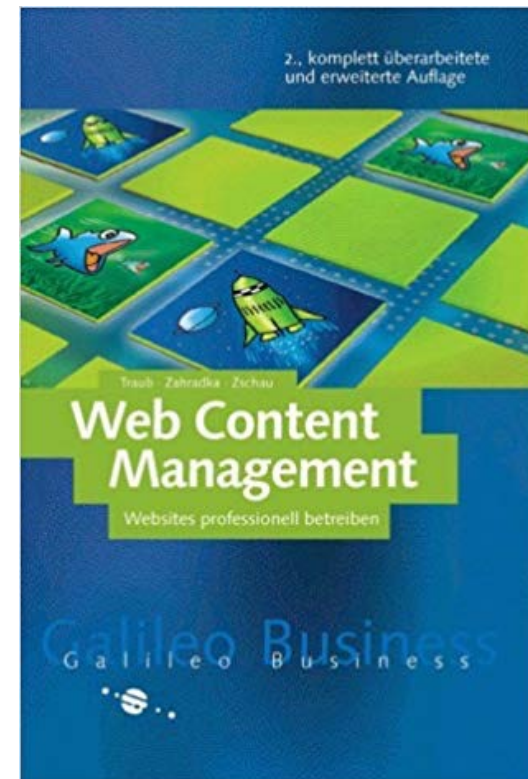
Miriam Löffler
Think Content!: Content-Strategie, Content-Marketing,
Texten fürs Web
ISBN: 978-3836220064 | 2014



Wilfred Lindo
Newsletter-Marketing - Das Praxisbuch
ISBN: 978-3772376689 | 2009



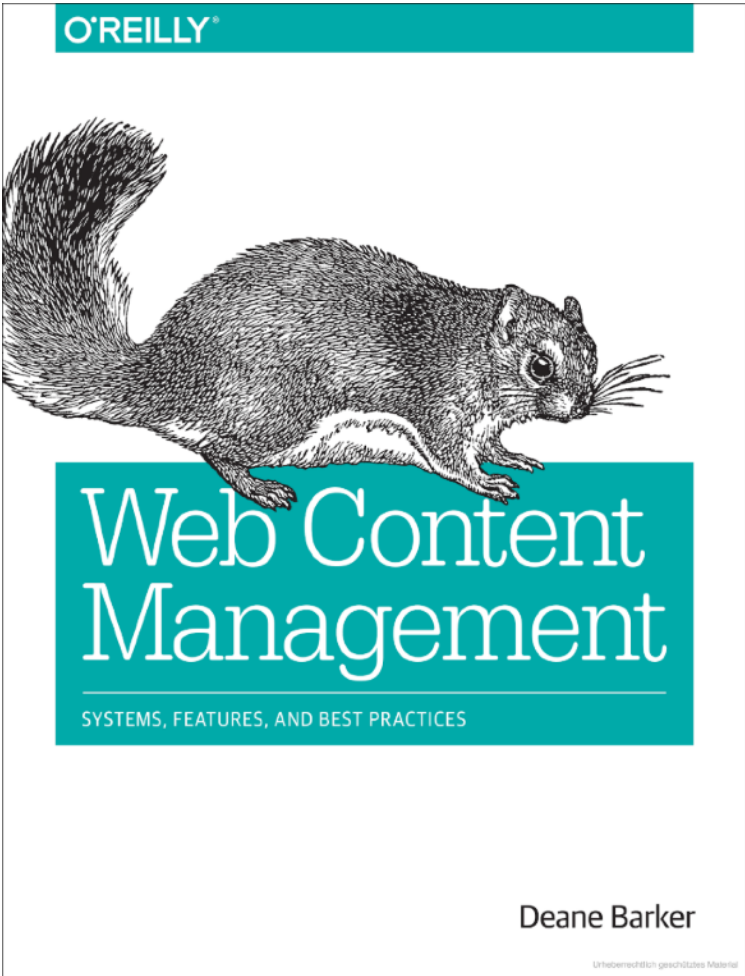
Tim Ash
Landing Pages: Optimieren, Testen, Conversions generieren
ISBN: 978-3826655142 | 2009



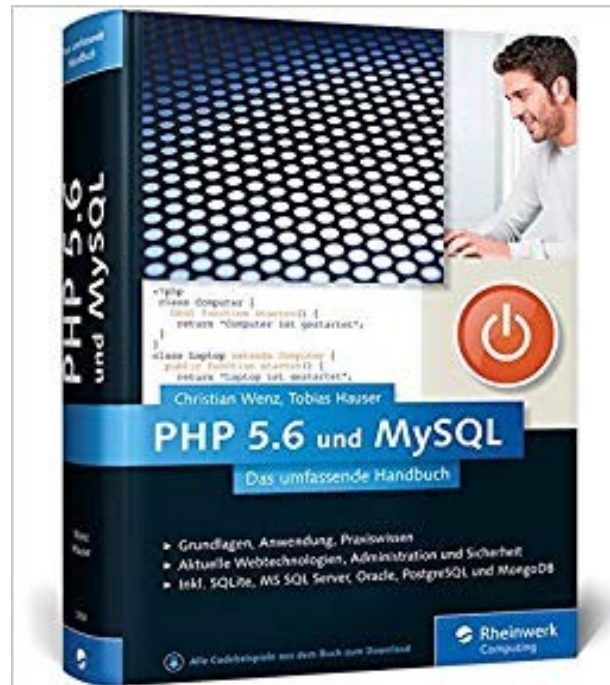
Oliver Zschau, Dennis Traub, Rik Zahradka
Web Content Management - Websites professionell planen und betreiben
ISBN: 978-3898421577 | 2001



Watrall, E. / Siarto, J.:
Webdesign von Kopf bis Fuß.
2009



Deane Barker
Web Content Management
Systems, Features, and Best Practices.
2016



Christian Wenz (Autor), Tobias Hauser (Autor)
PHP 5.6 und MySQL Von den Grundlagen
bis zur professionellen Programmierung PHP 5 / MySQL 5
ISBN: 978-3836230582 | 2015

6 Aufbau und grundlegende Funktionsweise der Webprogrammierung

Aufbau und grundlegende Funktionsweise der Webprogrammierung

Modul-/Veranstaltungsgliederung

5.3.1	Client (Browser)
5.3.2	Server (Webserver)
5.3.3	Konfiguration (Webserver)
5.4	Aufbau und Funktionsweise Domains /NICs
5.5	Mehrschichtenarchitekturen für Webanwendungen
6	Aufbau und grundlegende Funktionsweise der Webprogrammierung
6.1	Klassifikationen der Sprachen und deren Interaktionen
6.1.1	Auszeichnungssprache (HTML)
6.1.2	Stylesheet-Sprache (CSS)
6.1.3	Prototyp-basierte Sprache (JavaScript)
6.1.4	Skriptsprache (PHP, Perl)
6.1.5	OO-Sprache (Java, PHP, JS)
6.2	HTML
6.2.1	HTML-Syntax
6.2.2	Versionshistorie
6.2.3	Doctype, Aufbau
6.2.4	Versionen
6.2.5	HTML5
6.2.6	Shadow DOM

Aufbau und grundlegende Funktionsweise der Webprogrammierung

Modul-/Veranstaltungsgliederung

6.3	CSS
6.3.1	Selektoren
6.3.2	Attribute
6.3.3	BoxModell
6.3.4	CSS3
6.3.5	Animationen, Retuschen, etc.
6.4	JavaScript
6.4.1	Sprachgrundlagen
6.4.2	Eventhandling
6.4.3	Document Object Models (DOM)
6.4.4	Sicherheit (Client- und Serverseitige)
6.4.5	Frameworks
6.4.6	JSON / JSONP
6.5	XML
6.5.1	Anwendungsbeispiele
6.5.2	Physikalischer und logischer Aufbau
6.5.3	Parser
6.5.4	Schemata und Validierung

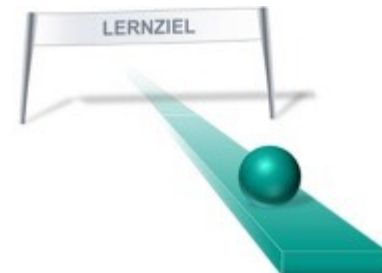
Aufbau und grundlegende Funktionsweise der Webprogrammierung

Modul-/Veranstaltungsgliederung

6.5.5	Web Service Basisarchitektur
6.5.6	Web Service Description Language (WSDL)
6.5.7	Simple Object Access Protocol (SOAP)
6.5.8	XML-RPC
7	Praktische Demonstration einer webbasierten Programmiersprache z.B. PHP
7.1	Laufzeitumgebungen
7.2	Allgemeine Syntax
7.3	Ausgabe
7.4	Skriptsprachenorientierte Programmierung
7.4.1	Typen und Datenstrukturen
7.4.2	Operatoren
7.4.3	Kontrollstrukturen
7.4.4	Funktionen
7.5	Request verarbeiten
8	Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress
8.1	Einrichtung von Wordpress
8.2	Backend und Frontend
8.3	Erstellen von Seiten, Blogbeiträgen und Inhalten

Im Anschluss an diesen Themenblock sollen Sie:

- Sie verstehen die clientseitigen Basissprachen HTML, CSS und JavaScript und können Programmcode erklären.
- Die Funktionsweise von serverseitigen Programmiersprachen können sie anhand von Programmierbeispiele in den serverseitigen Skriptsprachen z.B. PHP oder JSP oder ASP.Net verstehen und erläutern.
- Die notwendigen Bestandteile eines Content-Management-Systems in der Praxis definieren und beurteilen, sowie dafür notwendige Komponenten als Template erstellen
- Ebenso können sie die jeweiligen Vor- und Nachteile der einzelnen Technologien und Plattformen für Webprojekten erklären und bewerten.



Klassifikationen der Sprachen und deren Interaktionen

Auszeichnungssprache	Stylesheetsprache	Prototypbasierte Sprache	Skriptsprache	Objektorientierte Sprache
HTML	CSS	JavaScript	PHP	Java
XHTML	Sass	ActionScript	Perl	PHP
HTML 5	Less			ASP.NET
LaTeX	XSL			ObjectiveC

Aufbau und grundlegende Funktionsweise der Webprogrammierung

Auszeichnungssprache (HTML)

- Auszeichnungssprache beinhaltet Beschreibungs- und Nutzdaten
- Dateien die mit einer Auszeichnungssprache aufgebaut werden, können von einer Maschine bzw. einem Computer verarbeitet werden
- Beschreibungsdaten
 - Dienen dazu dass eine Datenstruktur vorliegt
 - Bestehen aus Bezeichner und Attributen (ergeben sogenannte Tags)
 - Mehrere Tags können verschachtelt aufgebaut werden
 - Eltern-Kind-Struktur
 - Vorgänger-Nachfolger-Struktur
 - Sind anhand einer definierten Logik aufgebaut
 - Logik kann in entsprechender Definition (Regelwerk oder Datei) vorgegeben werden
- Nutzdaten
 - Entsprechen den eigentlichen Werten/Inhalten
 - Können z.B. aus einer Datenbank entnommen werden
- Die Definition von z.B. HTML oder XML entsteht durch W3C
- Stellen in ihrem Bereich das Grundgerüst einer Datei dar
- Wird als Text in eine "Textdatei" geschrieben
- Wird auf dem Client zur Laufzeit ausgeführt

Beispiel:

```
<products>
  <article>
    <articlenumber length="4">1234</articlenumber>
    <matchcode>Playstation 4 128GB</matchcode>
    <price>129.99</price>
  </article>
  <article>
    <articlenumber length="4">9987</articlenumber>
    <matchcode>Apple MacBook Pro 15"</matchcode>
    <price>1299.99</price>
  </article>
</products>
```


Beispiel HTML:

```
<html>
  <head>
  </head>
  <body>
    <header>
      <p>Herzlich Willkommen auf unserer Webseite</p>
    </header>
    <footer>
      <p>Copyright 2018 Muster GmbH</p>
    </footer>
  </body>
</html>
```

- Stylesheet-Sprachen dienen dazu das Erscheinungsbild zu beeinflussen
- Stylesheet-Sprachen benötigen einen Grundrahmen bzw. Beschreibungs- und Nutzdaten um genutzt zu werden
- CSS funktioniert z.B. nicht ohne HTML
- Werden verwendet um Vorlagen/Templates/Layouts digital abzubilden
- Jedes Element auf einer modernen Webseite ist z.B. per CSS definiert
- Kann Einfluss auf Position, Größe, Farbe etc. Von Beschreibungsdaten (Tags) vornehmen
- Können aufeinander aufbauen (CSS => Less) oder auch zu den Beschreibungsdaten als Tag agieren (bei z.B. XSL)
- Wird als Text in eine "Textdatei" geschrieben
- Die Definition von z.B. CSS entsteht durch W3C
- Wird auf dem Client zur Laufzeit ausgeführt

Beispiel CSS:

```
body {  
    width:300px;  
    height:200px;  
}
```

```
footer {  
    background-color:red;  
}
```

Aufbau und grundlegende Funktionsweise der Webprogrammierung

Prototyp-basierte Sprache (JavaScript)

- Prototypen Ansatz entspricht einer anderen Art der Objektorientierung
- Form der Abbildung unterscheidet sich im Vergleich zu OO (Verzicht auf Klassen)
- Objekte geklont statt als Instanz erzeugt
- Wird auf dem Client zur Laufzeit ausgeführt
- Ermöglicht die Verwendung von richtigem Programmiercode
- Enthält Schleifen, Bedingungen etc. wie bei einer klassischen Programmiersprache
- Zusammenspiel mit HTML und CSS um Elemente zu bewegen, animieren oder zu verändern
- JavaScript kann z.B. Beschreibungs- und Nutzungsdaten während der Benutzung einer Webseite verändern
- JavaScript ist eine elementare Komponente heutiger Webseiten/-anwendungen
- Wird als Text in eine "Textdatei" geschrieben

Beispiel JavaScript:

```
if (3 < 4 ) {  
    alert(„3 ist kleiner als 4“);  
} else (3 > 4) {  
    alert(“3 ist größer als 4“);  
}
```

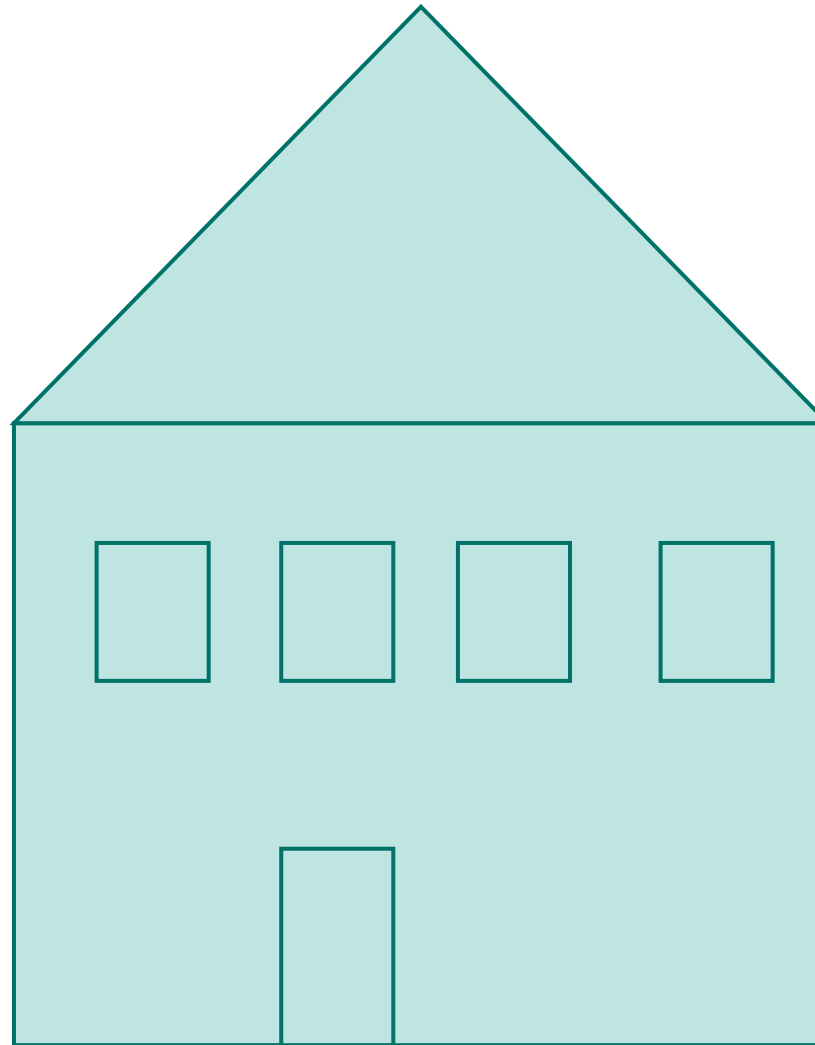
Assoziation

“Haus“

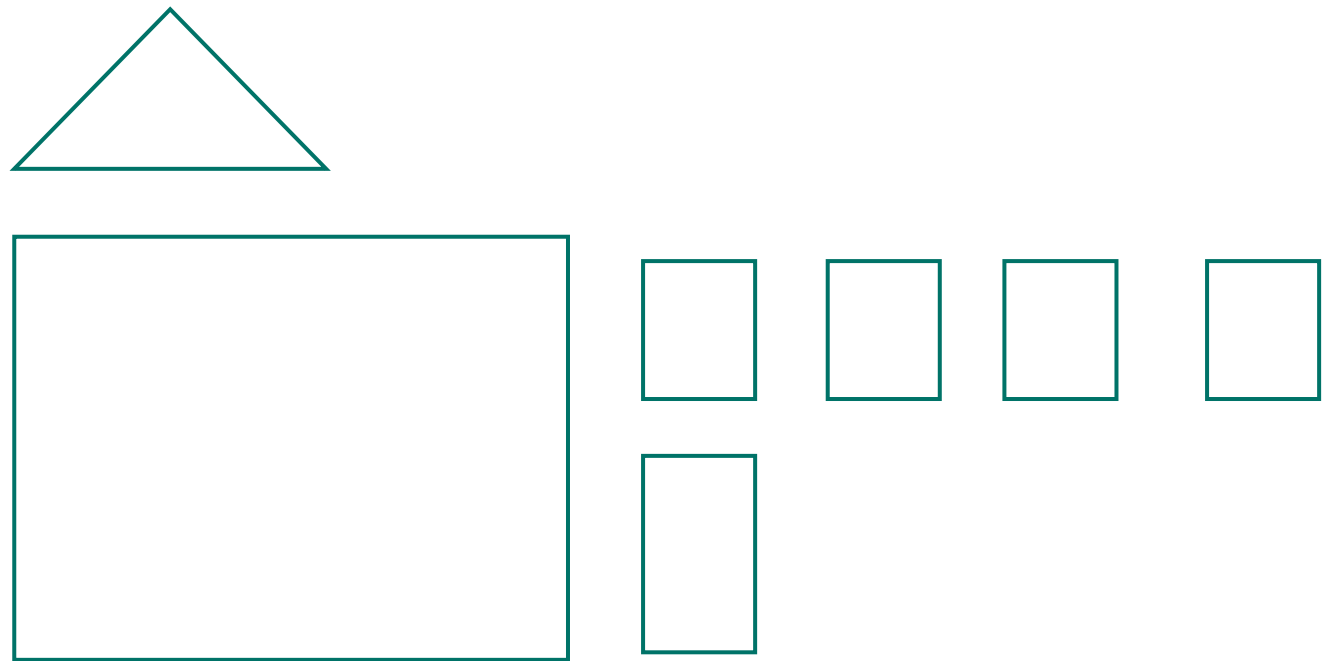
HTML?

CSS?

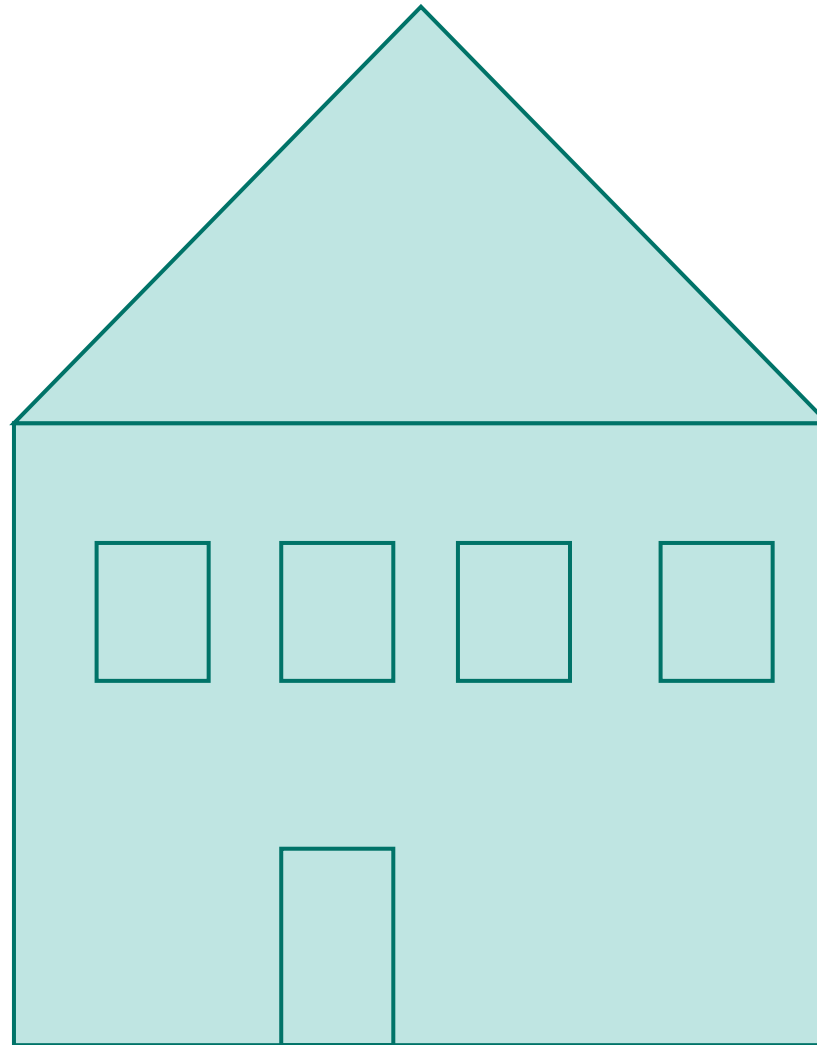
JS?



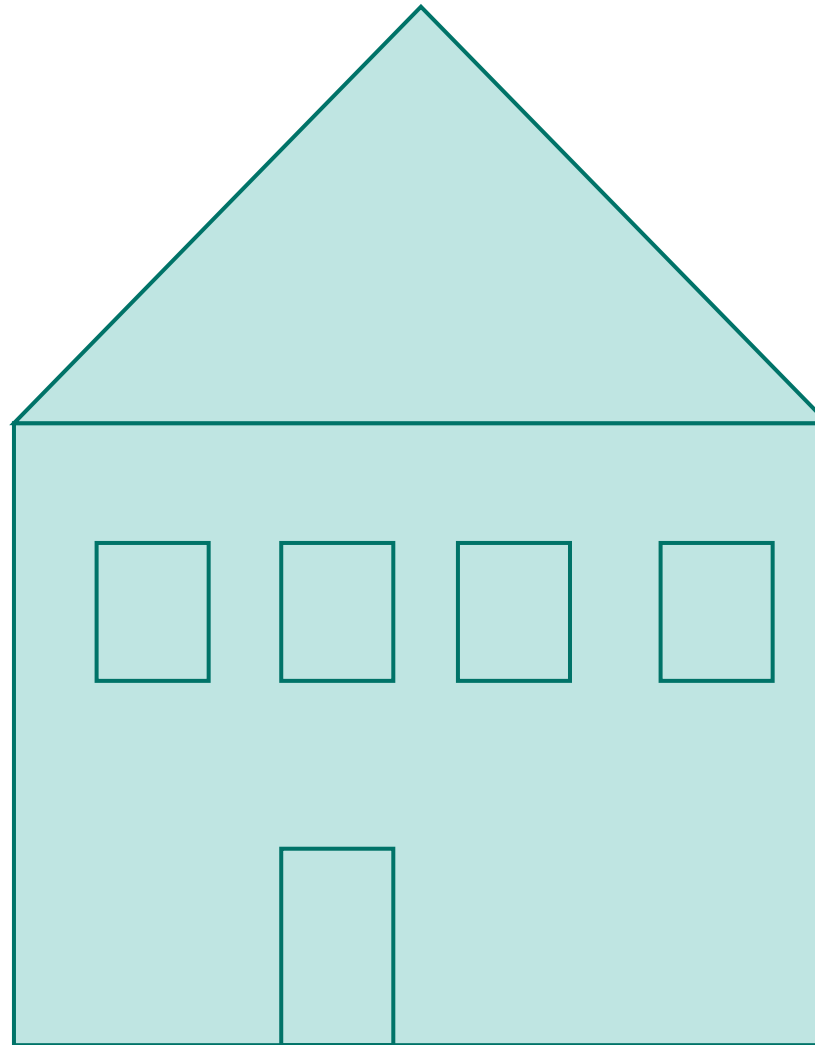
HTML



CSS



JavaScript



Quelle: amazon.com



Quelle: ces.eu

Aufbau und grundlegende Funktionsweise der Webprogrammierung

Skriptsprachen (PHP, Perl)

- Besteht aus Programmiercode der Grundbestandteile der klassischen Programmierung beinhaltet (Bedingungen, Schleifen, Funktionen etc.)
- In der Form von Skriptsprache ermöglicht er keine Logische Abbildung (Objekte, Modelle, Vererbung etc.) oder enthält keinen Deklarationszwang
- Wird auf dem Server zur Laufzeit ausgeführt
- Ein Interpreter liest den den Programmiercode, analysiert diesen und führt ihn aus
- Ermöglicht z.B. Verbindung zu Datenbank (lesen & schreiben von Daten)
- Ermöglicht z.B. Erstellen von Dateien aus Eingabe des Benutzers
- Können HTML Code an den Client ausliefern
- Auslieferung HTML z.B. je nach Uhrzeit, Datum oder Eingaben des Benutzers (Verwendung von if-Bedingungen)
- Wird häufig für die Entwicklung kleinerer Programme verwendet
- Wird als Text in eine "Textdatei" geschrieben

Beispiel PHP:

```
<?php
```

```
$silvester = „31.12.“;
```

```
function WannIstSilvester() {  
    echo „Silvester ist am $silvester“;  
}
```

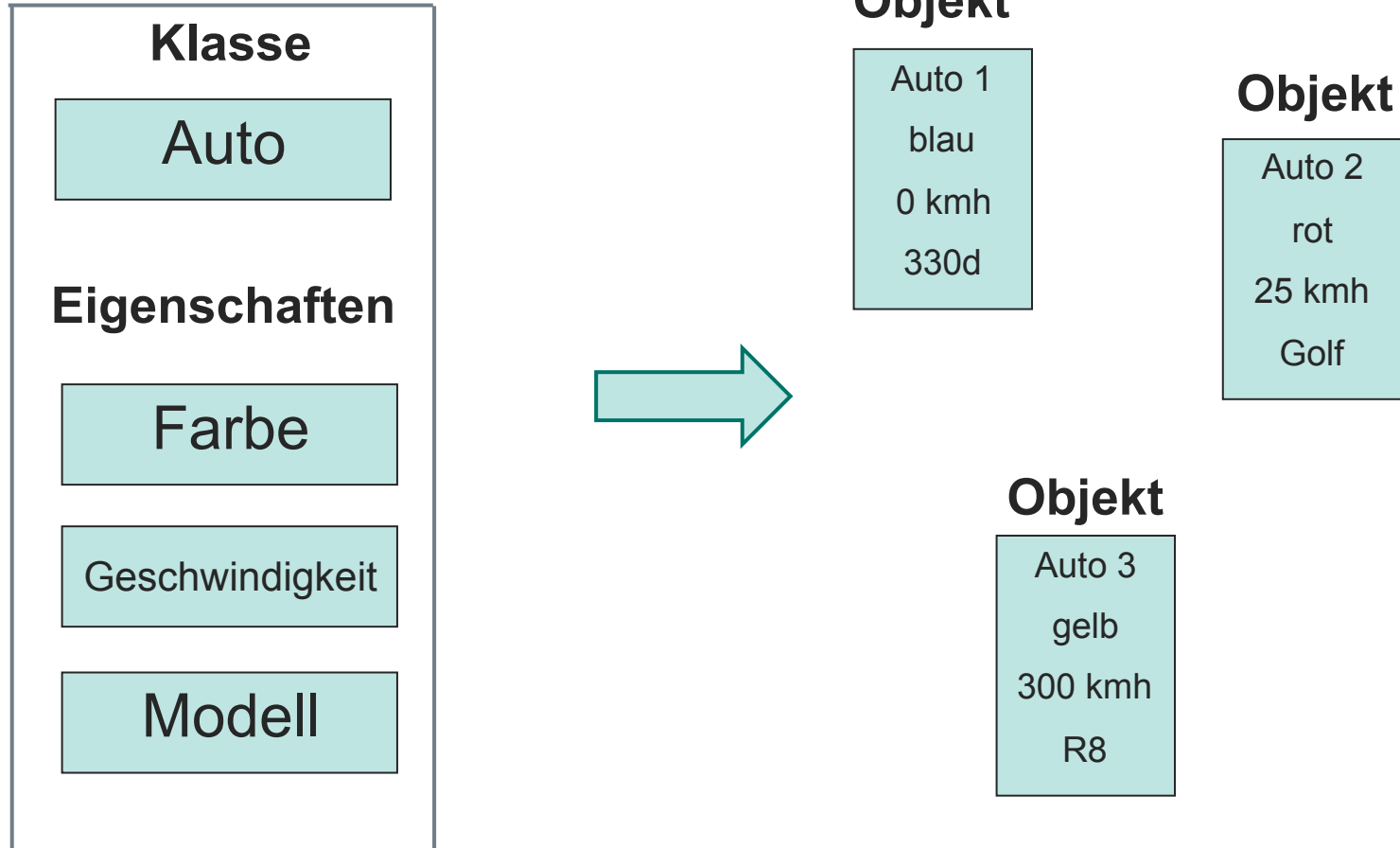
```
WannIstSilvester();
```

```
?>
```

- Ermöglicht Abbildung von
 - Objekten
 - Eigenschaften
 - Methoden
 - Vererbung
 - etc.
- Beinhaltet alle Abbildungsmöglichkeiten von Skriptsprachen
- Architektur der Software richtet sich sehr an realer Abbildung
- Architektur besteht aus wiederverwendbaren Objekten mit Eigenschaften
 - Auto => Farbe, Geschwindigkeit, Modell
 - Person => Vorname, Nachname, Geburtsdatum
- Hohe Abbildungslogik reduziert die Menge an Programmiercode

Aufbau und grundlegende Funktionsweise der Webprogrammierung

Objektorientierte Sprachen (Java, PHP, JS)



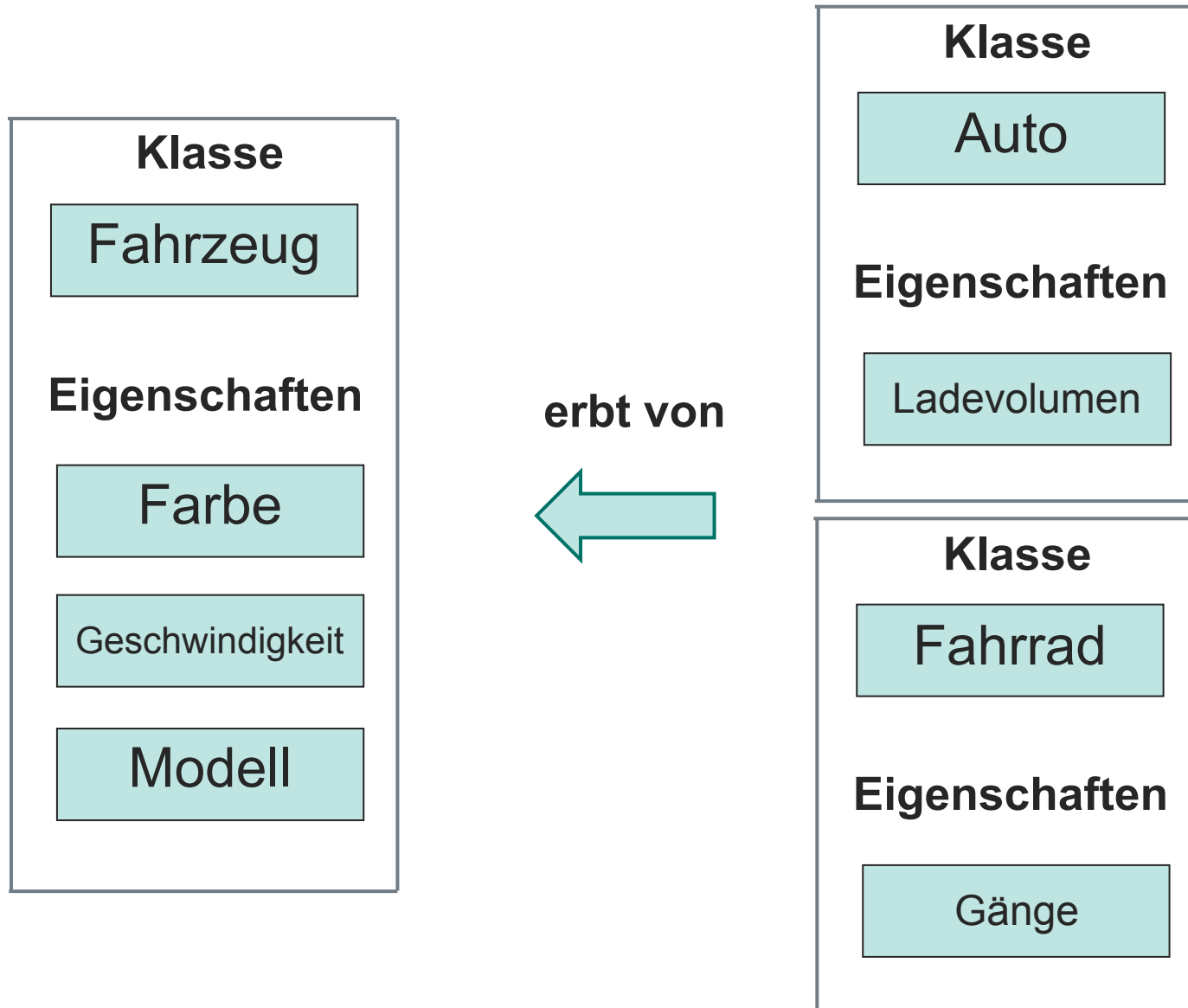
Beispiel PHP:

```
<?php
```

```
class Auto {  
    public $farbe;  
    public $geschwindigkeit;  
    public $modell;  
}
```

```
$auto = new Auto();  
$auto->farbe = "rot";  
$auto->modell = „bmw“;
```

```
?>
```



Beispiel PHP:

```
<?php
```

```
class Fahrrad extends Fahrzeug {  
    public $gaenge;  
}
```

```
$fahrrad = new Fahrrad();  
$fahrrad->gaenge = 24;  
$fahrrad->farbe = "rot";  
$fahrrad->modell = „Giant“;
```

```
?>
```



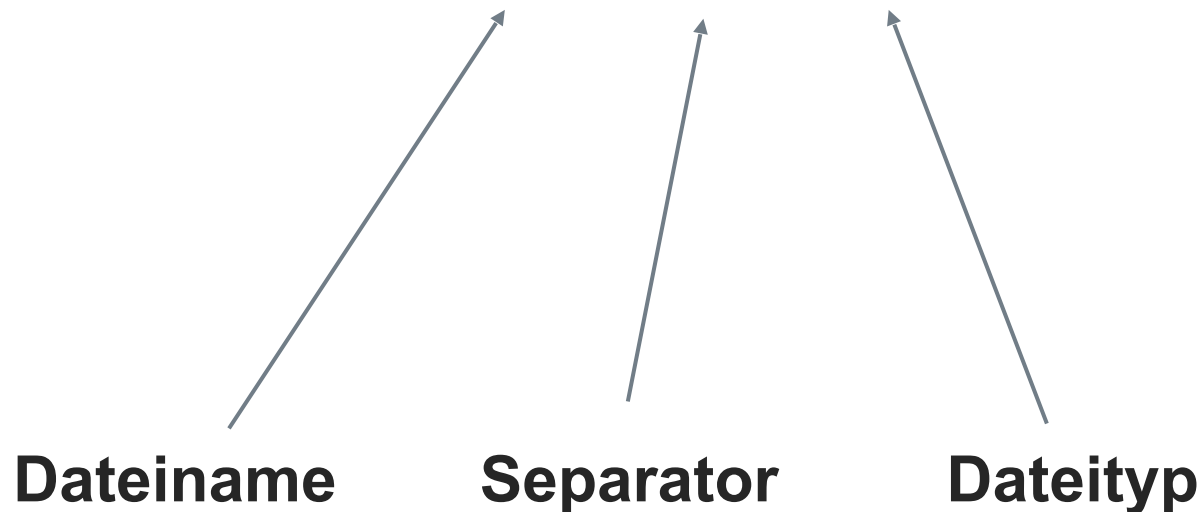

HTML

(Extensible) Hypertext Markup Language

- Internetseiten werden mit der Sprache HTML beschrieben
- HTML = Hyper Text Markup Language
- HTML ist eine „Auszeichnungssprache“, keine Programmiersprache
- Im Browser angekommen, spricht man vom DOM (**Document Object Mode**) und kann z. B. durch Javascript verändert werden

Textdokument

index.html



- Regeln für den Dateinamen einer HTML-Datei:
 - Datei-Endung immer .html oder .htm
 - Alles klein schreiben! Schlecht: ErsteSeite.html
 - keine Leerzeichen! Schlecht: erste seite.html
 - keine Sonderzeichen Schlecht: erste\$seite.html
 - keine Umlaute: ä, ü, ö Schlecht: übungsseite.html
 - kein scharfes s: „ß“ Schlecht: spaßseite.html
 - Nur ASCII nutzen Gut: ersteseite.html



index.html



- HTML wird Clientseitig ausgeführt
- Tags, bestehend aus Character
- Tags markiert durch Spitzklammer
- Tags agieren als Container
- Tags umfassen Content
- Tags können Attribute haben
- Tags (einleitende) enthalten Klassen
- Attribute erfordern einen Namen & Wert

- Welche Tags stehen zur Verfügung?
- Wofür stehen die Tags?
- Wohin gehören die Tags?

THINK FIRST!

HTML Tag Kategorien

- Text (Überschriften, Paragraphen, Absätze, Zitate etc.)
- Listen (Nummeriert, Aufzählung, Definition, Verzeichnis)
- Links (Extern, Intern, E-Mail)
- Bilder (IMG, Figure etc.)
- Tabellen
- Formulare (Input, Button, Checkbox etc.)
- Markup (Kommentare, Doctype, DIV, Span etc.)
- Multimedia (Flash, Video, Audio etc.)

Einleitender Tag

<html>



Spitzklammer

Element

Abschließender Tag

</html>



Slash

Einleitender Tag

Abschließender Tag

<h1 class=„bigger“>

</h1>

Attribut Wert

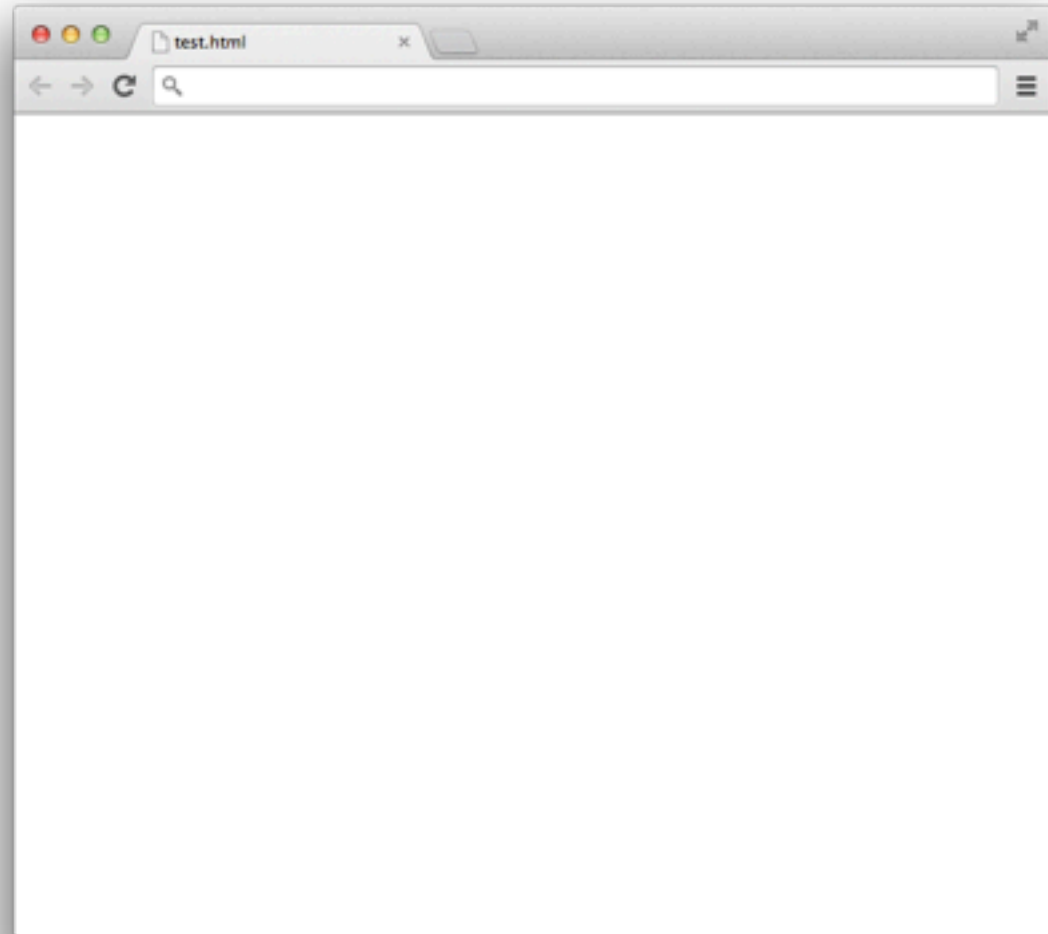
Attribut

Spitzklammer Element

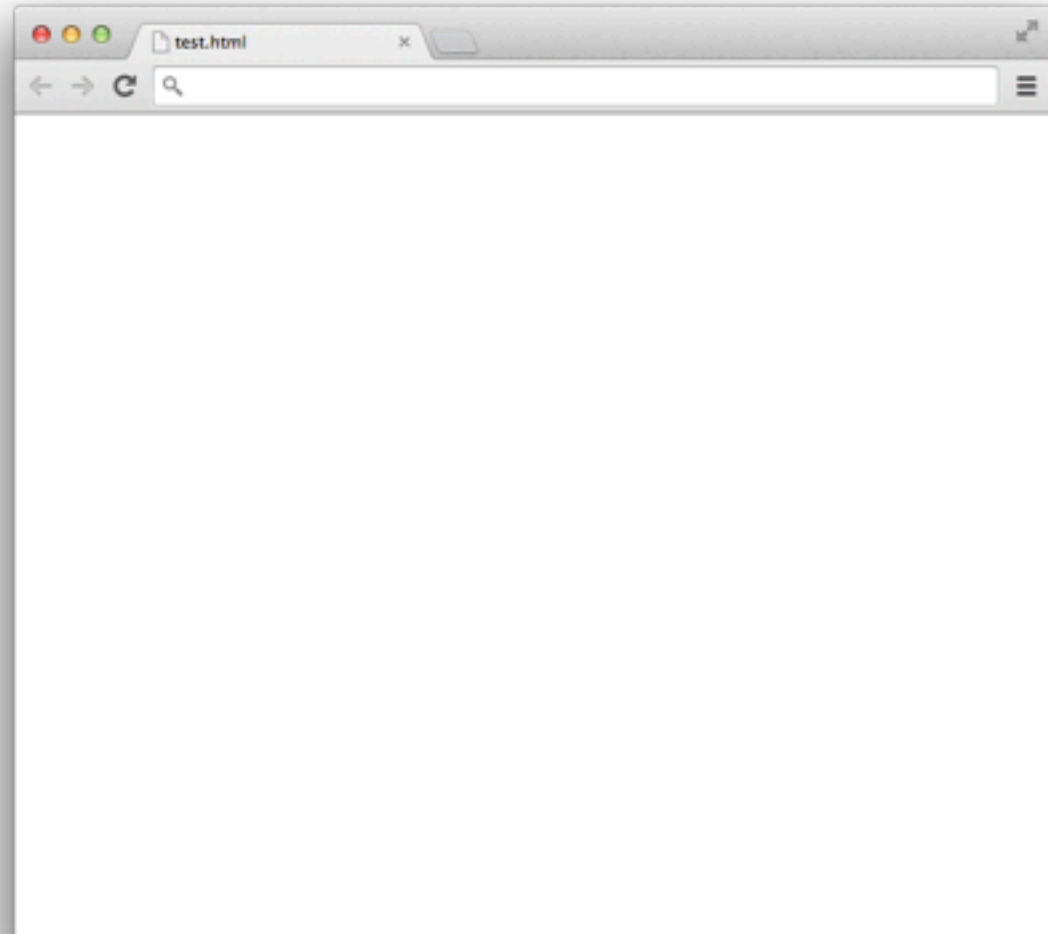
- Programme zur Darstellung von HTML-Seiten sind die „Web-Browser“ z.B.:
- Internet Explorer Trident
- Chrome, Safari Webkit
- Opera Presto
- Mozilla Firefox Gecko

- Problem: Darstellung der Browser variiert.

```
<html>  
</html>
```

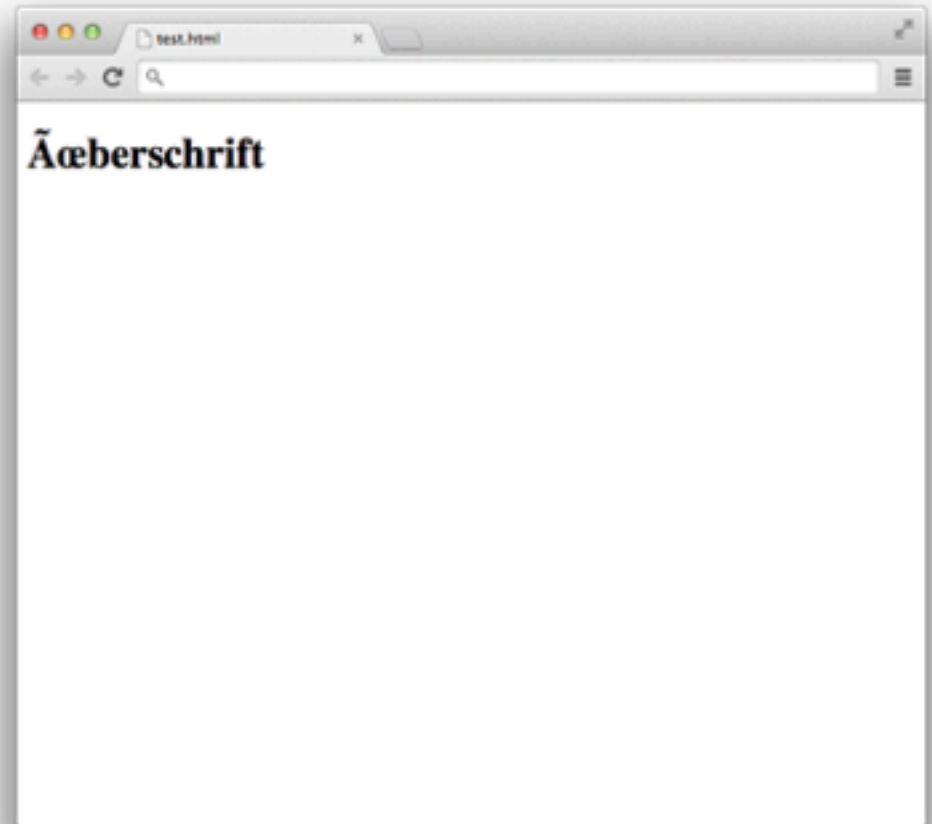


```
<html>  
  <head>  
  </head>  
  <body>  
  </body>  
</html>
```



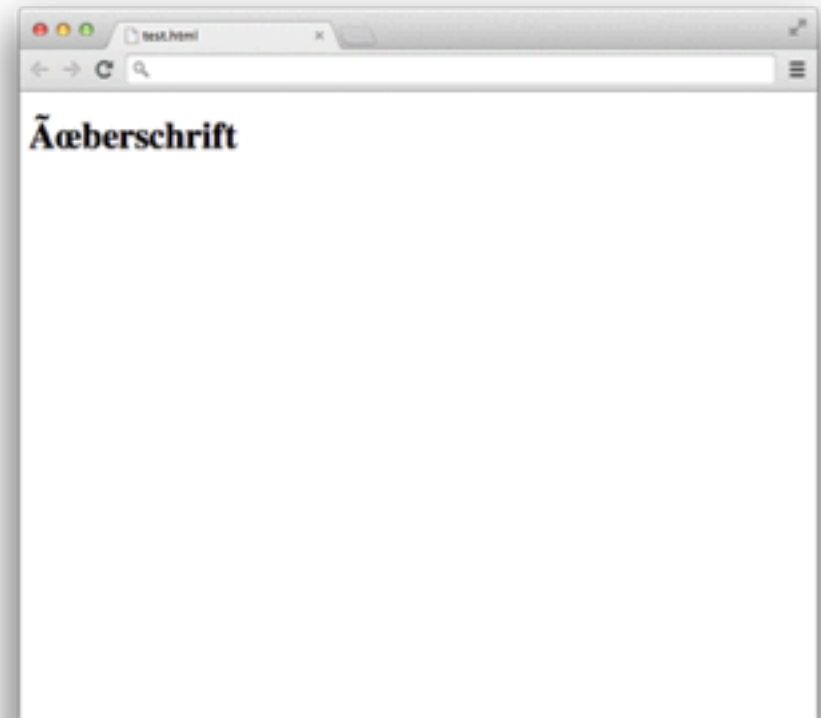
HTML Beispiel

```
<html>
  <head>
  </head>
  <body>
    <h1>Überschrift</h1>
  </body>
</html>
```



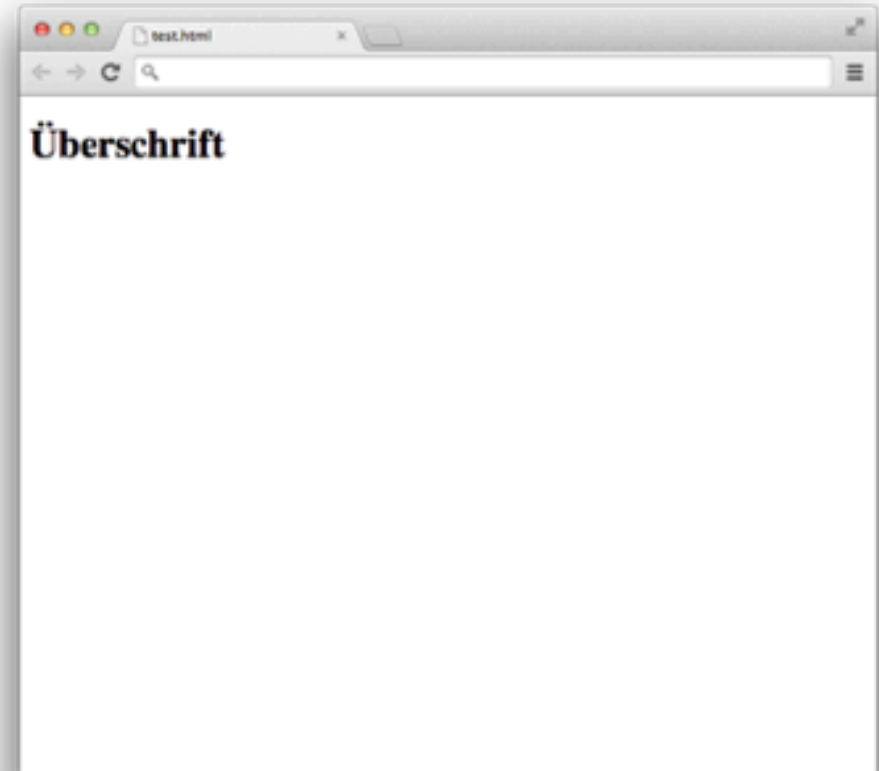
HTML Beispiel

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01 Transitional//EN" "http://  
www.w3.org/TR/html4/loose.dtd">  
<html>  
  <head>  
  </head>  
  <body>  
    <h1>Überschrift</h1>  
  </body>  
</html>
```



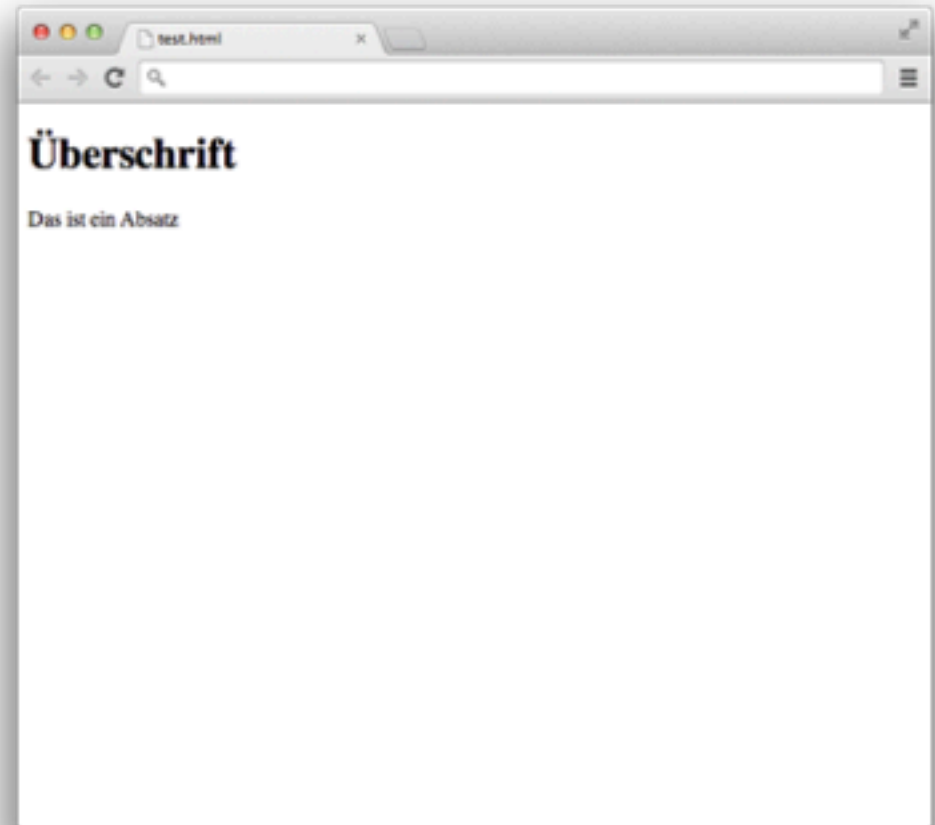
HTML Beispiel

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01 Transitional//EN" "http://  
www.w3.org/TR/html4/loose.dtd">  
<html>  
  <head>  
    <meta http-equiv="Content-Type"  
content="text/html; charset=utf-8">  
  </head>  
  <body>  
    <h1>Überschrift</h1>  
  </body>  
</html>
```



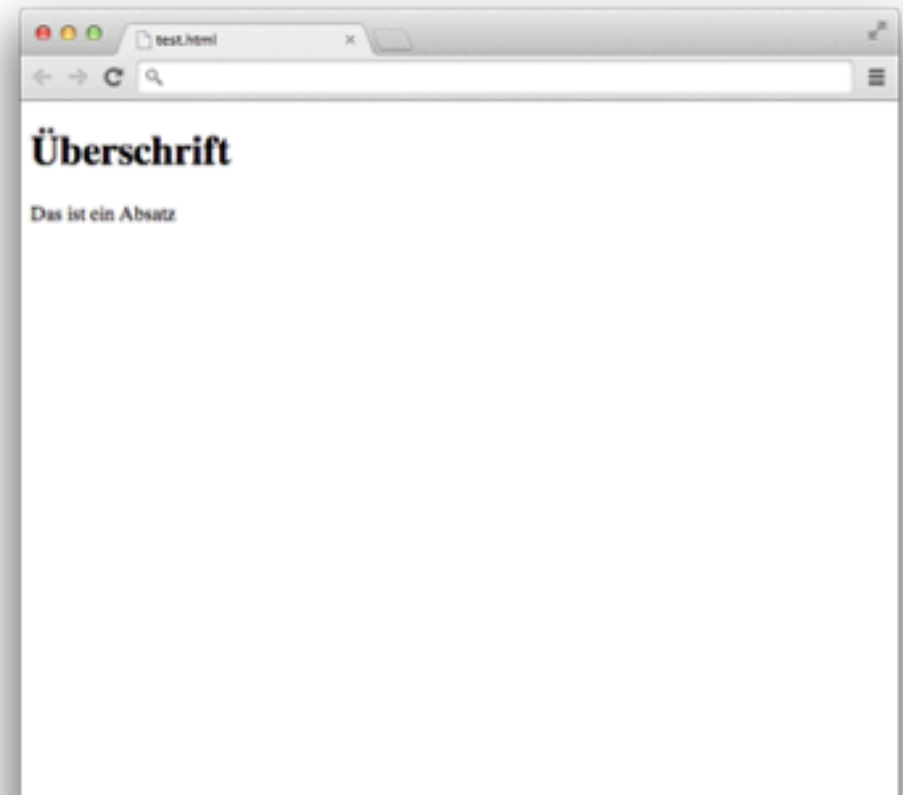
HTML Beispiel

```
...  
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    <h1>Überschrift</h1>  
    <p>Das ist ein Absatz</p>  
  </body>  
</html>
```



HTML Beispiel

```
...  
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    <div id="header"></div>  
    <div id="content">  
      <h1>Überschrift</h1>  
      <p>Das ist ein Absatz</p>  
    </div>  
    <div id="footer"></div>  
  </body>  
</html>
```



- Link auf eine externe Seite
 - `FOM`
 - `FOM`

- Link auf eine eigene, interne Seite
 - `klicken`
(Datei im gleichen Verzeichnis)

- ``
- **Achtung:**
 - Speicherort des Bildes beachten!
 - ``
- **Zusätzliche Angaben:**
 - Text, falls Bild nicht angezeigt wird (PFLICHT!):
 - + `alt="text"`
 - Text als Beschreibung:
 - + `title="Hier sieht ein Bild"`
- ``

Tabellen in HTML

Tabellen Zeilen engl. table rows (`<tr>`)

Tabellen Rand engl. border

Tabellen Überschrift engl. heading (`<th>`)

Tabellen Inhalte table data (`<td>`)

Stichwort	Bedeutung
Cookie	Benutzerdaten
Firewall	Sicherung

<table>

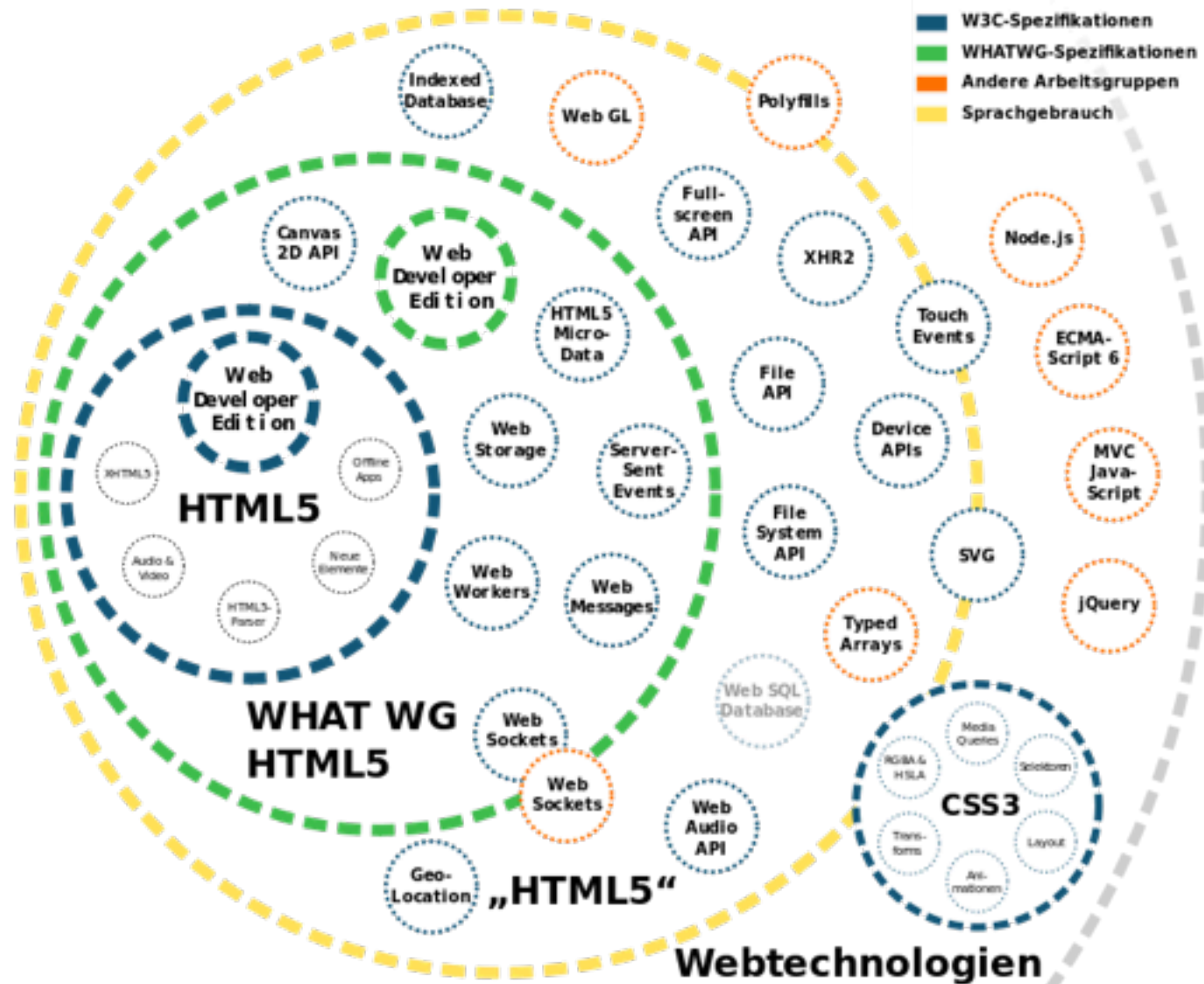
<tr>	<code><th> Stichwort </th></code>	<code><th>Bedeutung</th></code>	</tr>
<tr>	<code><td> VPN </td></code>	<code><td>Virtual Private ...</td></code>	</tr>
<tr>	<code><td> Phishing </td></code>	<code><td>...</td></code>	</tr>

</table>

- Hypertext Markup Language Version 5
- Nachfolger von HTML 4.01 (1999), 1st Draft 2008
- Ersetzt HTML4 & XHTML (Offiziell 2014)
- Entwickelt durch
 - W3C (World Wide Web Consortium)
 - WHATWG (Web Hypertext Application Technology Working Group)
- Gründe für die Entwicklung
 - Neue Funktionen mit Basis von HTML, CSS, DOM & JS
 - Externe Plugins sollen vermieden werden (z.B. Flash)
 - Komfortables Error Handling
 - Scripting soll durch Markup (Auszeichnung) ersetzt werden
 - Gerätunabhängigkeit



WORK IN PROGRESS!



HTML Doctype



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01 Transitional//EN" "http://
www.w3.org/TR/html4/loose.dtd">
```

```
<html>
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

1997

```
<?xml version="1.0" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
```

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

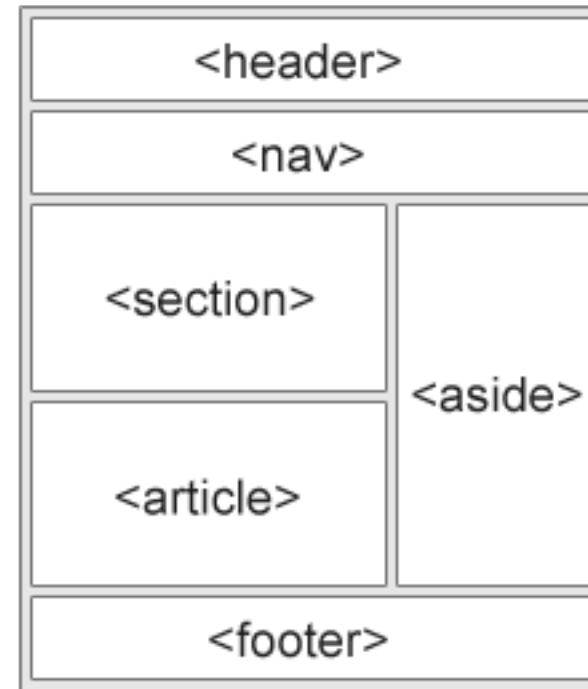
2000



```
<!doctype html>
<html lang="de">
  <head>
    <title></title>
  </head>
  <body>
    </body>
</html>
```

HTML 5 neue Elemente

```
<!doctype html>
<html lang="de">
  <head>
    ...
  </head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <section>...</section>
    <article>
      <video>...</video>
    </article>
    <aside></aside>
    <footer>...</footer>
  </body>
</html>
```



`<input type="text" required />`

`<input type="text" pattern="[0-9]{13,16}"/>`

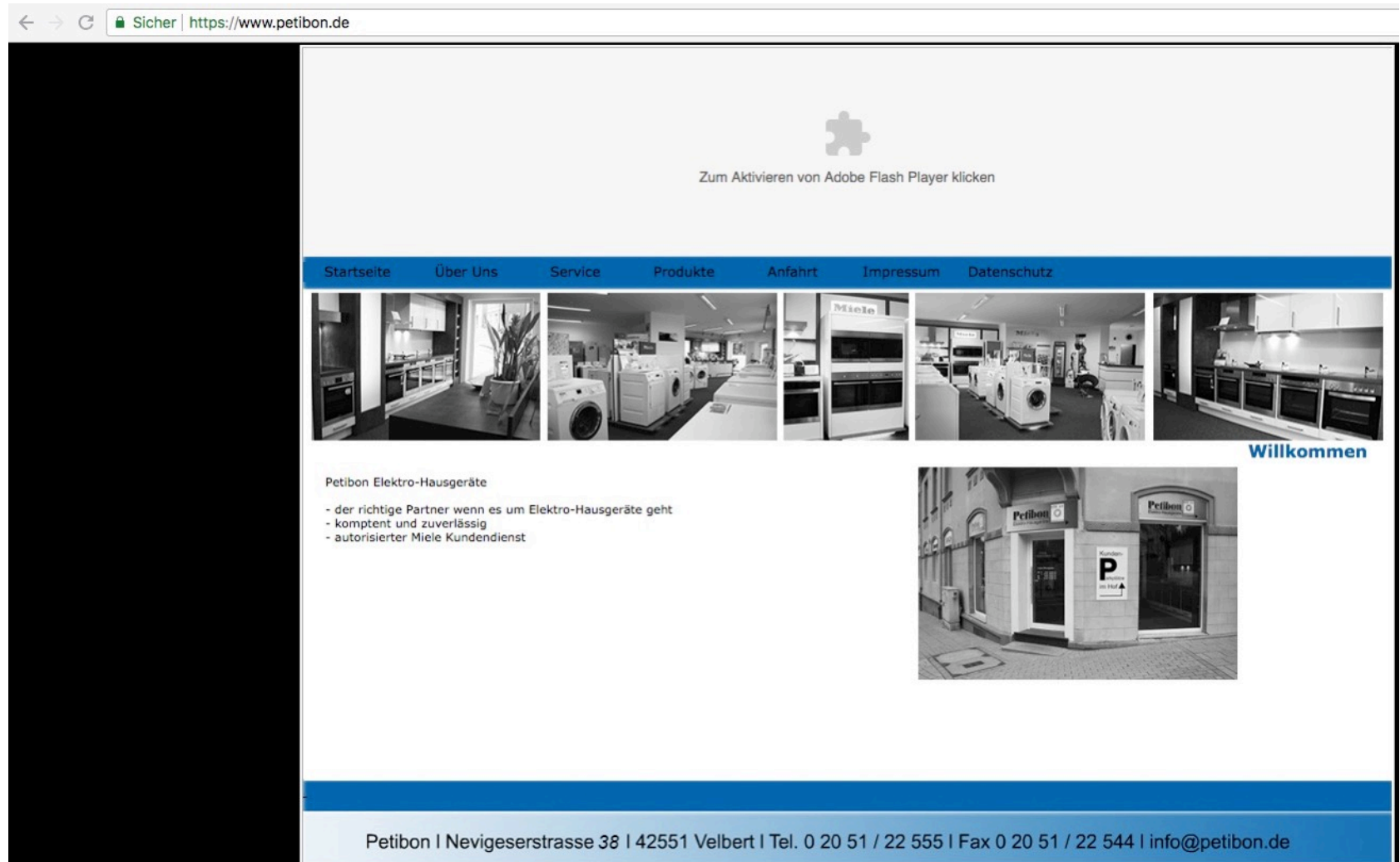
`<input type="text" name="ustidnr" pattern="^[A-Z]{2}[0-9A-Z]{1,12}$"/>`

`<input type="email"/>`

`<input type="datetime" min="2011-12-01T00:00Z"/>`

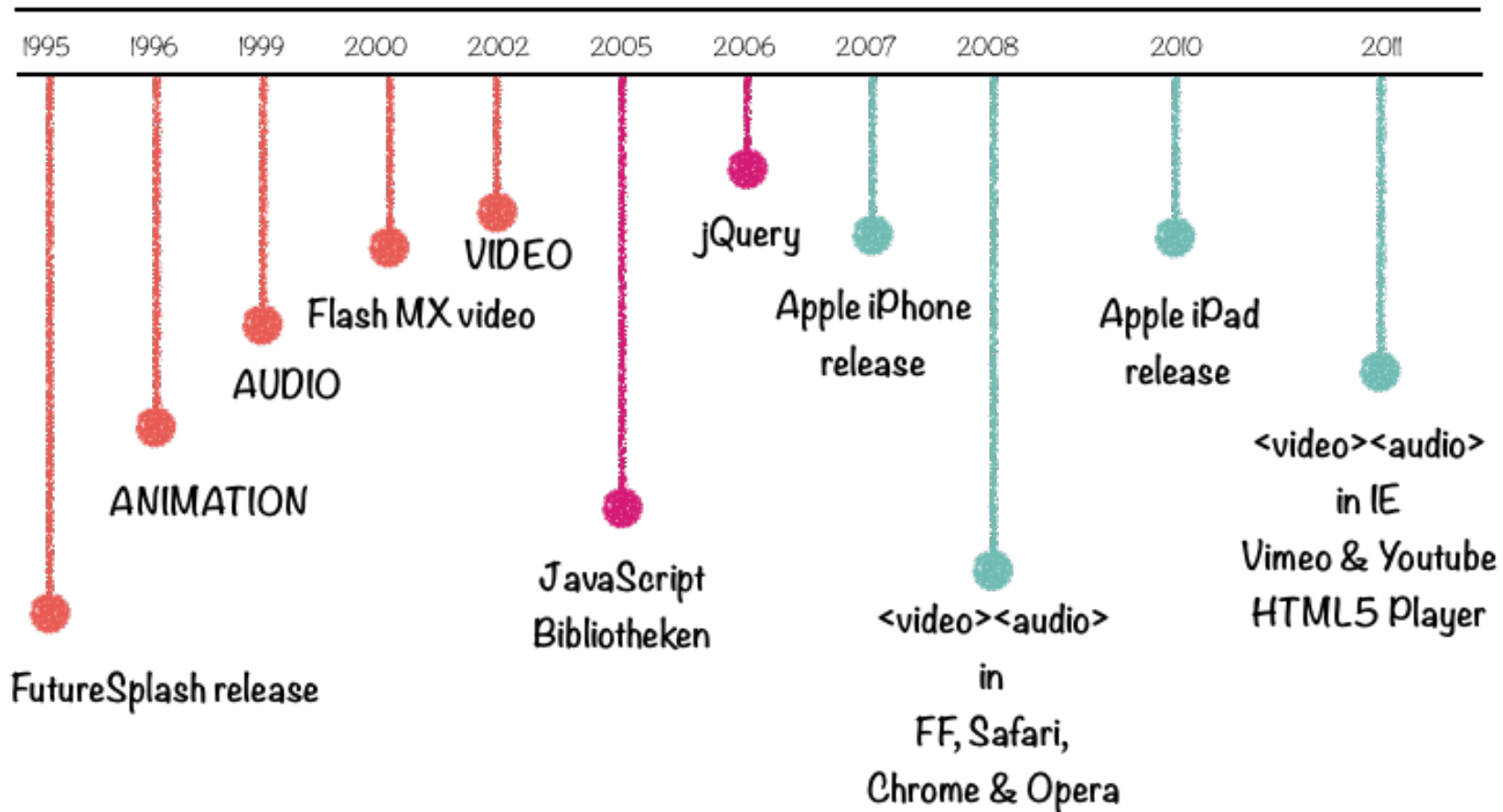
Aufbau und grundlegende Funktionsweise der Webprogrammierung

HTML 5 vs. Flash



Aufbau und grundlegende Funktionsweise der Webprogrammierung

HTML 5 vs. Flash



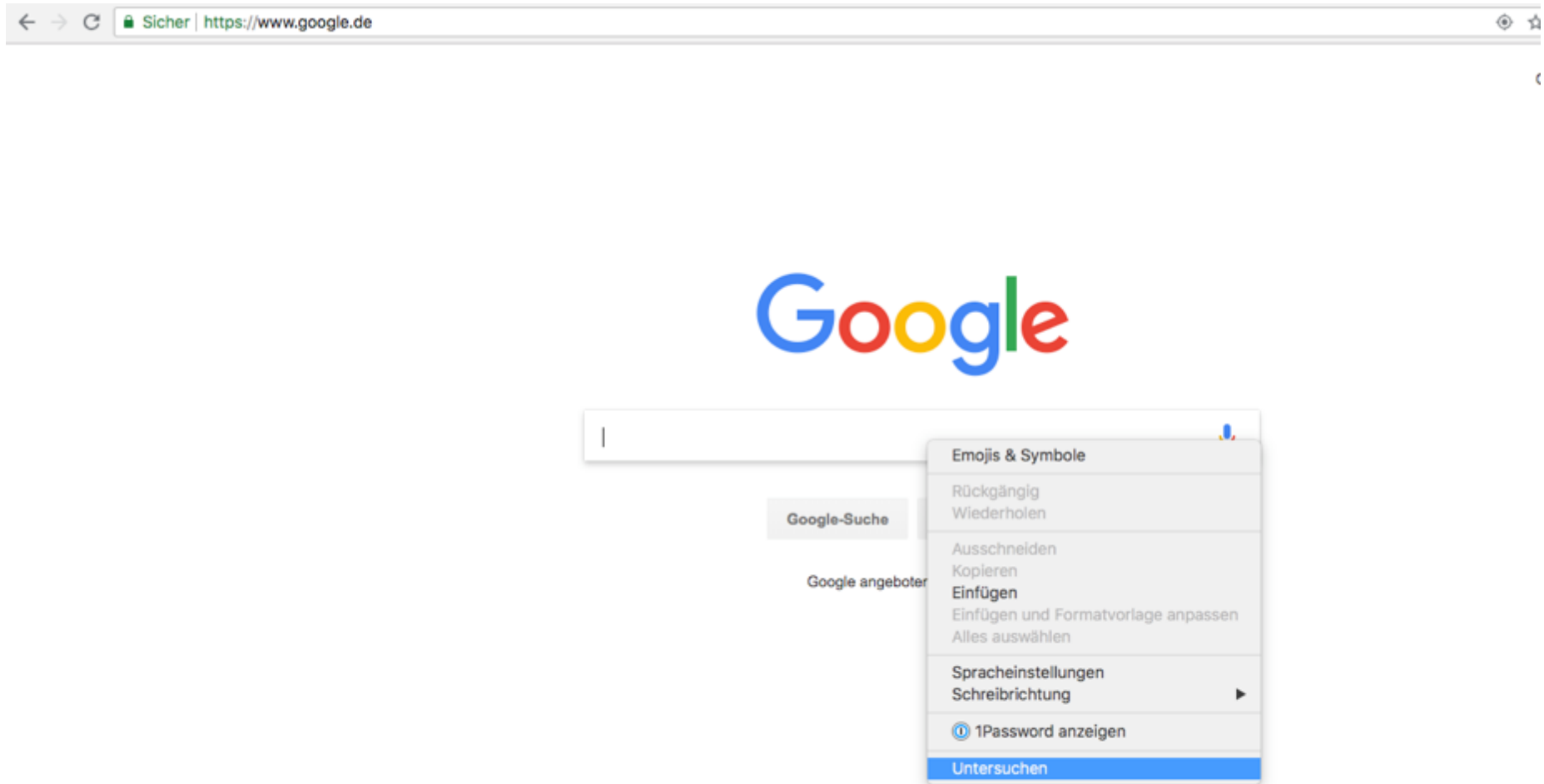
```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
  <source src="movie.webm" type="video/webm">  
  <object data="movie.mp4" width="320" height="240">  
    <embed src="movie.swf" width="320" height="240">  
  </object>  
</video>
```

HTML5 Video

HTML Document Object Model (DOM) Beispiel

1. <https://www.google.de> im beliebigen Browser aufrufen
2. Ggf. Entwickler Funktion im Browser aktivieren
3. Quellcode des Suchfeldes anzeigen lassen (in Chrome z.B. per Rechtsklick und dann „Untersuchen“ auswählen)
4. Welches HTML Element bzw. welcher Tag wird für das Suchfeld verwendet?
5. Welche Elemente/Tags befinden sich auf der ersten und zweiten Ebene des HTML Dokumentes?
6. Welche Elemente/Tags befinden sich innerhalb des <body> Tag?

HTML Document Object Model (DOM) Beispiel



Cookies sind einfache Textdateien, die der beschränkten Archivierung von Informationen dienen. Dieser Sitzungsbezeichner hilft einer Server-Instanz den aufrufenden Client wiederzuerkennen.

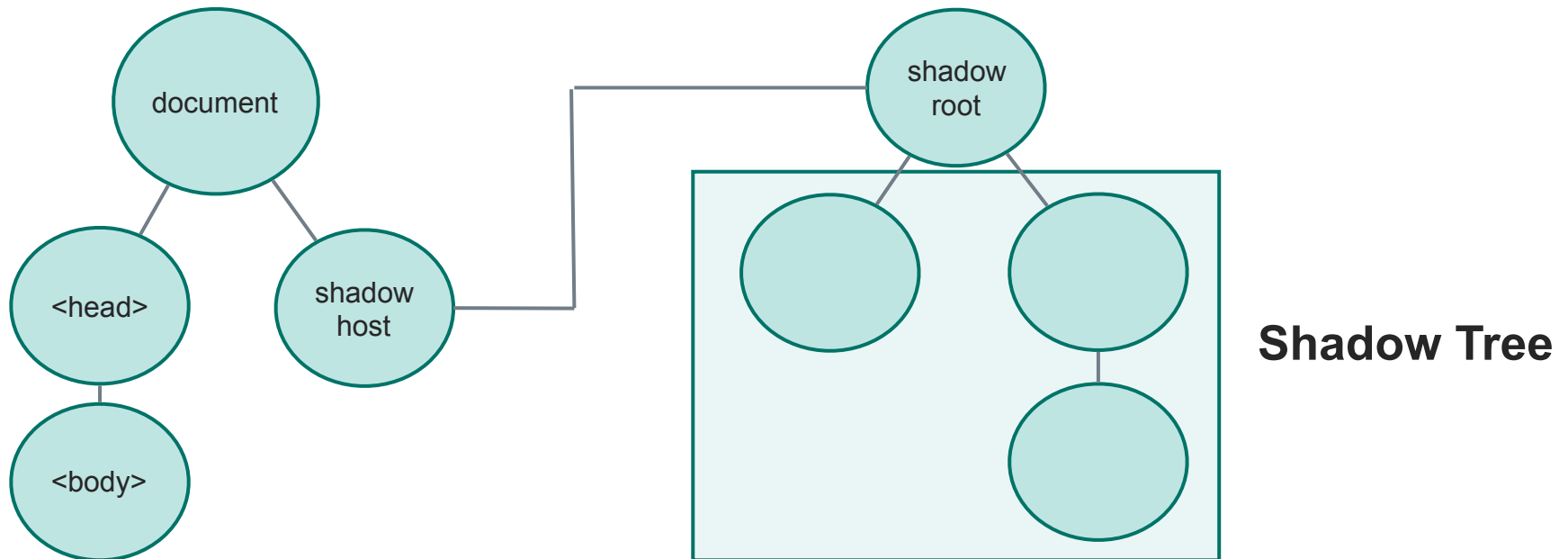
Es existieren **drei Basis-Arten** von Cookies: Session-Cookies, Permanente Cookies und Sicherheits-Cookies (nur via HTTPS). Cookies haben eine bestimmte Lebensdauer.

Darüber hinaus muss unterschieden werden in SameSite-Cookies oder ThirdParty-Cookies (oftmals bei Werbeanzeigen von Drittanbietern).

Super-Cookies werden dauerhaft auf dem Rechner des Nutzers gespeichert.

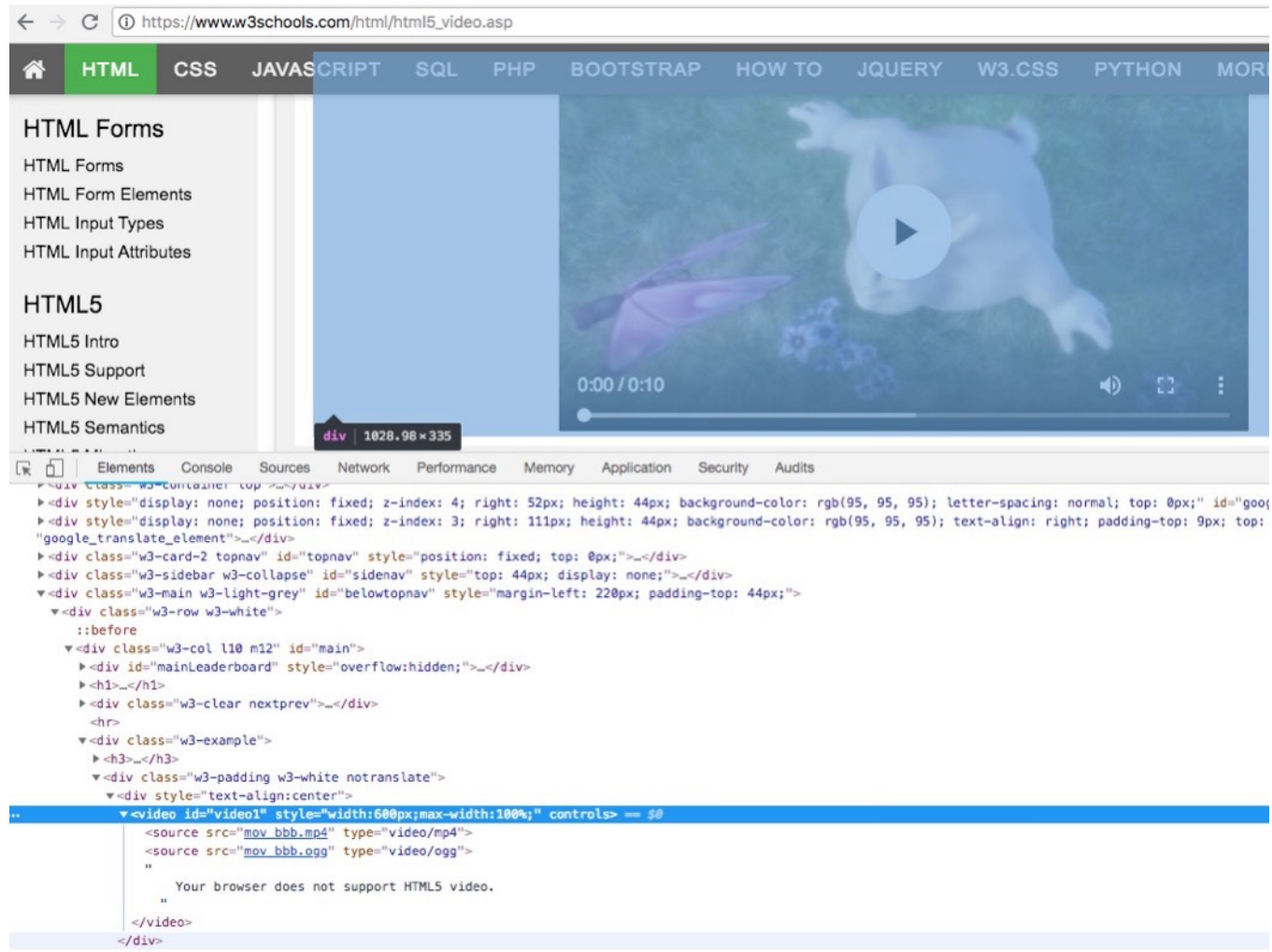
- Web Storage (auch DOM-Storage).
- Strings und JSON-Objekte.
- Bis zu 5MB in Firefox, 10 MB im Internet Explorer.

- Ermöglicht einen versteckten DOM Zweig
- Im Quellcode nicht ersichtlich
- wird z.B. bei `<video>` Element verwendet
 - Video ist sichtbar
 - Buttons/Controls sind nicht im HTML Quellcode zusehen
 - Buttons/Controls werden aber im Browser angezeigt



Aufbau und grundlegende Funktionsweise der Webprogrammierung

Shadow DOM

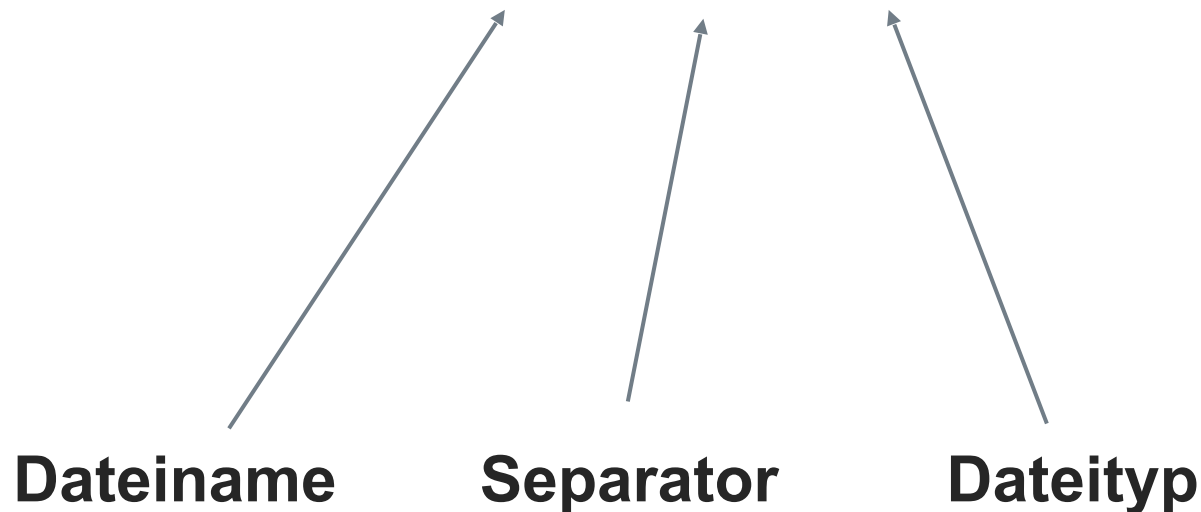


https://www.w3schools.com/html/html5_video.asp

CSS

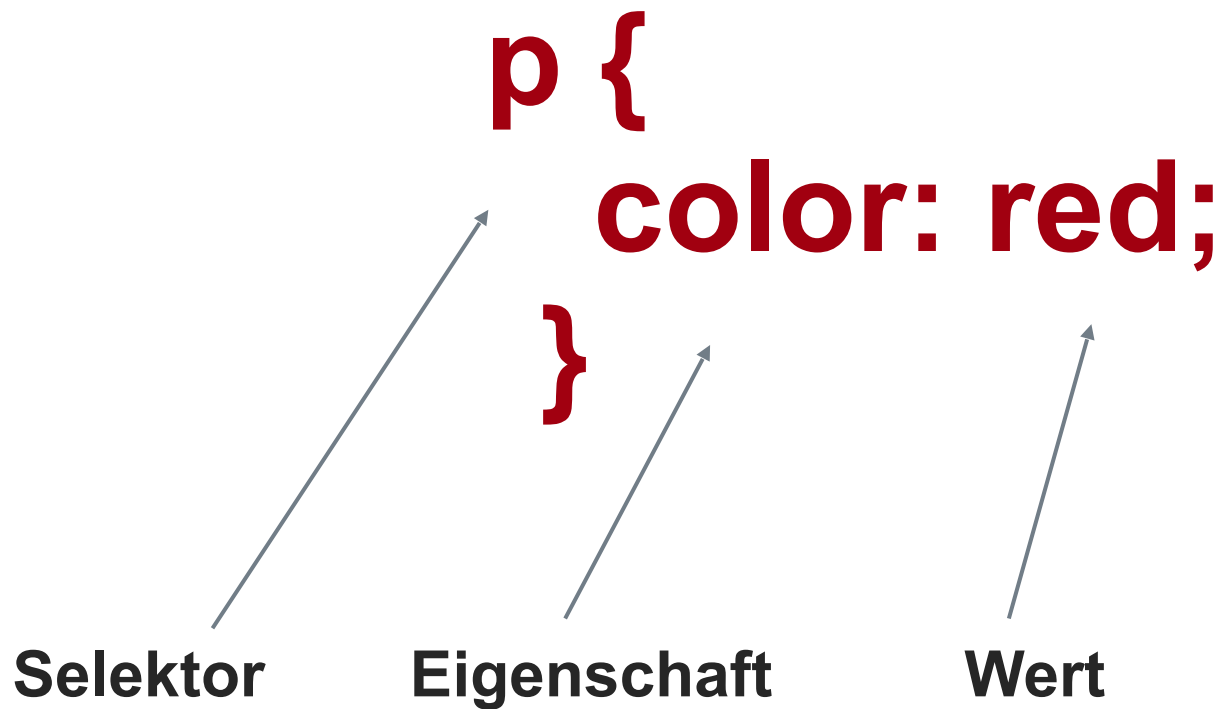
Textdokument

style.css



- Cascading Stylesheets
- Dient dem Layouting von HTML
- Ermöglicht Manipulation jedes HTML Elementes
- Enthält Selektoren und Eigenschaften
- Klassen und ID des HTML dienen der Individualisierung
- Trennt Layout vom Inhalt (Text) und Grundrahmen (HTML)
- Ermöglicht das Austauschen von Layout ohne den Inhalt oder Grundrahmen zu ändern
- Muss ins HTML Dokument eingebunden werden
 - inline (direkt im HTML Code geschrieben)
 - internal (in separaten `<style></style>` Block)
 - external (als separate CSS Datei einbinden)

Trennung von Layout & Content
<http://www.csszengarden.com/>



**a:hover {
color: red;
}**

Selektor

Eigenschaft

Wert

```
.class {  
    color: red;  
}
```

Selektor

Eigenschaft

Wert

**#id {
color: red;
}**

Selektor

Eigenschaft

Wert

Aufbau und grundlegende Funktionsweise der Webprogrammierung

CSS Kompatibilität

← → ↻ Sicher | <https://www.canIuse.com/#search=background-color> ☆

Home News July 29, 2018 - New feature: createImageBitmap Compare browsers About

Slack — It's teamwork, but simpler, more pleasant and more productive.

Can I use ? Settings

2 results found

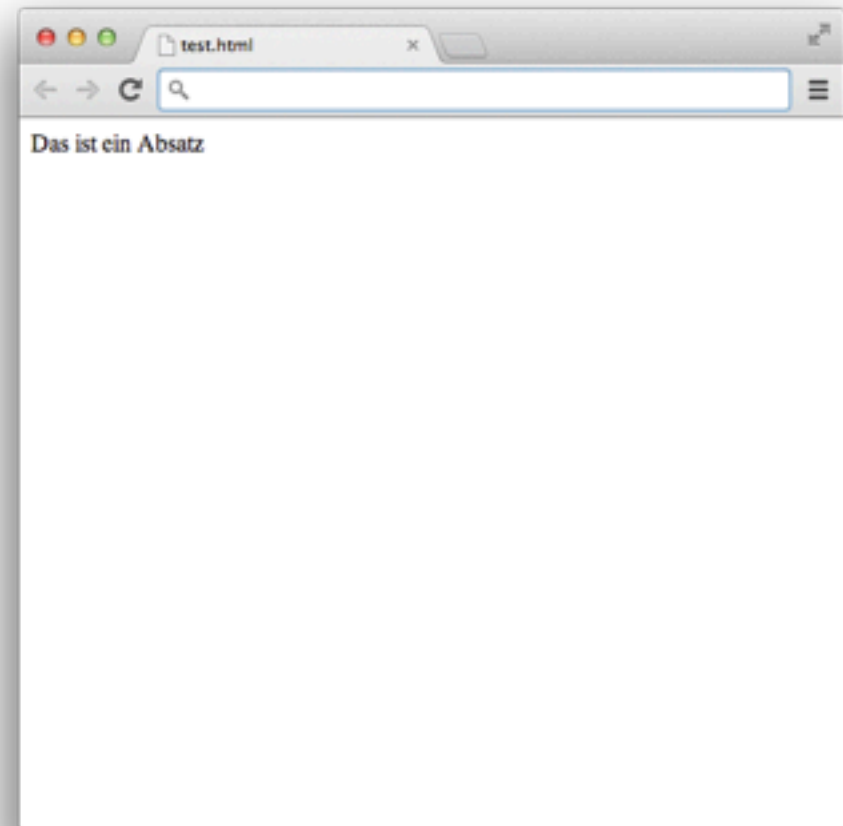
Detected your country as "Germany". Would you like to import usage data for that country?

CSS `currentColor` value - REC Usage % of: all users 95.46%

A CSS value that will apply the existing `color` value to other properties like `background-color`, etc.

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			66		10.3				
			67		11.2				4
11	17	61	68	11.1	11.4	all	67	11.8	7.2
	18	62	69	12	12				

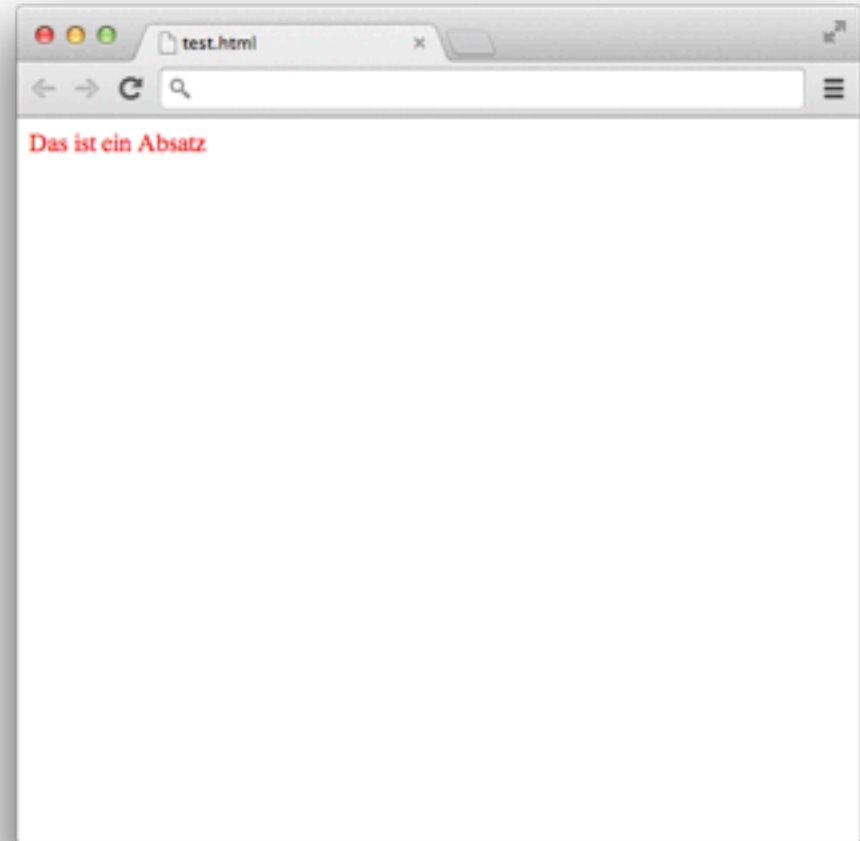
```
...  
<html>  
  <head>...</head>  
  <body>  
    <p>Das ist ein Absatz</p>  
  </body>  
</html>
```



Aufbau und grundlegende Funktionsweise der Webprogrammierung

CSS Inline Style Sheets

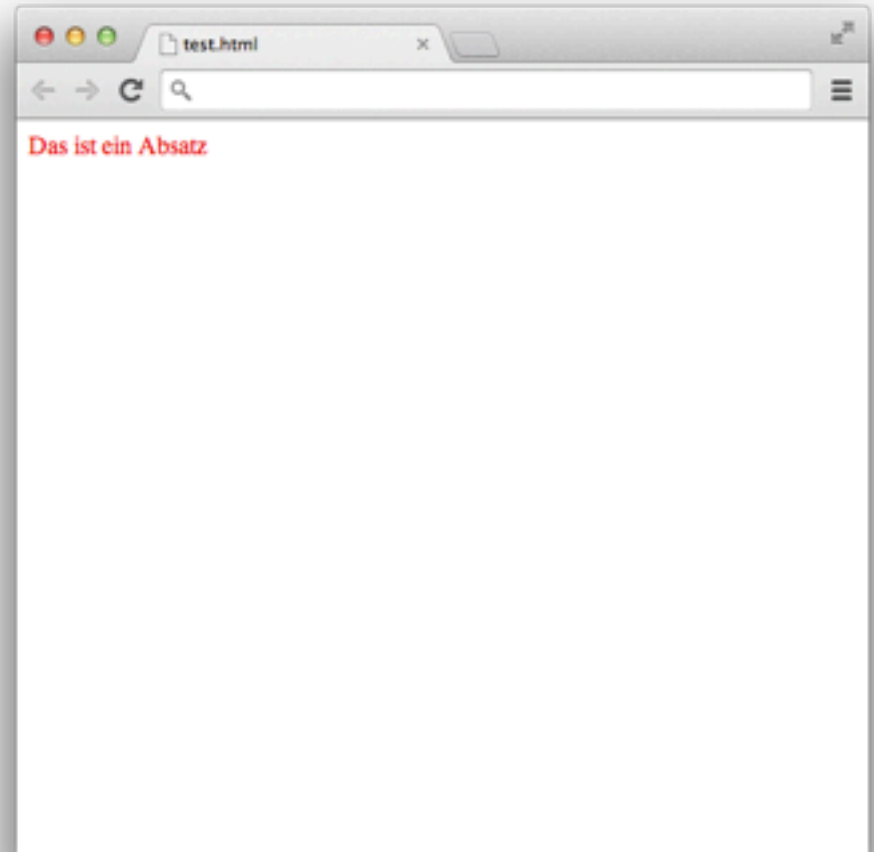
```
...  
<html>  
  <head>...</head>  
  <body>  
    <p style="color:red;">  
      Das ist ein Absatz  
    </p>  
  </body>  
</html>
```



Aufbau und grundlegende Funktionsweise der Webprogrammierung

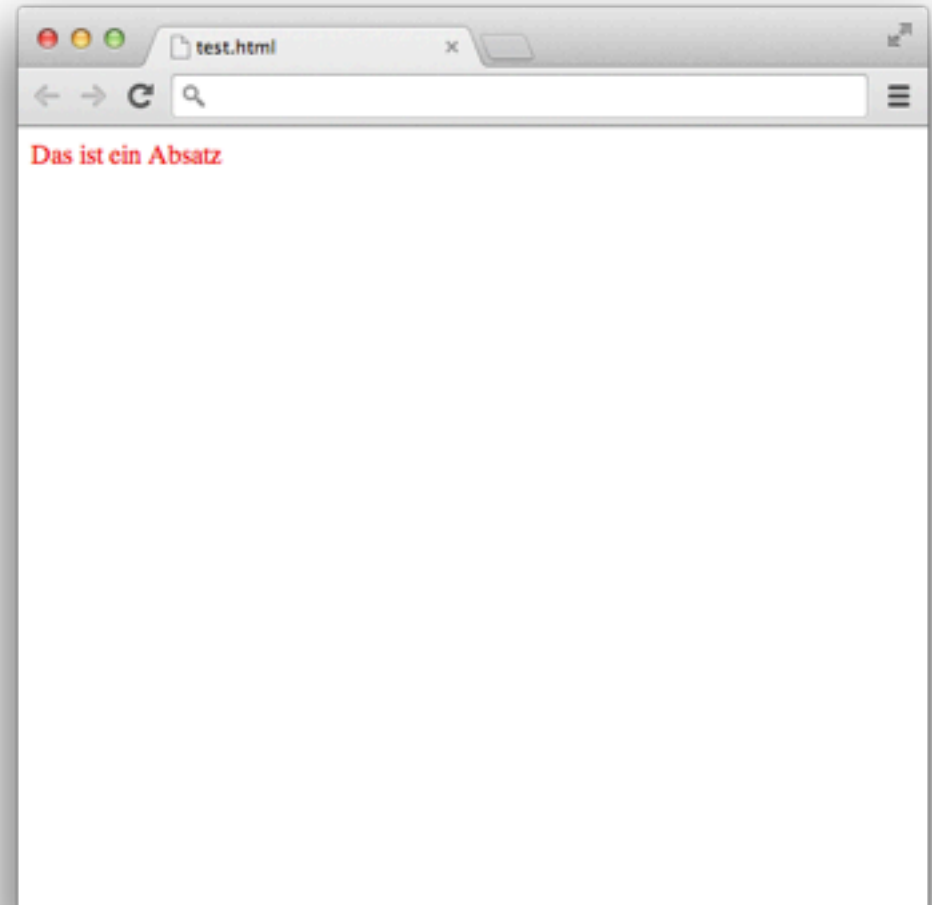
CSS Internal Style Sheets

```
...  
<html>  
  <head>  
    <style>  
      p {color: red;}  
    </style>  
  </head>  
  <body>  
    <p>Das ist ein Absatz</p>  
  </body>  
</html>
```

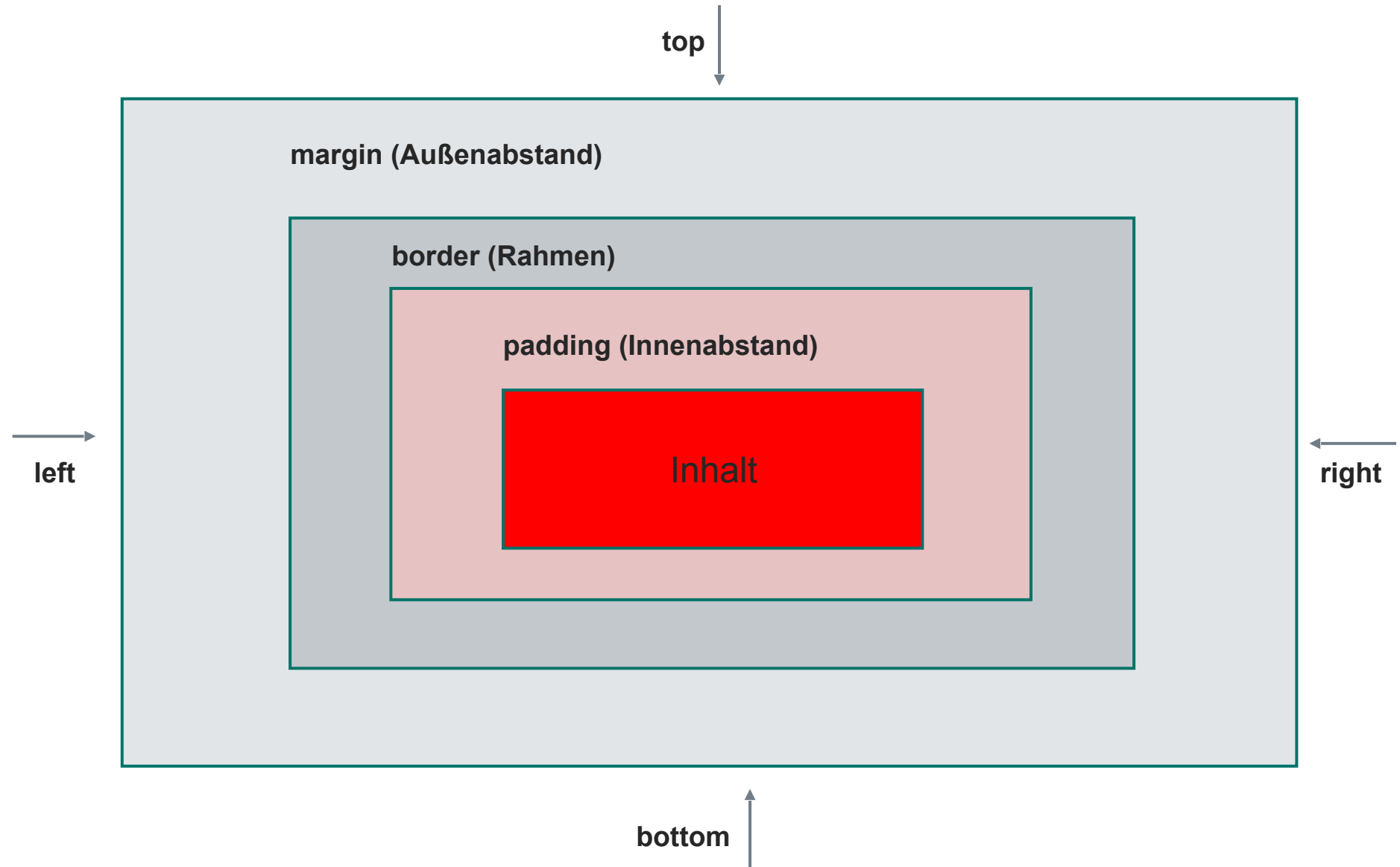


CSS External Style Sheets

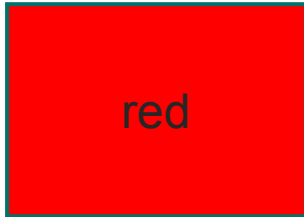
```
***  
<html>  
  <head>  
    <link rel="stylesheet"  
type="text/css" href="style.css">  
  </head>  
  <body>  
    <p>Das ist ein Absatz</p>  
  </body>  
</html>
```



CSS BoxModell



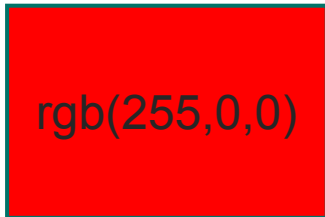
```
box-eins {  
    width: 300px;  
    height: 300px;  
    margin: 100px;  
    padding: 2em;  
}  
  
box-zwei {  
    width: 300px;  
    height: 300px;  
    position: absolute;  
    top: 20px;  
    right: 40px;  
    bottom: 150px;  
    left: 0px;  
}
```



```
box {  
  background-color: red;  
}
```



```
box {  
  background-color: #ff0000;  
}
```



```
box {  
  background-color: rgb(255,0,0);  
}
```

Test von Farben: <http://www.colorblender.com>

```
@keyframes animation1 {  
    from {width: 250px;}  
    to {width: 500px;}  
}  
  
div {  
    width: 250px;  
    height: 100px;  
    background-color: red;  
    animation-name: animation1;  
    animation-duration: 3s;  
}
```

Aufbau und grundlegende Funktionsweise der Webprogrammierung

CSS Animation Beispiel



Siehe: https://www.w3schools.com/css/css3_animations.asp

- Cascading Stylesheets Level 3
- Nachfolger von Level 2 (1998)
- Ausführung ist Clientseitig
- Modulare Entwicklung
- Gesplittet in diverse Teiltechniken
- Entwickelt durch W3C
(World Wide Web Consortium)
- Vollständig Rückwärtskompatibel



- Alle vorherigen CSS Funktionen wurden implementiert
- Gründe für die Entwicklung
 - Media Queries
 - Animationen
 - Effekte (Verläufe etc.)
 - Webfonts
 - mobile Geräte
 - „Intelligenz“ von CSS

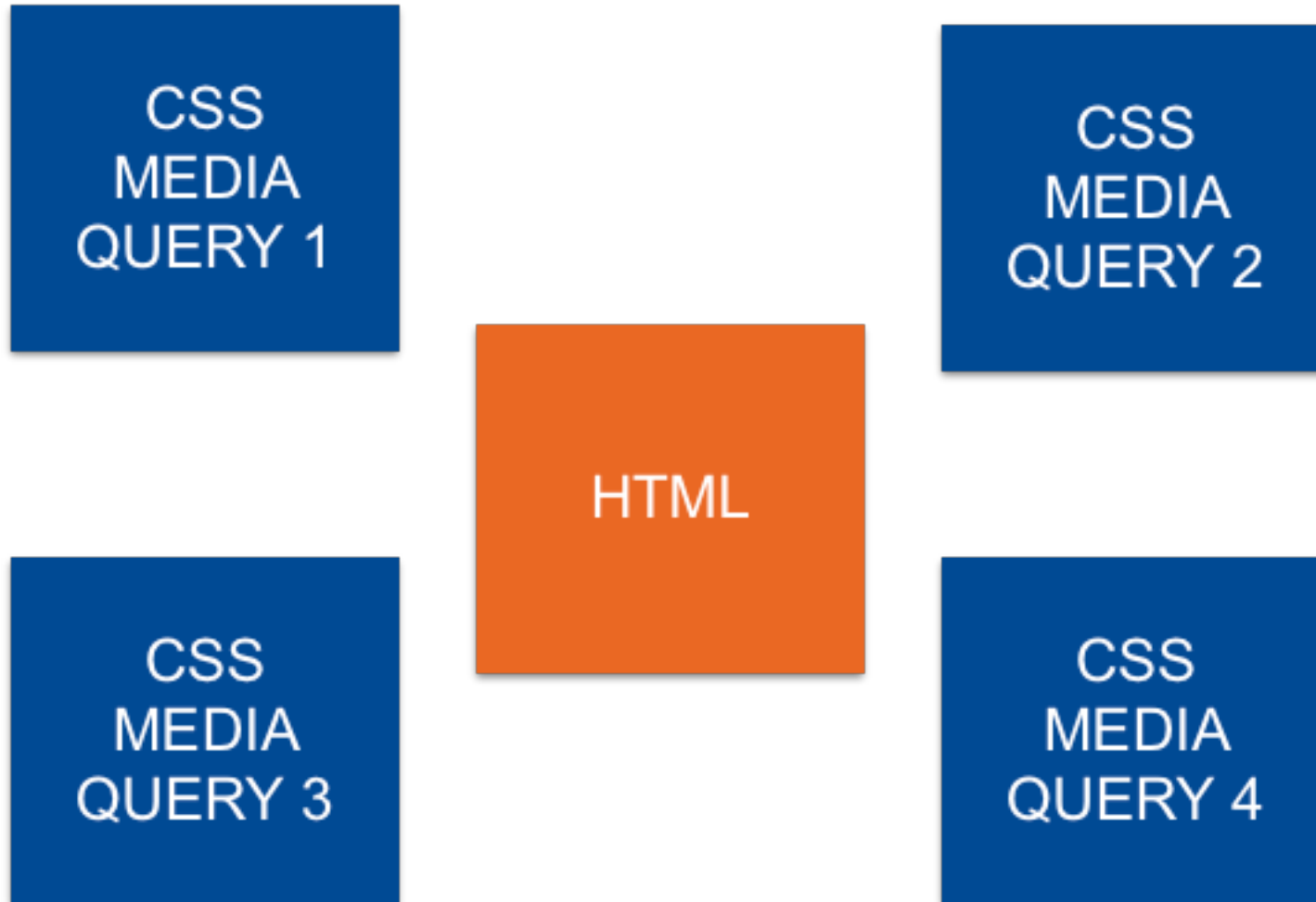
CSS



- CSS pro Ausgabemedium
- Zuordnung erfolgt per Medienabfrage (Media Queries)
- Media Queries bestehen aus
 - Medientyp (z.B. Screen oder Printer)
 - Medienmerkmal (z.B. Farbfähigkeit oder Landscape Modus)
- Medienabfrage entweder
 - Im HTML Dokument über `<link rel="stylesheet" href="print.css" media="print">`
 - Oder im CSS Dokument über `@media print {}`

- Grundlage für Responsive Webdesign und Grid-Systeme wie z.B.
 - bootstrap
 - Foundation
 - 1140px Grid-System

- Gründe für die Entwicklung
 - Mobile Endgeräte
 - Förderung der Barrierefreiheit
 - Dynamischere Anzeige (Auflösung)
 - Regelwerk“ für mehrspaltige Inhalte
 - Konsistentes Design (durch Grid-Systeme)



- Frameworks beinhalten vorgefertigte Funktionen
- Frameworks sind nicht CSS typisch, sondern gibt es in fast jeder Stylesheet-/ Programmiersprache
- Framework Einbindung über Datei
- Funktionen Verwendung per Aufruf
- Ermöglichen Wiederverwendbarkeit
- Verhindern Code Wiederholung
- Frameworks bilden z.B. auch Pattern/Paradigmen ab
- CSS Frameworks bieten u.a. auch Responsive Web Design
 - bootstrap
 - Foundation
 - 114px System
- CSS Frameworks für Responsive Web Design basieren u.a. auf Grid System



Foundation
Start here, build everywhere.

Bootstrap Framework

- Entwickelt durch twitter
- Gliedert das Layout in ein Grid von 12 Spalten
- Einbindung von CSS & JS -> get started!
- Populärstes GitHub Projekt



```
// Extra small devices (portrait phones, less than 576px)
// No media query for `xs` since this is the default in Bootstrap

// Small devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }

// Medium devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

// Extra large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }
```

Aufbau und grundlegende Funktionsweise der Webprogrammierung

Bootstrap Framework Grid System

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

```

<div class="row">
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
</div>
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>

```

CSS Framework Beispiel

1. **Editor öffnen**
2. **Datei mit dem Namen index.html auf dem Desktop/Schreibtisch ablegen**
3. **Starter Template von Bootstrap in den Inhalt übernehmen**
<https://getbootstrap.com/docs/4.1/getting-started/introduction/>
4. **Bootstrap einbinden (bereits im Starter Template enthalten)**
5. **Datei im Browser öffnen („Hello, world!“ muss angezeigt werden)**
6. **DIVs aus der vorherigen Folie übernehmen**
7. **Browserfenster neuladen**
8. **Browserfenster vergrößern/verkleinern**
9. **Welches Verhalten kann festgestellt werden?**

Schreinerei

Als Bau- und Möbelschreinerei,
ansässig in Velbert, sind wir seit vielen
Jahrzehnten für unsere Kunden tätig.
Wir überzeugen mit unserer
langjährigen Erfahrung und dem
Perfektionismus in allen Bereichen für
die Werkstoffe Holz, Kunststoff und
Glas.

Wir sind mit Leidenschaft bei der
Arbeit, was sich in unseren
Werkstücken widerspiegelt.

Schreinerei

Als Bau- und Möbelschreinerei, ansässig in Velbert, sind wir seit vielen Jahrzehnten für unsere Kunden tätig. Wir überzeugen mit unserer langjährigen Erfahrung und dem Perfektionismus in allen Bereichen für die Werkstoffe Holz, Kunststoff und Glas.

Wir sind mit Leidenschaft bei der Arbeit, was sich in unseren Werkstücken widerspiegelt.

} `<h2>`

} `<p>`

} `<p>`

Aufbau und grundlegende Funktionsweise der Webprogrammierung

CSS – Responsive Design

Schreinerei

Als Bau- und Möbelschreinerei, ansässig in Velbert, sind wir seit vielen Jahrzehnten für unsere Kunden tätig. Wir überzeugen mit unserer langjährigen Erfahrung und dem Perfektionismus in allen Bereichen für die Werkstoffe Holz, Kunststoff und Glas.

Wir sind mit Leidenschaft bei der Arbeit, was sich in unseren Werkstücken widerspiegelt.

Familienbetrieb

Unser Betrieb wird bereits seit mehreren Generationen durch die Familie Bruckhaus geführt und passt sich ständig den neuesten Entwicklungen an, was sich vor allem im Bereich der Designermöbel widerspiegelt.

Seit über zwanzig Jahren wird der Betrieb vom Inhaber Stefan Bruckhaus geleitet und passt sich dem stetigen Wachstum des Betriebes an.

Meisterbetrieb

Durch die lange Historie unseres Unternehmens, legen wir natürlich viel Wert auf Erfahrung und Wissen. Dadurch ist es selbstverständlich, dass wir ein Meisterbetrieb sind und auch stetig jungen Mitarbeiter ermöglichen diesen Weg zu gehen.

In unseren Projekten profitieren Sie von unserem Know-How und den Erfahrungen, die sich im Ergebnis widerspiegeln.

Ausbildungsbetrieb

Wir legen viel Wert darauf jungen Menschen eine Chance zu geben und bilden daher im Schreinerhandwerk aus. Über ein Praktikum ermöglichen wir jungen Menschen, den Beruf kennen zu lernen, um bereits die richtigen Weichen zu stellen.

Schicken Sie uns einfach Ihre Initiativbewerbung per Post zu und nutzen Ihre Chance, um den Beruf des Tischlers zu erlernen!

Aufbau und grundlegende Funktionsweise der Webprogrammierung

CSS – Responsive Design

Schreinerei

Als Bau- und Möbelschreinerei, ansässig in Velbert, sind wir seit vielen Jahrzehnten für unsere Kunden tätig. Wir überzeugen mit unserer langjährigen Erfahrung und dem Perfektionismus in allen Bereichen für die Werkstoffe Holz, Kunststoff und Glas.

Wir sind mit Leidenschaft bei der Arbeit, was sich in unseren Werkstücken widerspiegelt.

Familienbetrieb

Unser Betrieb wird bereits seit mehreren Generationen durch die Familie Bruckhaus geführt und passt sich ständig den neusten Entwicklungen an, was sich vor allem im Bereich der Designermöbel widerspiegelt.

Seit über zwanzig Jahren wird der Betrieb vom Inhaber Stefan Bruckhaus geleitet und passt sich dem stetigen Wachstum des Betriebes an.

Meisterbetrieb

Durch die lange Historie unseres Unternehmens, legen wir natürlich viel Wert auf Erfahrung und Wissen. Dadurch ist es selbstverständlich, dass wir ein Meisterbetrieb sind und auch stetig jungen Mitarbeiter ermöglichen diesen Weg zu gehen.

In unseren Projekten profitieren Sie von unserem Know-How und den Erfahrungen, die sich im Ergebnis widerspiegeln.

Ausbildungsbetrieb

Wir legen viel Wert darauf jungen Menschen eine Chance zu geben und bilden daher im Schreinerhandwerk aus. Über ein Praktikum ermöglichen wir jungen Menschen, den Beruf kennen zu lernen, um bereits die richtigen Weichen zu stellen.

Schicken Sie uns einfach Ihre Initiativbewerbung per Post zu und nutzen Ihre Chance, um den Beruf des Tischlers zu erlernen!

<div>

<div>

<div>

<div>

```
...
<body>
  <div class="container">...</div>
  <div class="container">...</div>
  <div class="container">...</div>
  <div class="container">...</div>
</body>
...
```

```
.container {
...
  width:200px;
  height:400px;
...
}
```

**width und height
sind nicht flexibel!!**

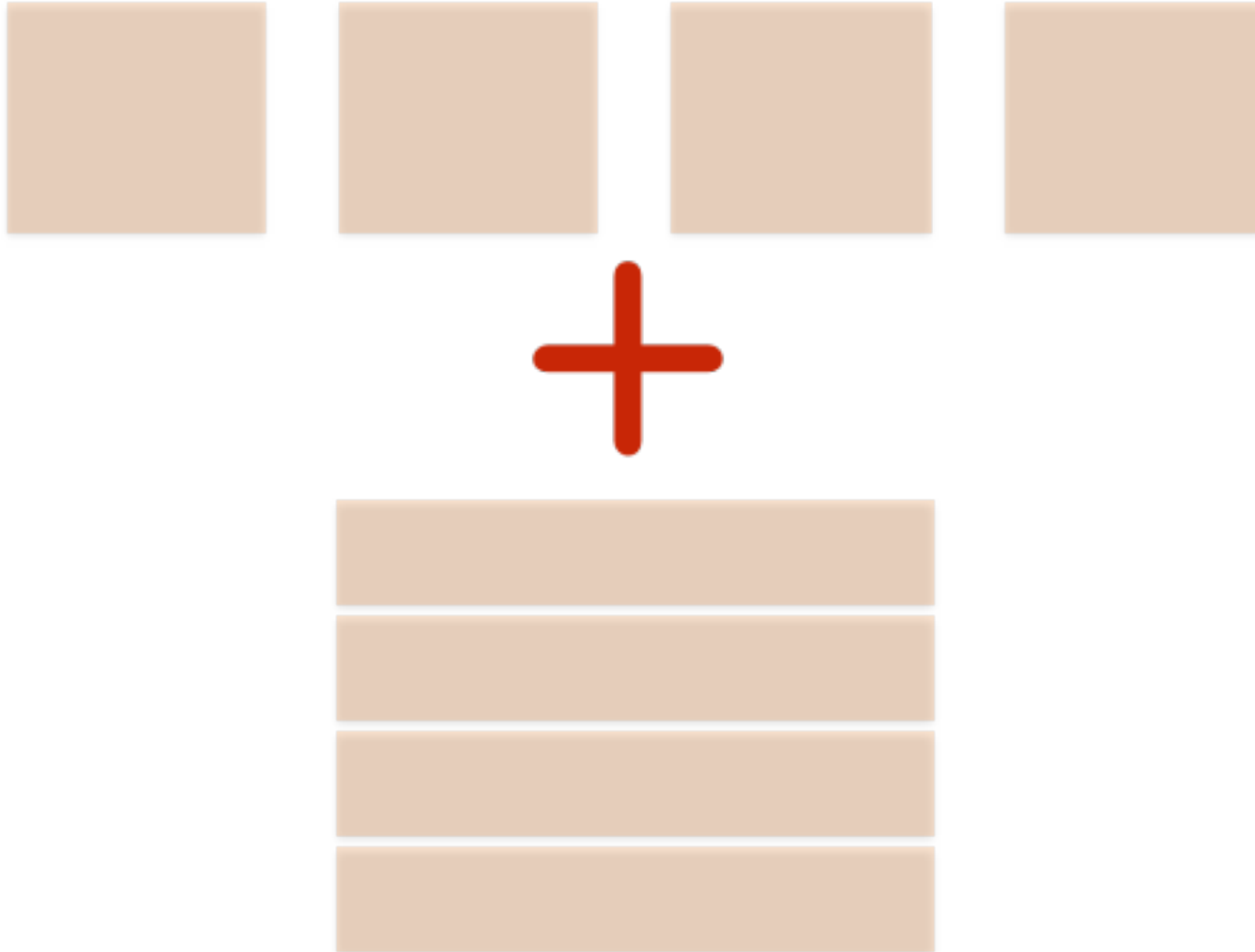
```
...
<body>
  <div class="container">...</div>
  <div class="container">...</div>
  <div class="container">...</div>
  <div class="container">...</div>
</body>
...
```

```
.container {
...
  width:200px;
  height:400px;
...
}
```

**width und height
sind nicht flexibel!!**

Aufbau und grundlegende Funktionsweise der Webprogrammierung

CSS – Responsive Design



DESKTOP



MOBILE

Aufbau und grundlegende Funktionsweise der Webprogrammierung

CSS – Responsive Design



Aufbau und grundlegende Funktionsweise der Webprogrammierung

CSS – Responsive Design



- Verschiedene Geräte erfordern flexibles Layout/Template
- Responsives und Adaptive Design sind Lösungsansätze
- Responsives Web Design (lt. Ethan Marcotte)
 - Prozentbasiertes/fests Grid
 - Flexible Breite der Medien (Bilder etc.)
 - Verwendung von Media Queries
 - Vergleichbar mit Responsive Web Design aus der Architektur
 - Liefert flexibles/fluides Layout für alle Screengrößen
- Adaptive Design
 - Verwendung von Media Queries
 - Liefert verschiedene Layouts je Screengröße

- Als Ergänzung der Standardschriftarten auf dem Clientsystem
- Webfonts zur Einbindung von Schriftarten/Icons
- Werden vom Client beim Aufruf geladen
- Entweder per Einbindung vom Server oder per CDN (Content-Delivery-Network)

font-awesome.css

```
1  /*!  
2   * Font Awesome 4.0.3 by @davegandy - http://fontawesome.io - @fontawesome  
3   * License - http://fontawesome.io/license (Font: SIL OFL 1.1, CSS: MIT License)  
4   */  
5  /* FONT PATH  
6   * _____ */  
7  @font-face {  
8    font-family: 'FontAwesome';  
9    src: url('../fonts/fontawesome-webfont.eot?v=4.0.3');  
10   src: url('../fonts/fontawesome-webfont.eot?#iefix&v=4.0.3') format('embedded-opentype'),  
11        url('../fonts/fontawesome-webfont.woff?v=4.0.3') format('woff'),  
12        url('../fonts/fontawesome-webfont.ttf?v=4.0.3') format('truetype'),  
13        url('../fonts/fontawesome-webfont.svg?v=4.0.3#fontawesomeregular') format('svg');  
14    font-weight: normal;  
15    font-style: normal;  
16  }
```

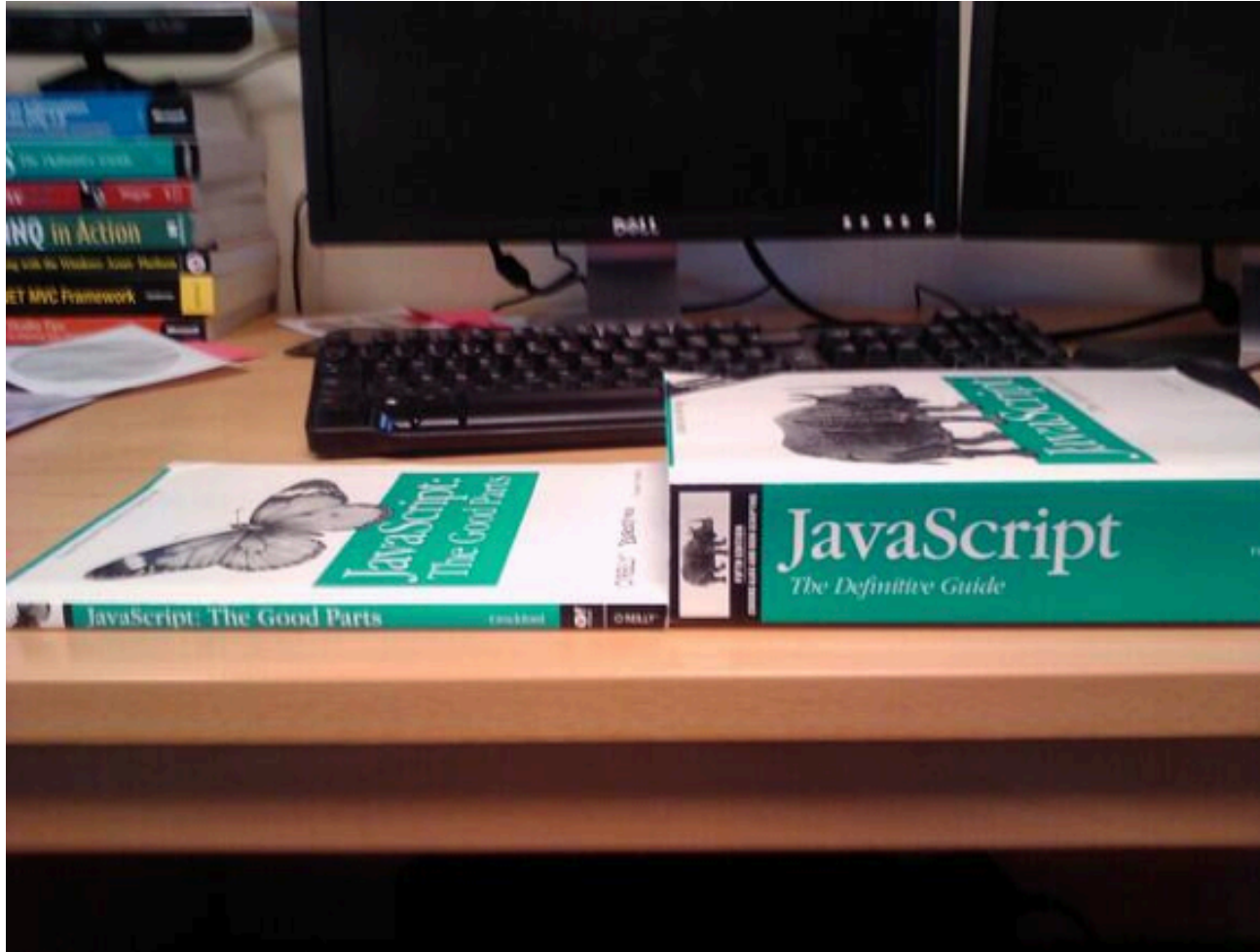


→ **Webfont
„Questrial“**



→ **Arial**

JavaScript



**Vorurteile
JavaScript**

- JavaScript, kurz JS
- Nachfolger von LiveScript (1995), Netscape
- Ausführung ist Clientseitig (wie HTML & CSS)
- Ursprünglicher Zweck
 - Dynamische Generierung von HTML
 - Aktionen auswerten/abfangen
 - Inhalte verändern/generieren
- Heutiger Zweck
 - Asynchrones Ausführen von HTTP Requests & Response (AJAX)
 - Auslagerung Clientseitiger Aktionen durch z.B. Ember.js
- Manipuliert das DOM
- Prozedural/Objektorientiert/Funktional
- Pattern wie z.B. MVC oder Prototypen möglich

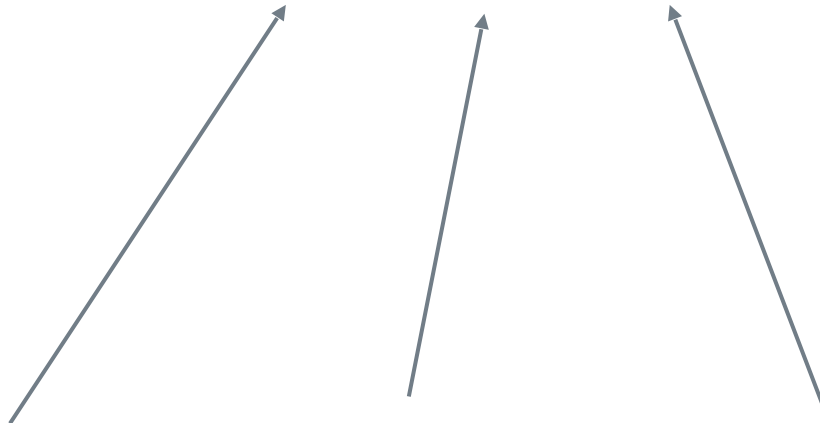
Textdokument

script.js

Dateiname

Separator

Dateityp



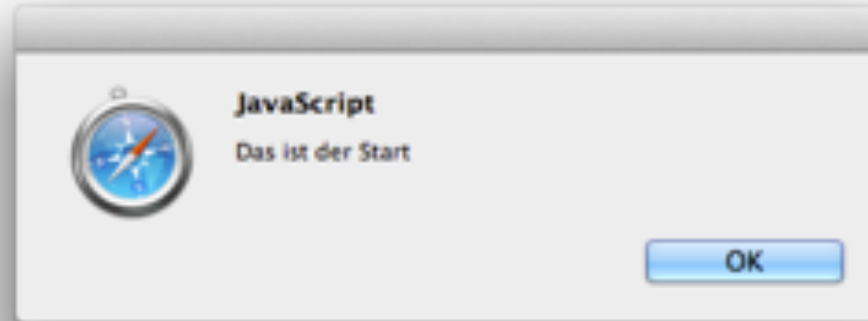
```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <script> ... </script>
  </head>
  <body>
    <p onclick=„window.alert(Es wurde geklickt)“>Das ist ein Absatz</p>
  </body>
</html>
```

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <script type="text/javascript" src="script.js"></script>
  </head>
  <body>
    <p>Das ist ein Absatz</p>
  </body>
</html>
```



script.js

```
function start(text) {  
    window.alert(text);  
}  
start("Das ist der Start");
```



demo.html

```
....  
<body>  
<div id=„demo“></div>  
</body>  
...
```

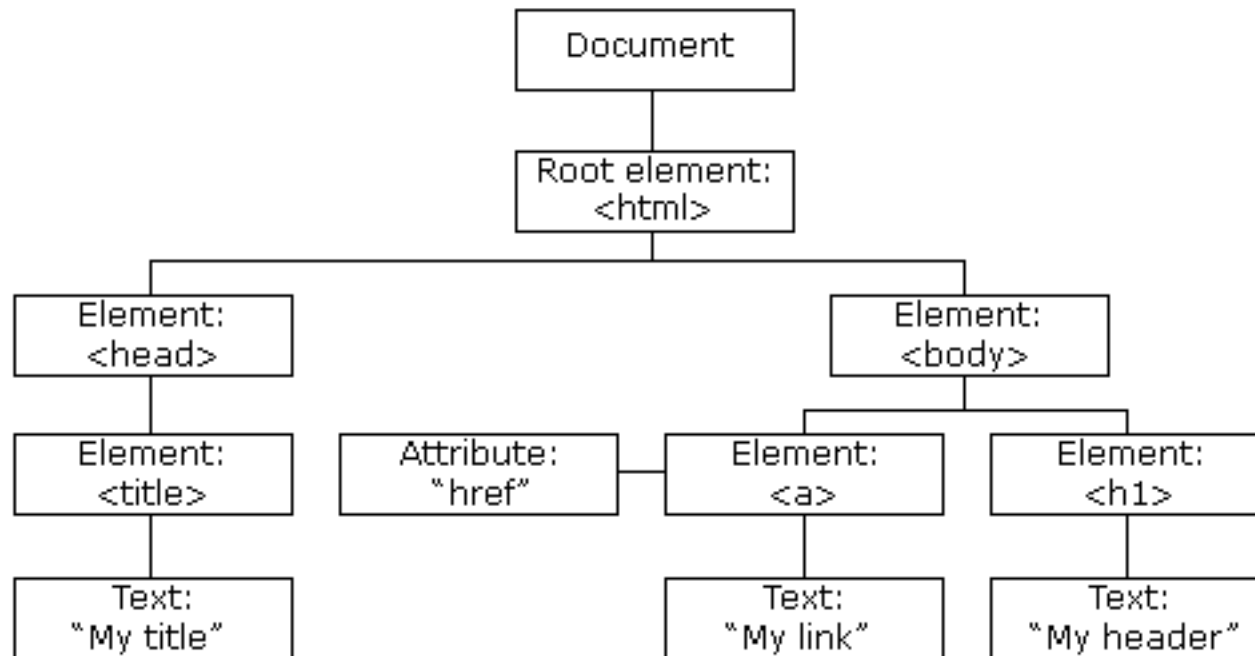
script.js

```
document.getElementById("demo").onclick = function()  
{meineFunktion()};  
  
function meineFunktion() {  
    document.getElementById("demo").innerHTML = "Es wurde gklickt!";  
}
```

Document Object Models

- Von W3C (World Wide Web Consortium) als Standard eingeführt
- Unterteilt in
 - Core DOM (Standard für alle Dokumententypen)
 - XML DOM (Standard für alle XML Dokumente)
 - HTML DOM (Standard für alle HTML Dokumente)
- Beinhaltet alle
 - HTML Elemente als Objekt in Baumstruktur ab
 - Attribute und Methoden um auf alle Elemente zu zugreifen
 - Events (onclick etc.) von allen HTML Elementen
- Das HTML DOM ist ein Standard um auf alle Elemente zu zugreifen, diese zu verändern, neu hinzuzufügen oder zu löschen.

The HTML DOM Tree of Objects



- JavaScript kann
 - alle HTML Elemente der Seite verändern
 - alle HTML Attribute der Seite verändern
 - alle CSS Styles der Seite verändern
 - HTML Elemente und Attribute entfernen
 - neue HTML Elemente erzeugen
 - auf alle HTML Elemente reagieren (u.a. durch Event Handling)
 - neue Events erzeugen

- JavaScript Clientseitig
 - Agiert im Browserfenster (Sandbox)
 - Kann nicht auf das Filesystem zugreifen
 - Kann aber Download von Dateien erzwingen/ermöglichen
 - Same-Origin-Policy (Grundsatz des selben Ursprungs)
 - Es dürfen nur Skripte derselben URL ausgeführt werden
 - Es dürfen nur Skripte im selben Fenster/Frame ausgeführt werden
 - Reguliert AJAX Requests (Asynchrone Abfragen ohne Reload des Fensters)
 - Regelt NICHT dass im <script> Element fremde Scripts geladen werden dürfen (z.B. CDN)
 - Keinen Zugriff auf andere Funktionen des Clients außerdem des Browsers
 - Cross-Site Scripting (XSS) ist Problem beim Response auf dem Server und nicht erst auf dem Client

- JavaScript Serverseitig
 - Cross-Site Scripting (XSS) muss vermieden werden (Schadcode im Response)
 - Serverseitige Scripte (wie z.B. per NodeJS) haben Zugriff auf Datenbanken und Serversysteme
 - CPU-lastige Aufgaben können Server belasten

- Frameworks beinhalten vorgefertigte Funktionen
- Frameworks sind nicht JS typisch, sondern gibt es in fast jeder Programmiersprache
- Framework Einbindung über Datei
- Funktionen Verwendung per Aufruf
- Ermöglichen Wiederverwendbarkeit
- Verhindern Code Wiederholung
- Frameworks bilden z.B. auch Pattern/Paradigmen ab

- JS Frameworks bieten auch vollständige Trennung zwischen Frontend/Backend
 - ReactJS
 - NodeJS

- Bekannte JS Frameworks:
 - jQuery
 - AngularJS
 - Ext JS
 - ReactJS
 - emberJS

- Bekannteste JavaScript Bibliothek
- Ursprünglich von John Resig (2006)
- Nicht nur eine Bibliothek (jQuery Code)
- Ziel: Vereinfachung des Zugriffs auf das HTML DOM
- Funktionsumfang
 - Elementselektion (ähnlich CSS3)
 - DOM Manipulation
 - Animationen & Effekte (Accordeon, Slides etc.)
 - AJAX
 - Verwendung diverser Module wie z.B. jQuery UI
- Gründe für die Entwicklung
 - Fehlende Bibliothek für den heutigen Funktionsumfang
 - Minderung der LoC (Wiederverwendbarkeit von Code)
 - Verhindert im besten Fall Codewiederholung
 - Flexiblere Sprache für DOM Selektierung



```
$("#button").remove();
```

```
<div id="button"></div>  
<input type="submit" id="button2">
```

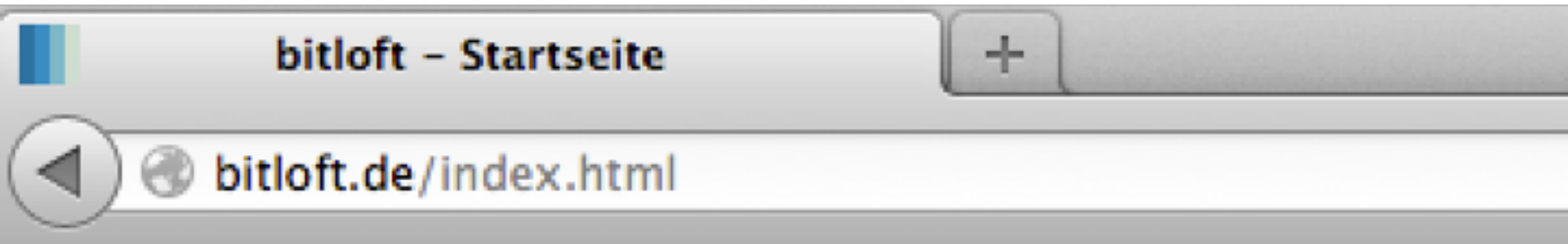
```
$("#button").click(function() {  
    $("#image").slideDown("slow");  
});
```

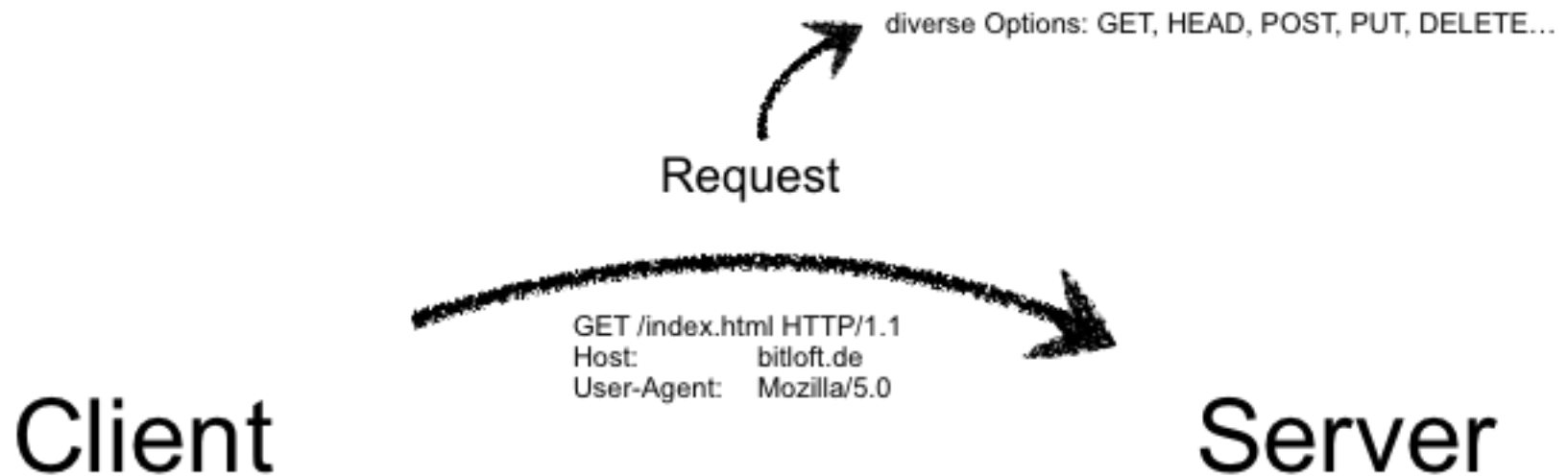
```
<div id="button">  
    Hier klicken  
</div>  

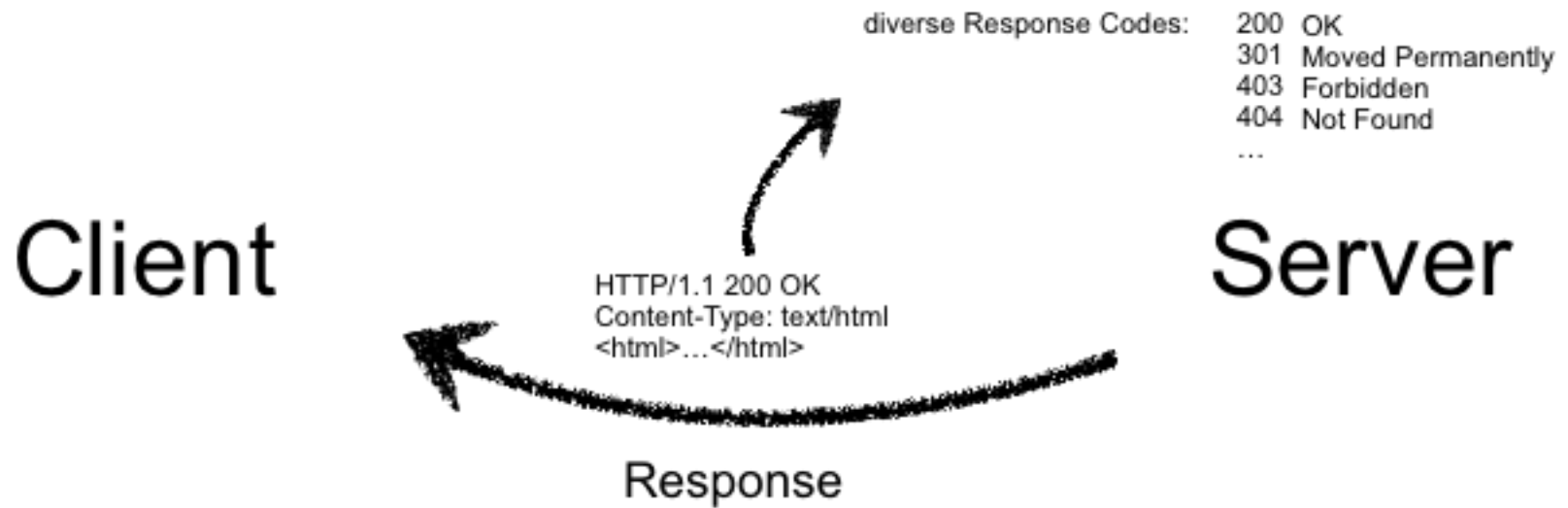
```



Client





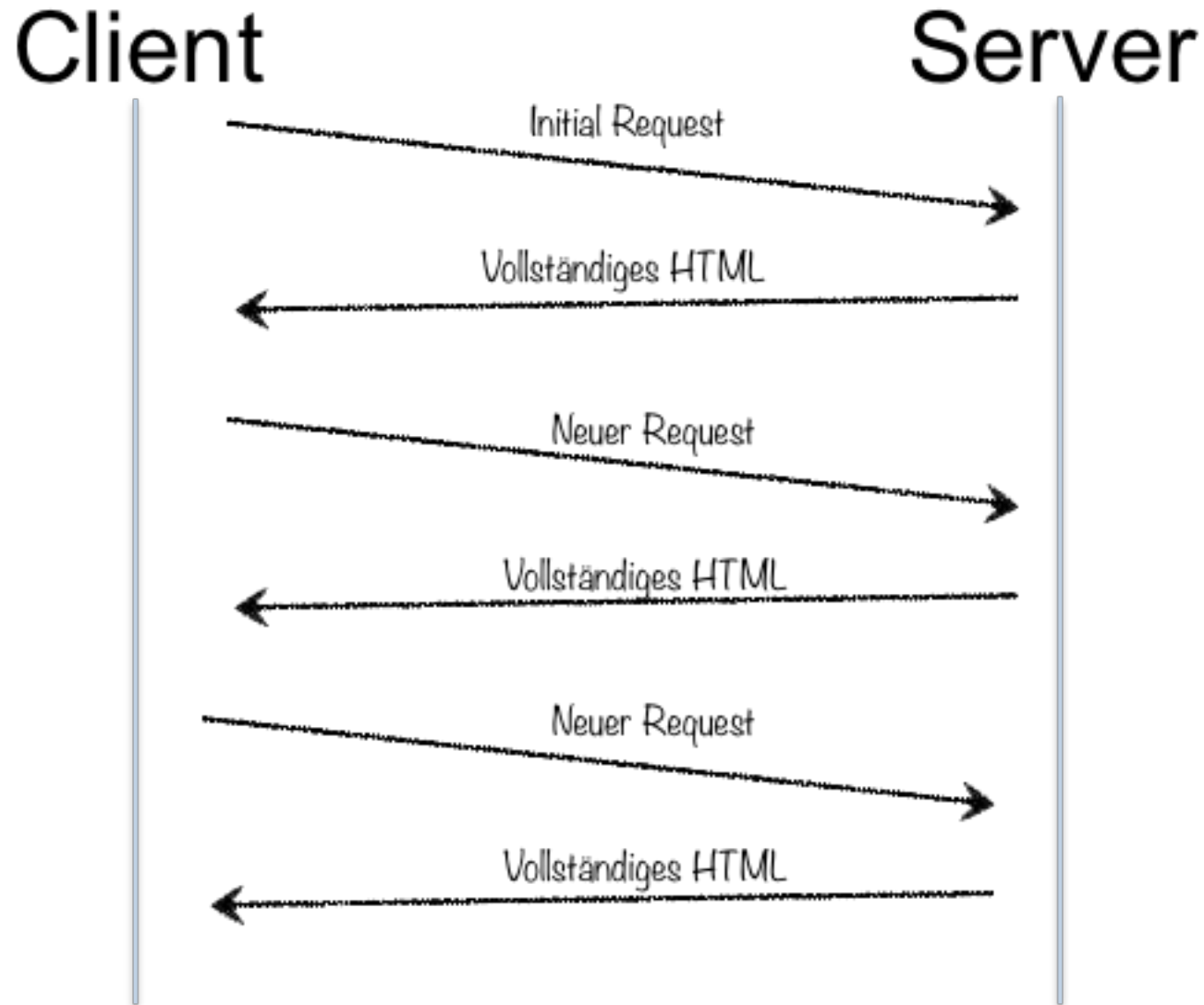


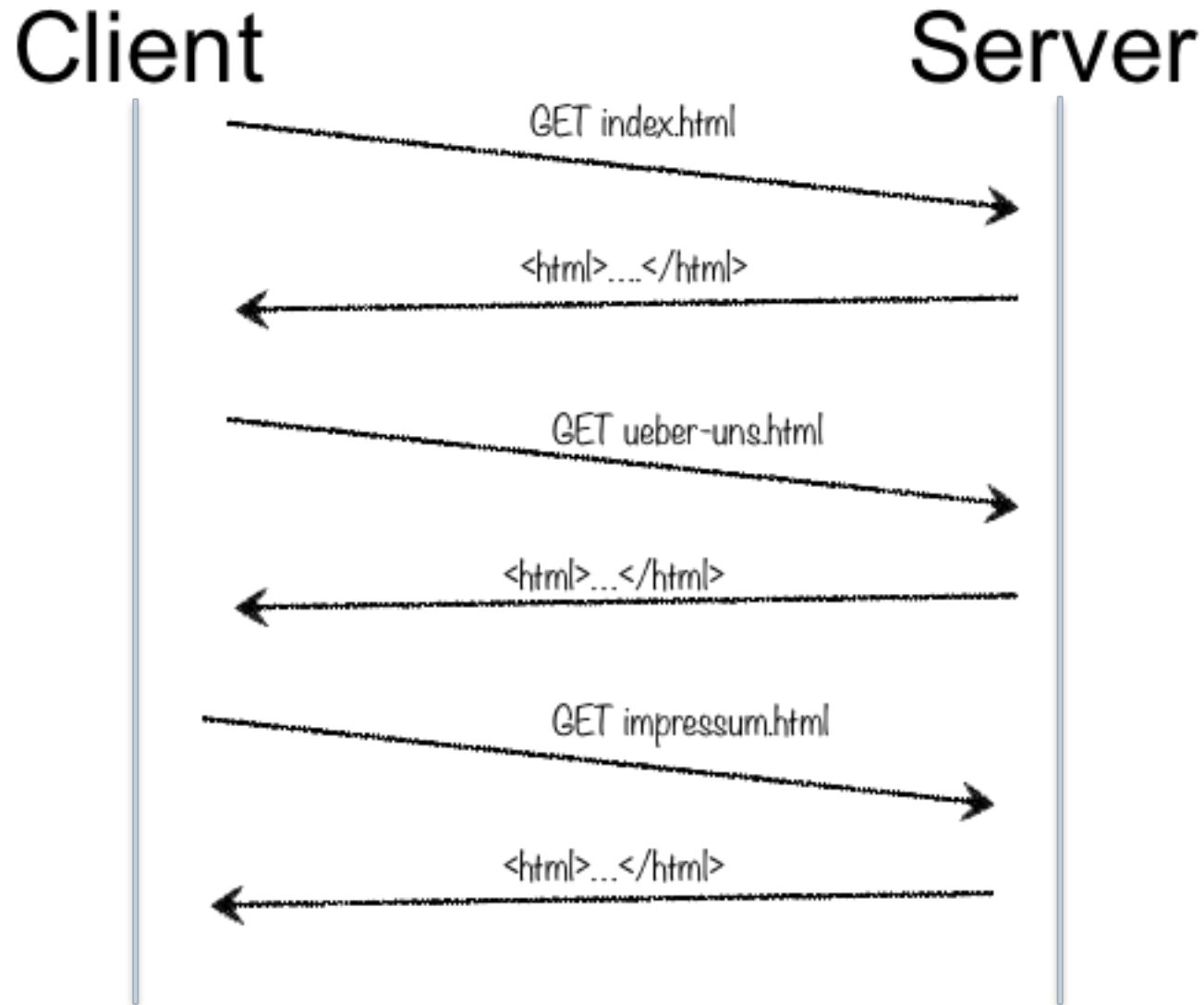
Response

URL	Status	Domain	Größe	Remote-IP	Zeitlinie
GET bitloft.de	200 OK	bitloft.de	24.2 KB	85.114.145.20:80	466ms
<div>Header Antwort HTML Cache Cookies</div> <div>Antwort-Header Quelltext anzeigen</div> <div> Connection Keep-Alive Content-Type text/html; charset=utf-8 Date Tue, 15 Apr 2014 11:52:01 GMT Keep-Alive timeout=5, max=100 Server Apache/2.2.16 (Unix) mod_ssl/2.2.16 OpenSSL/1.0.0 DAV/2 mod_mono/2.10 Set-Cookie fe_tipo_user=024dd30ceb58e38ff0e921109522d22a; path=/ Transfer-Encoding chunked </div>					

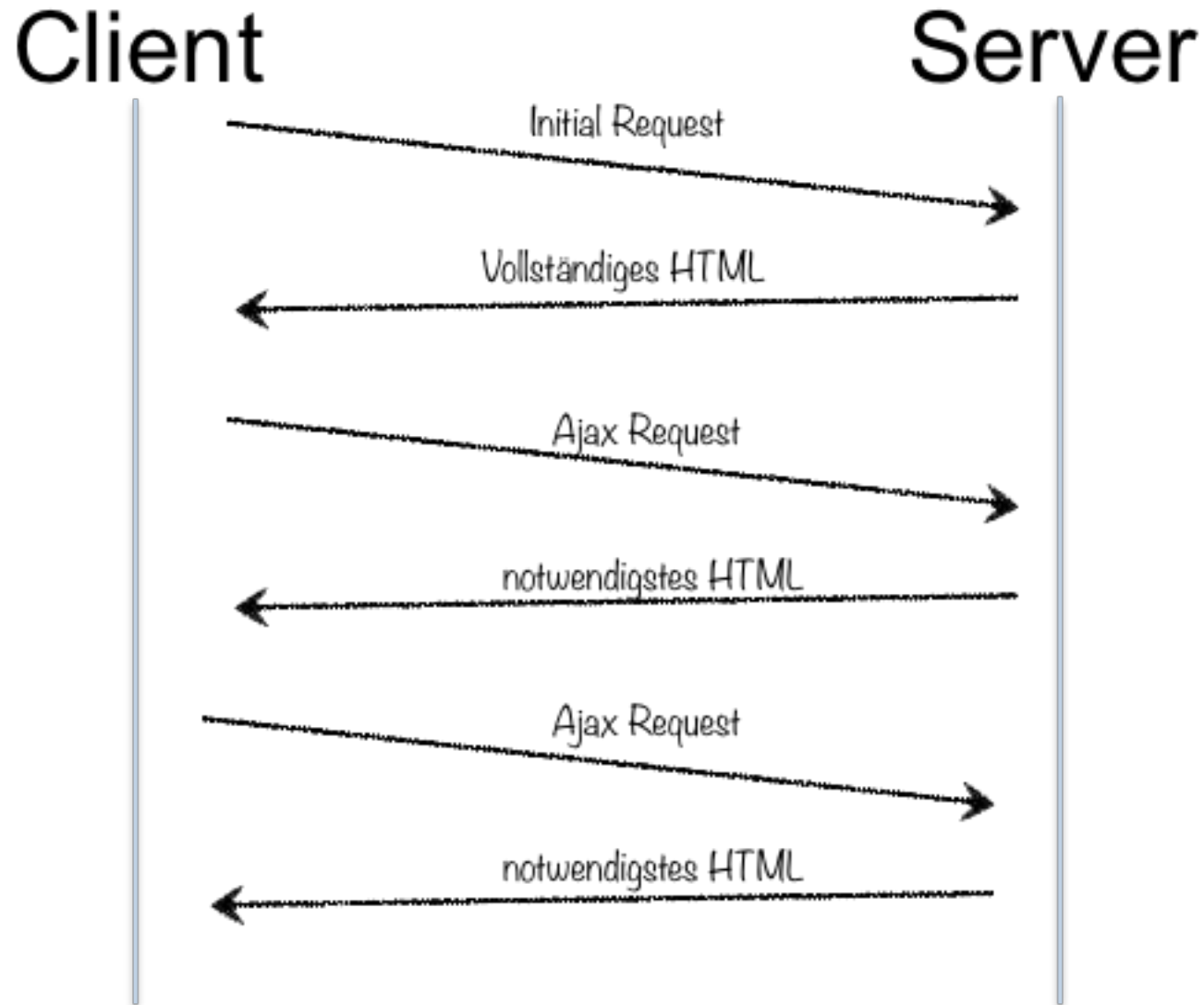
URL	Status	Domain	Größe	Remote-IP	Zeitlinie
GET bitloft.de	200 OK	bitloft.de	24.2 KB	85.114.145.20:80	466ms
<div>Header Antwort HTML Cache Cookies</div> <div> <pre> <!DOCTYPE html> <html lang="en"> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8"> <!-- This website is powered by TYPO3 - inspiring people to share! TYPO3 is a free open source Content Management Framework initially created by Kasper Skaarhoj and licensed under GNU/GPL. TYPO3 is copyright 1998-2013 of Kasper Skaarhoj. Extensions are copyright of their respective owners * Information and contribution at http://typo3.org/ --> <base href="http://www.bitloft.de"> <link rel="shortcut icon" href="http://bitloft.de/fileadmin/template/img/favicon.ico" type="image/x-icon" ; charset=binary"> <link rel="icon" href="http://bitloft.de/fileadmin/template/img/favicon.ico" type="image/x-icon; charset =binary"> <meta name="generator" content="TYPO3 6.0 CMS"> </pre> </div>					

Request / Response

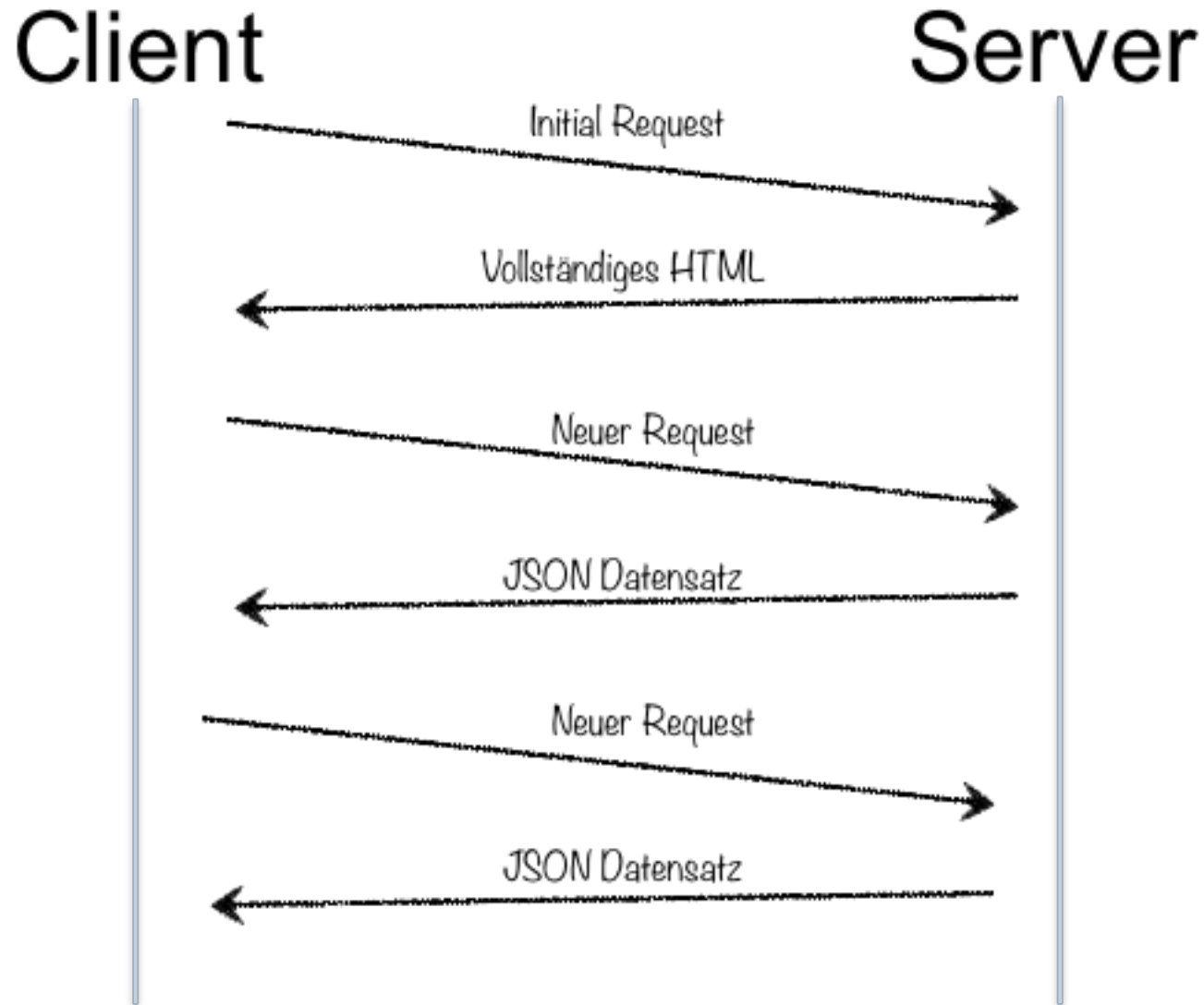


Request / Response

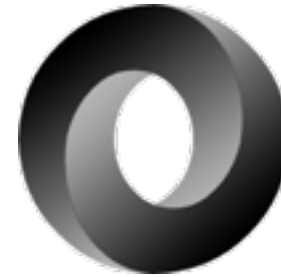
Request / Response



Request / Response

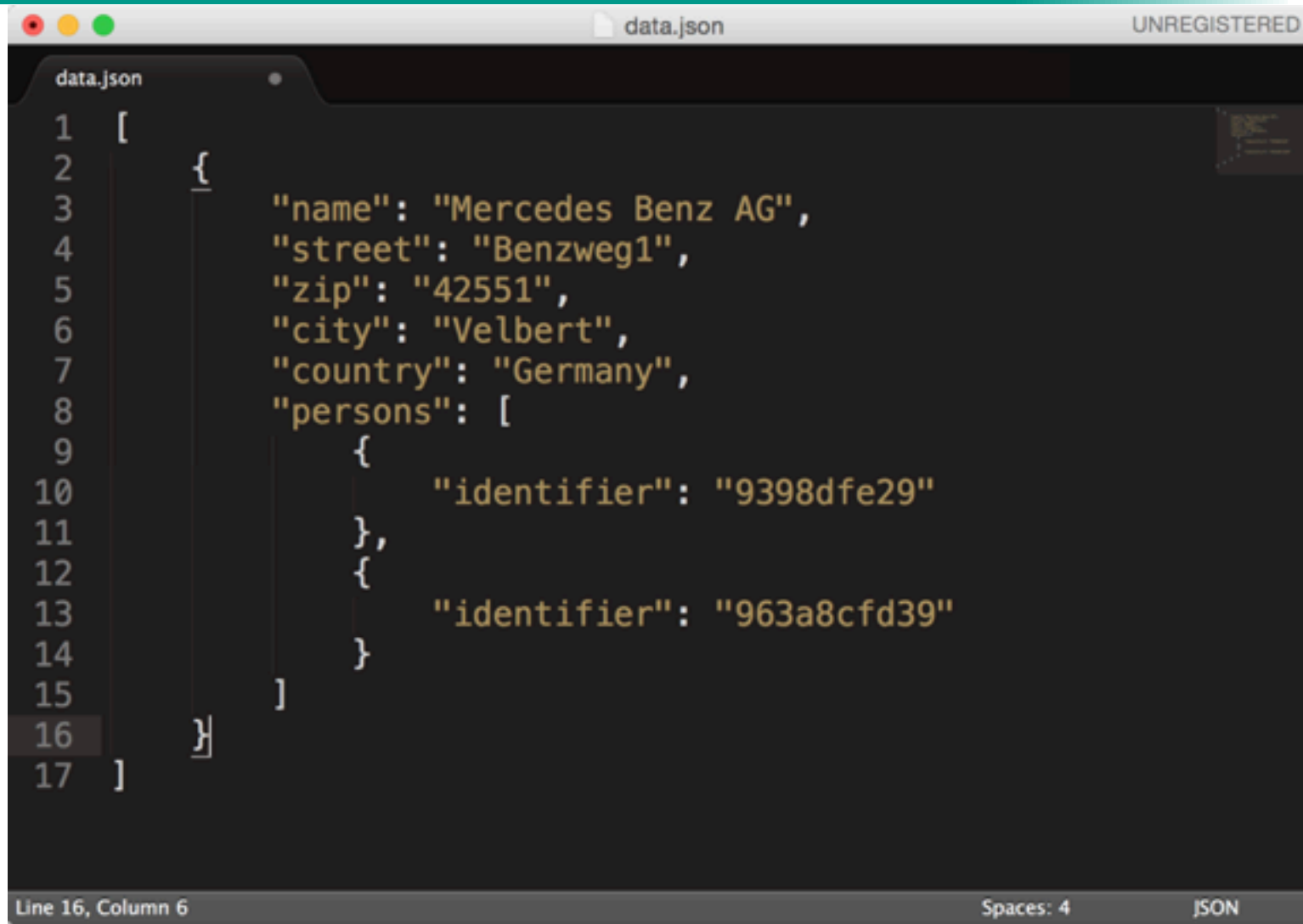


- Datenformat für Mensch & Maschine
- Unabhängig von der Programmiersprache
- Häufig im JS Umfeld eingesetzt
- Prädestiniert für asynchrone Übermittlung
- Einfacheres Format wie z.B. XML



Aufbau und grundlegende Funktionsweise der Webprogrammierung

JSON

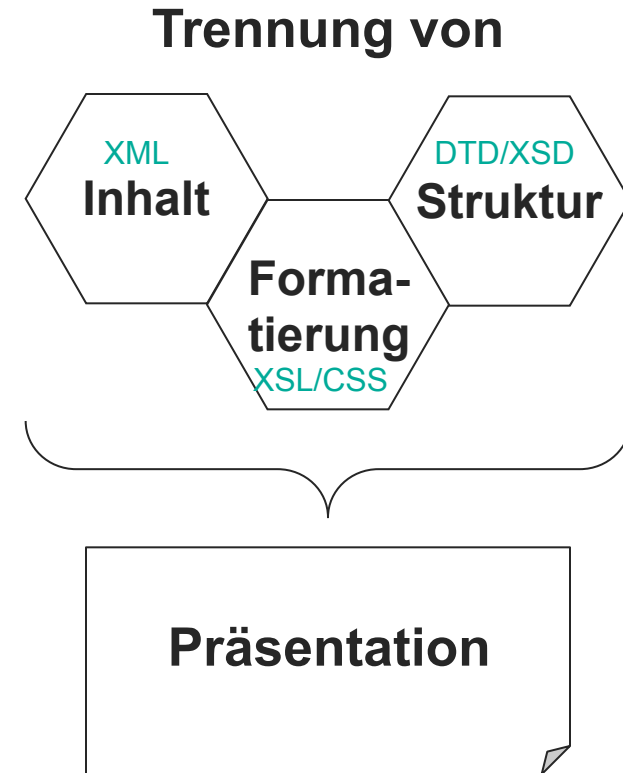


```
1  [  
2      {  
3          "name": "Mercedes Benz AG",  
4          "street": "Benzweg1",  
5          "zip": "42551",  
6          "city": "Velbert",  
7          "country": "Germany",  
8          "persons": [  
9              {  
10                 "identifier": "9398dfe29"  
11             },  
12             {  
13                 "identifier": "963a8cfd39"  
14             }  
15         ]  
16     }  
17 ]
```

Line 16, Column 6 Spaces: 4 JSON

XML

- **Untermenge von SGML**
(Standard Generalized Markup Language,
dt. Normierte Verallgemeinerte Auszeichnungssprache)
 - Vereinfachte Version von SGML
 - Für den Einsatz im WWW angepasst
 - Plattformunabhängig
 - Maschinenlesbar
 - Seit 1996 unter der Schirmherrschaft des World Wide Web Consortiums (W3C)
- XML ermöglicht
 - Beschreibung
 - Austausch
 - Darstellung und
 - Manipulation von strukturierten Daten



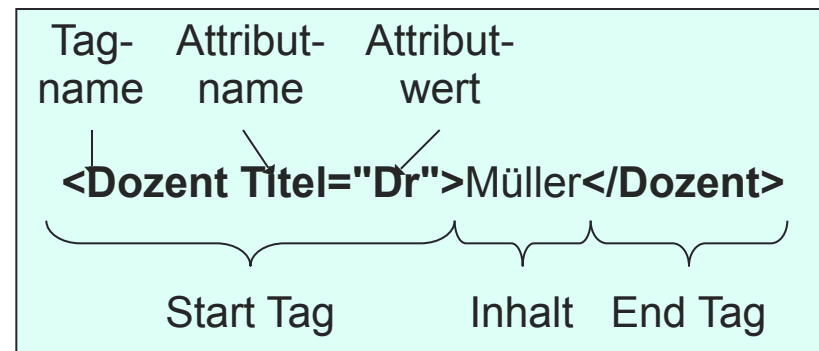
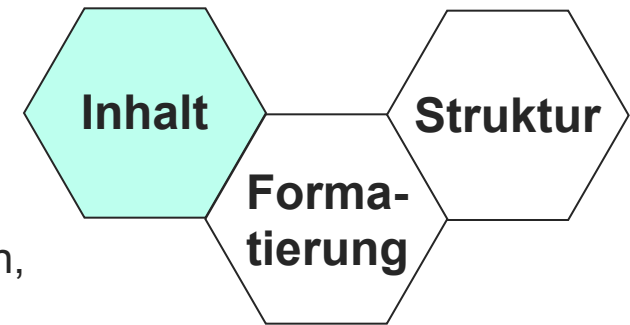
Extensible Markup Language (XML)

- **Metasprache**, mit der verschiedenste Markup-Sprachen definiert werden können
- **Interpretation und Präsentation** erfolgt durch separate Techniken, z.B.
 - Extensible Style Sheet Language (XSL)
 - Cascading Style Sheets (CSS)
- Für eine (individuelle) Auszeichnungssprache kann eine formale Grammatik angegeben werden, um nur **syntaktische korrekte Dokumente** zuzulassen

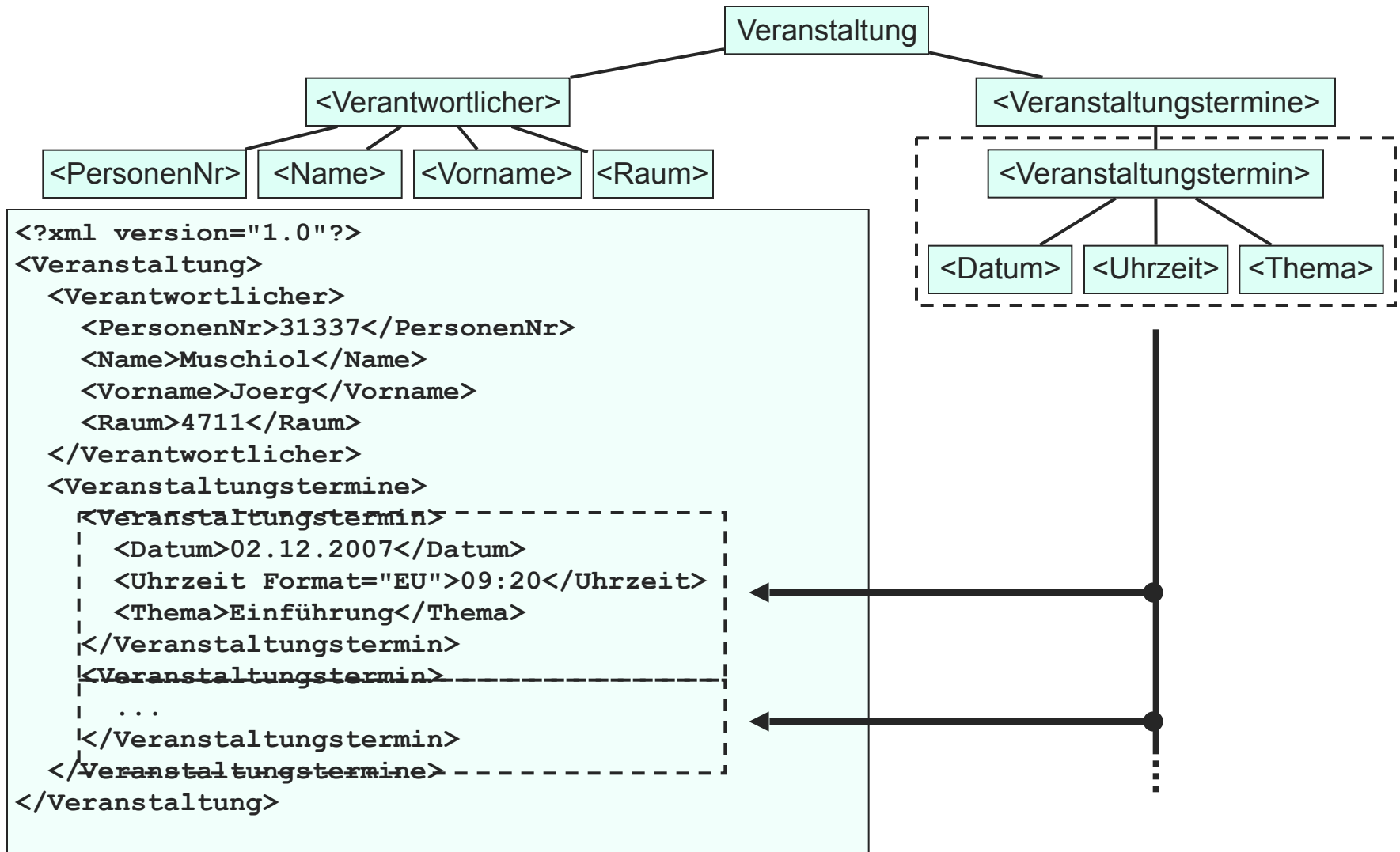
XML – Inhalt

- XML-Tags
 - Können beliebig, individuell, benannt werden
- XML-Syntaxregeln
 - XML-Dokument muss ein Wurzelement beinhalten, das alle anderen Elemente umfasst
 - Elemente müssen korrekt geschachtelt sein
 - Tags müssen immer geschlossen werden
 - Attributwerte immer in Anführungszeichen
 - Groß-/Kleinschreibung wird unterschieden

➤ **wohlgeformt**



```
<?xml version="1.0"?>
<Veranstaltung>
  <Name>Web Developement</Name>
  <Dozent Titel="Dipl-Inf">Muschiol</Dozent>
  <Dozent Titel="Dipl-Kfm">Meier</Dozent>
</Veranstaltung>
```



```
<?xml version="1.0"?>
<Kunde>
  <Kund-Nr>34</Kund-Nr>
  <Name>Bayer</Name>
  <Artikel />
  <Nick></Nick>
  <Adresse>
    <Str>Waldweg</Str>
    <Nr>12</Nr>
    <PLZ>39106</PLZ>
    <Ort>Magdeburg</Ort>
  </Adresse>
</Kunde>
```

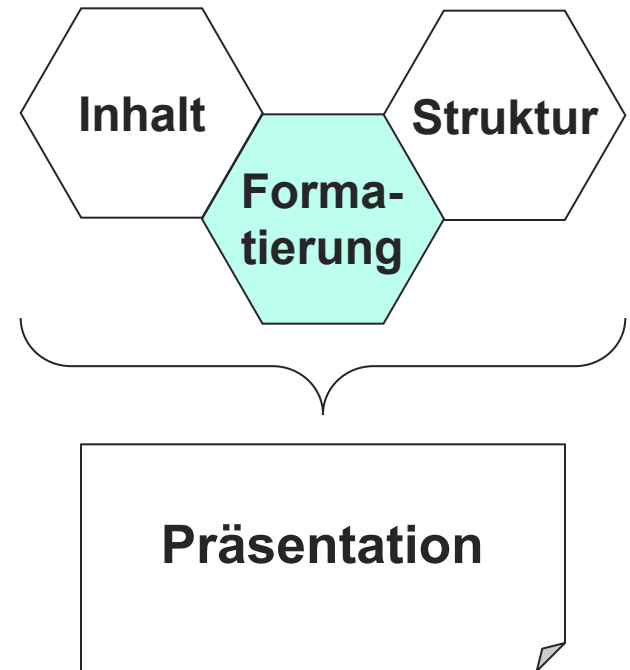
■ XML-Tag, Markup

■ Wohlgeformtheit

■ **NICHT wohlgeformt!**

■ Baumstruktur

- **Formatvorlagen** für XML-Dokumente
- **Trennung** von Anweisungen zur **Formatierung von Inhalt** und **Beschreibung der Struktur**
- Realisierung **verschiedener** Präsentationsformen durch Änderung oder Austausch der Formatvorlage



Aufbau und grundlegende Funktionsweise der Webprogrammierung


Beispiel: XML und CSS

```


xml.xml - Editor
Datei Bearbeiten Format Ansicht ?

<?xml version="1.0"?>
<?xml-stylesheet href="style.css" type="text/css"?>

<Kunden>
  <Kunde>
    <KdNr>1337</KdNr>
    <Name>Muschiol</Name>
    <Adresse>
      <Strasse>Konrad-Zuse-Str.</Strasse>
      <Nummer>10</Nummer>
      <PLZ>44801</PLZ>
      <Ort>Bochum</Ort>
    </Adresse>
  </Kunde>
  <Kunde>
    <KdNr>4711</KdNr>
    <Name>Meier</Name>
    <Adresse>
      <Strasse>Holsterhauser Str.</Strasse>
      <Nummer>123</Nummer>
      <PLZ>44625</PLZ>
      <Ort>Herne</Ort>
    </Adresse>
  </Kunde>
</Kunden>
  
```



Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe



http://folien.nf.de/folien/xml.xml

1337

Muschiol
Konrad-Zuse-Str. 10 44801 Bochum

4711

Meier
Holsterhauser Str. 123 44625 Herne

style.css - Editor
Datei Bearbeiten Format Ansicht ?

```

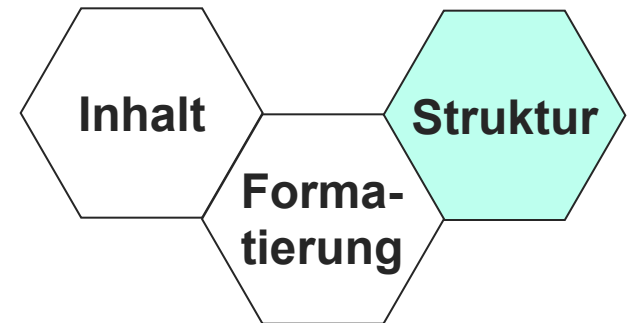
Kunde {
  font-size: 14pt;
  border: 1px solid red;
  display: block;
  margin-bottom: 10px;
}
KdNr {
  padding: 5px;
  background-color: #ccc;
  display: block;
}
Name {
  display: block;
  margin-left: 20px;
  font-weight: bold;
}
Adresse {
  display: block;
  margin-left: 20px;
}

```

29.09.2018

154

- Ein XML-Parser ist ein **Werkzeug**, um XML-Dokumente auf **Wohlgeformtheit** oder **syntaktische Korrektheit** bzgl. einer vorgegebenen **Grammatik** zu überprüfen
- Kann auch zur **Filterung** bestimmter Daten aus einem XML-Dokument eingesetzt werden



- DTD = Dokumentendefinition für XML-Dokumente
- Innerhalb einer DTD kann die Dokumentstruktur mit
 - Deklarationen von Elementtypen,
 - Attributlisten,
 - Entities und
 - Notationen
 definiert werden
- XML-Dokument mit interner DTD

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE hallo [
    <!ELEMENT hallo (#PCDATA)>
]>
<hallo>Hallo Welt!</hallo>
```


XML – Schema Description Language (XSD)

- XSD definiert die Struktur eines XML-Dokuments

- Struktur

- Ist ein XML-Dokument
- Schema ist immer Wurzelement
- Verwendung von Namensräumen
 - Für das Schema selbst
 - Für das Dokument

```
<?xml version="1.0"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://netzfactor.de"
  xmlns="http://netzfactor.de">
  <xsd:element name="Name">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:maxLength value="127" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:schema>
```

- Vorteile

- Schemata sind in XML geschrieben und können mit den selben Werkzeugen wie XML-Dokumente eingelesen werden
- Erweiterte Typdefinitionen für Elemente möglich (beliebige Strukturen)
- Angabe von Wertebereichen möglich

- Nachteile

- Sehr umfangreich und komplex!

Aufbau und grundlegende Funktionsweise der Webprogrammierung

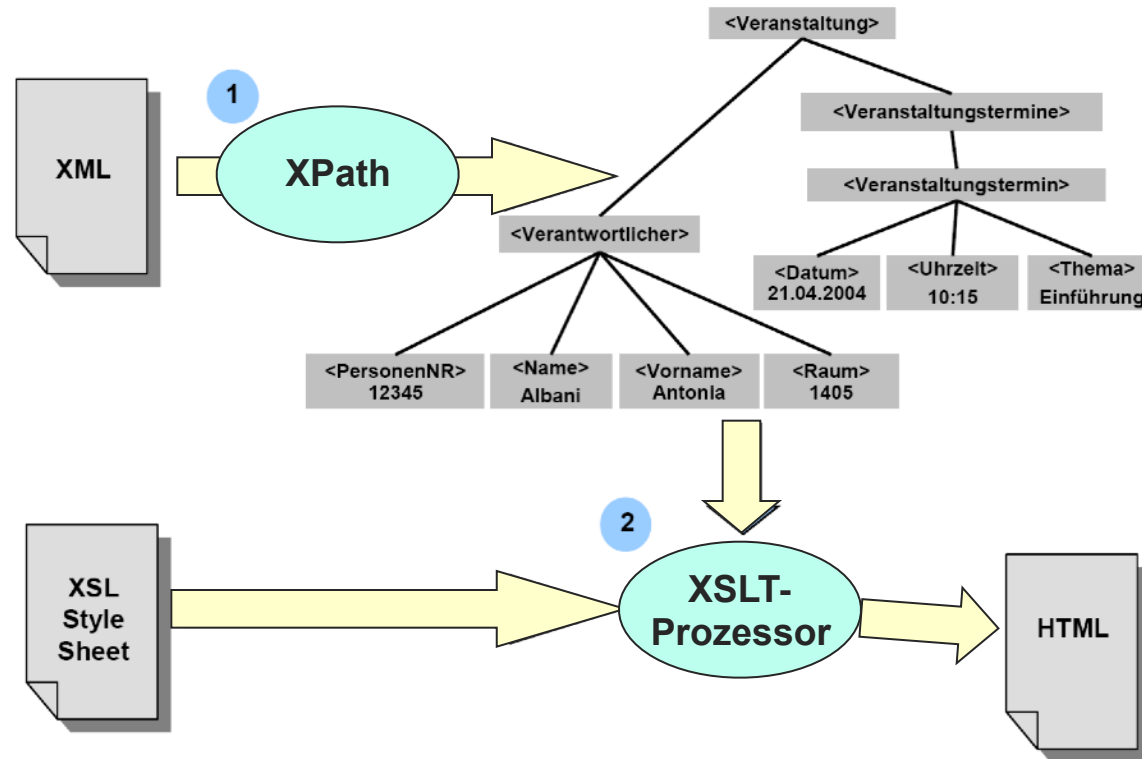
XML – Schema Description Language (XSD)

```

1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3    <xsd:element name="Veranstaltung">
4      <xsd:complexType>
5        <xsd:sequence>
6          <xsd:element name="Verantwortlicher">
7            <xsd:complexType>
8              <xsd:sequence>
9                <xsd:element name="PersonenNr" type="xsd:string" />
10               <xsd:element name="Name" type="xsd:string" />
11               <xsd:element name="Vorname" type="xsd:string" />
12               <xsd:element name="Raum" type="xsd:string" />
13             </xsd:sequence>
14           </xsd:complexType>
15         </xsd:sequence>
16       </xsd:complexType>
17     <xsd:element name="Veranstaltungstermine">
18       <xsd:complexType>
19         <xsd:sequence>
20           <xsd:element name="Veranstaltungstermin">
21             <xsd:complexType>
22               <xsd:sequence>
23                 <xsd:element name="Datum" type="xsd:string" />
24                 <xsd:element name="Uhrzeit" type="xsd:string" />
25                 <xsd:element name="Thema" type="xsd:string" />
26               </xsd:sequence>
27             </xsd:complexType>
28           </xsd:sequence>
29         </xsd:complexType>
30       </xsd:element>
31     </xsd:element>
32   </xsd:schema>

```

XML-Transformation / XSLT

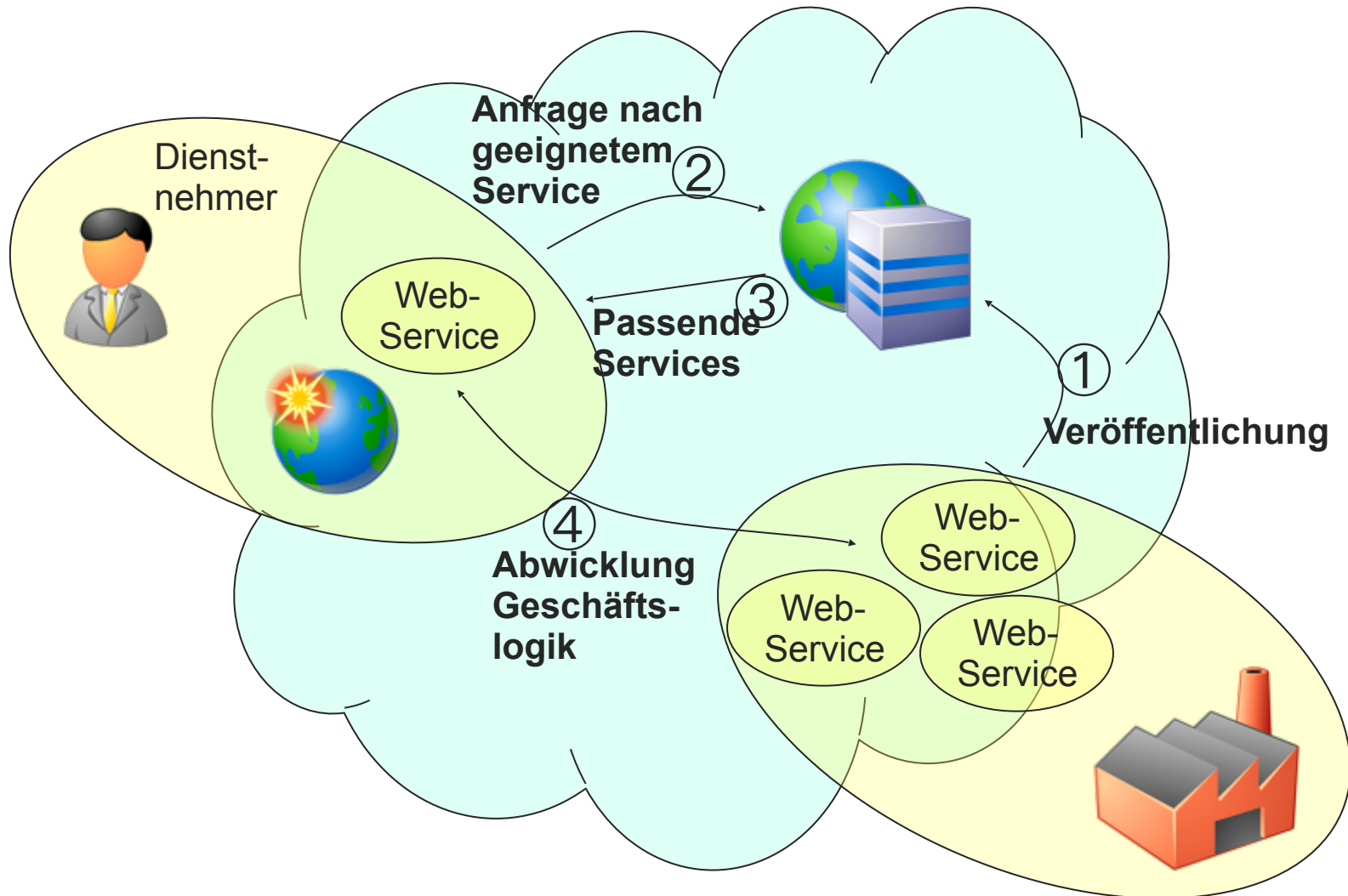


1. Darstellung des XML-Dokuments als hierarchischer Baum und Auswahl der gewünschten Elemente
(z.B.: Veranstaltung/Veranstaltungstermine/Veranstaltungstermin/Thema)
2. Transformation in HTML gemäß XSL-Formatvorlage

XML & Webservice

Aufbau und grundlegende Funktionsweise der Webprogrammierung

Web Services Überblick



Web Services Definition

Als Web Service wird eine:

- **zusammengehörende Menge** von
- **vermarktbaren Diensten** bezeichnet,
- die über das **World Wide Web** einem
- **autorisierten Nutzerkreis**
- unter Nutzung **standardisierter Kommunikationsprotokolle** über
- **wohldefinierte Schnittstellen** angeboten werden.

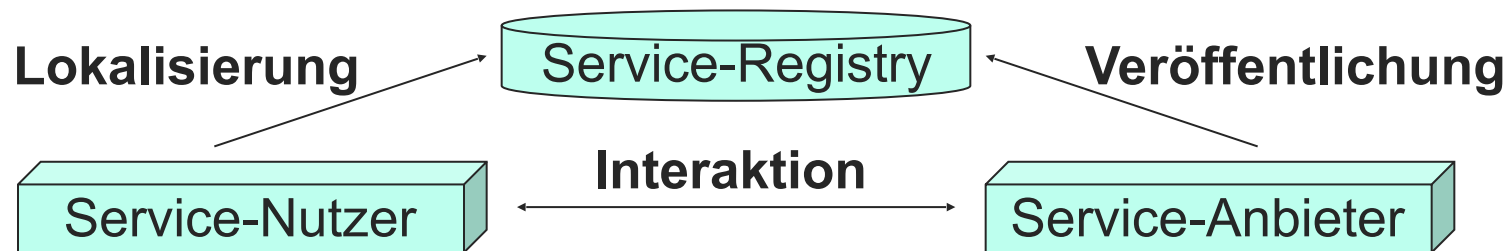
Implementierungsdetails bleiben dem Nutzer verborgen.

Web Services Architekturüberblick

- **Service-Anbieter**
 - Stellt seine Anwendungen als Dienste zur Verfügung
 - Ermöglicht standardisierten Zugriff auf seine Dienste
 - Macht Beschreibung seiner Dienste öffentlich bekannt

- **Service-Nutzer**
 - Nimmt Web Services als Dienste in Anspruch
 - Lokalisiert benötigte Dienste
 - Interagiert mit gefundenen Diensten (Zugriffsstandards)

- **Service-Registry**
 - Standardisierte Beschreibung der Dienste
 - Informationen über den Dienstanbieter
 - Informationen über den Zugriff auf den Dienst



- **XML als Basistechnologie**

- Interoperabilität in heterogenen, verteilten Umgebungen
- Austausch strukturierter und typisierter Dokumente
- Tools zur Auswertung und Prüfung der Validität vorhanden

- **Web Service Standards**

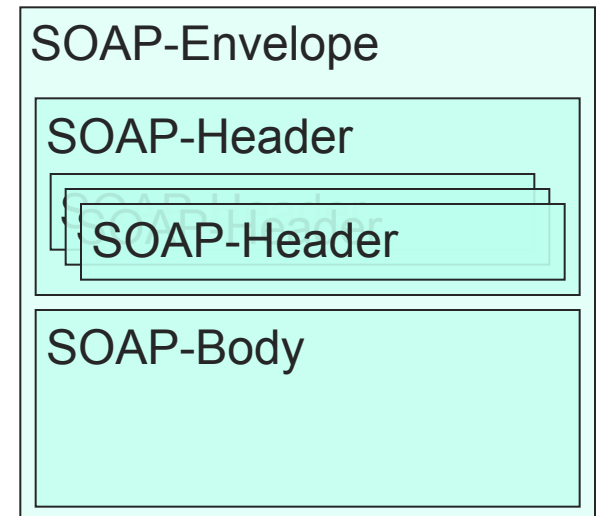
- Plattformunabhängiges Format für den Datenaustausch
 - **SOAP** (früher für "*Simple Object Access Protocol*")
- Beschreibung des Web Service
 - Web Service Description Language (**WSDL**)
- Rahmen zur Definition von Registries (Branchenverzeichnis)
 - Universal Description, Discovery and Integration (**UDDI**)

- **Breite Unterstützung durch Industrie und Standardisierungsorganisationen**
 - Microsoft, IBM, OASIS, W3C
 - Aktueller W3C-Standard ist SOAP 1.1

- **Erweiterbares, modulares Design**
 - Module: Syntax, Daten, Transport, Einsatz
 - Spezifikation enthält mögliche Implementierungen für Module
 - Daten
 - Transport:: Nachrichtenversand über HTTP
 - Einsatz: Übertragung von Daten, Aufruf von entfernten Prozeduren
 - Erweiterbarkeit
 - Beliebige andere Transportprotokolle (SMTP, MQSeries (Message Queueing), ...)
 - Attachments, Routing, Sicherheit, QoS, ...

SOAPNachrichtenaufbau

- **Envelope**
 - Klammert optionalen Header und Body
 - Stellt somit die SOAP-Nachricht dar
 - Verwendung von Namensräumen
- **Header**
 - Metainformationen der Nachricht
 - z.B. Transaktions-ID, Authentifizierung, Bedingungen ...
- **Body**
 - Enthält eigentliche Nutzdaten
 - Aufbau ist vom Nachrichtentyp abhängig
 - Methodenaufruf
 - Datenübertragung
 - Fehlermeldung



```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
```

SOAP-Nachricht im HTTP-Request

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

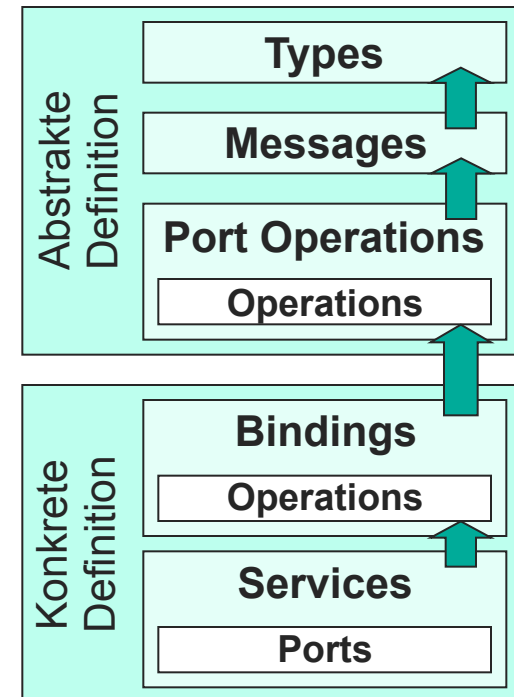
Problem: ?

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP-Nachricht im HTTP-Response

- **Schnittstellenbeschreibung** von Diensten
 - SOAP-Nachrichten, die mit einem Dienst kommunizieren, müssen seiner WSDL-Beschreibung entsprechen
 - Beschreibung ist Grundlage für die Nachrichtenerstellung

- **Types**
 - Definition der verwendeten Datentypen (meistens in XML-Schema)
 - Datentypen sind abstrakt und müssen noch denen des tatsächlich verwendeten Protokolls zugeordnet werden
- **Messages**
 - Part-Elemente werden jeweils mit definierten Types verbunden
 - Logische Nachrichten, die part-Elemente enthalten
- **Operations**
 - Beschreibt die Signatur von Funktionen mit Hilfe von Messages (input, output, fault)
 - Port Type umschließt mehrere Operations
- **Bindings**
 - Beschreibt das verwendete Nachrichtenformat und an welches Protokoll eine Operation gebunden wird
- **Ports**
 - Gibt eine protokollspezifische Adresse für das Binding an
 - Service umschließt mehrere Ports



Aufbau und grundlegende Funktionsweise der Webprogrammierung

WSDL Beispiel

```
<?xml version="1.0" ?>
- <definitions name="urn:GetQuote" targetNamespace="urn:xmlday-delayed-quotes"
  xmlns:tns="urn:xmlday-delayed-quotes" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/">
- <message name="GetQuoteRequest">
  <part name="symbol" type="xsd:string" />
</message>
- <message name="GetQuoteResponse">
  <part name="result" type="xsd:float" />
</message>
- <portType name="GetQuote">
  - <operation name="getQuote">
    <input message="tns:GetQuoteRequest" />
    <output message="tns:GetQuoteResponse" />
  </operation>
</portType>
- <binding name="GetQuoteBinding" type="tns:GetQuote">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  - <operation name="getQuote">
    <soap:operation soapAction="getQuote" />
    - <input>
      <soap:body use="encoded" namespace="urn:xmlday-delayed-quotes"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    - <output>
      <soap:body use="encoded" namespace="urn:xmlday-delayed-quotes"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
  </operation>
</binding>
- <service name="GetQuoteService">
  - <port name="GetQuote" binding="tns:GetQuoteBinding">
    <soap:address location="http://localhost:8080/axis/servlet/AxisServlet" />
  </port>
  - <port name="GetQuoteJava" binding="tns:GetQuoteBinding">
    <soap:address location="java:samples.stock.StockQuoteService" />
  </port>
</service>
</definitions>
```

UDDI - Universal Description Discovery and Integration

- Infrastruktur zum globalen Austausch von (Meta-) Informationen über Web Services
- Drei Kategorien von Informationen
 - White pages
 - Adressdaten, Kontaktinformationen über den Dienstanbieter
 - Yellow pages
 - Analog zu den "Gelben Seiten" eine Kategorisierung des Angebots
 - Green pages
 - Technische Information, Verweise auf die Spezifikation (z.B. im WSDLFormat), URL des Web Service
- Verzeichnisdienst selbst ist ein Web Service
 - Über SOAP-Nachrichten können Dienste registriert und die Registrierung administriert werden

- Kundennutzen
 - Markt- und Preistransparenz
 - Reduktion der Anzahl der Vertragspartner
 - Trusted third party = Portalbetreiber
 - Schaffung integrativer Mehrwerte
 - Angebot eines Leistungsbündels anstatt einer Menge von Einzelleistungen
 - Mehrwerte mit Effizienzwirkung
 - Verbesserung der Wirtschaftlichkeit, Kosten-, Zeitvorteile
- Erlösquellen
 - Kundenprofilierung
 - Cross Selling
 - Provisionen
 - Erschließen neuer Kundengruppen
 - Zusätzliche Margen durch vertikale Integration

7 **Praktische Demonstration einer webbasierten Programmiersprache** **z.B. PHP:**

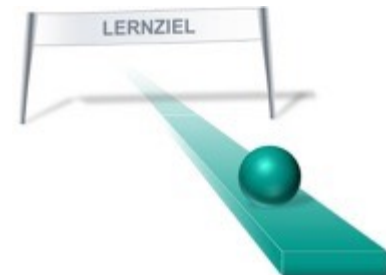
Praktische Demonstration einer webbasierten Programmiersprache z.B. PHP:

Modul-/Veranstaltungsgliederung

6.5.5	Web Service Basisarchitektur
6.5.6	Web Service Description Language (WSDL)
6.5.7	Simple Object Access Protocol (SOAP)
6.5.8	XML-RPC
7	Praktische Demonstration einer webbasierten Programmiersprache z.B. PHP
7.1	Laufzeitumgebungen
7.2	Allgemeine Syntax
7.3	Ausgabe
7.4	Skriptsprachenorientierte Programmierung
7.4.1	Typen und Datenstrukturen
7.4.2	Operatoren
7.4.3	Kontrollstrukturen
7.4.4	Funktionen
7.5	Request verarbeiten
8	Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress
8.1	Einrichtung von Wordpress
8.2	Backend und Frontend
8.3	Erstellen von Seiten, Blogbeiträgen und Inhalten

Im Anschluss an diesen Themenblock sollen Sie:

- die einzelnen Komponenten für die Entwicklung mit PHP kennen
- Anforderungen an die Entwicklungsumgebung nennen können
- Das Zusammenspiel von PHP und Webserver beschreiben können
- Allgemeine Syntaxregeln von PHP verstanden haben
- Die grundlegenden Bestandteile der Programmiersprache kennen
- Request und Response im Bezug auf PHP verstehen
- Die Verarbeitung von Daten aus Formularen verarbeiten können
- Die Verbindung von PHP zur Datenbank verstehen



- Hypertext Preprocessor urspr. **P**ersonal **H**ome **P**age Tools
- PHP ist eine Programmiersprache die serverseitig ausgeführt wird
 - Zum Webserver-Dienst wird der PHP Interpreter hinzu installiert
 - PHP Interpreter ermöglicht die Ausführung von PHP Dateien auf dem Server
 - Webserver liefert den Request auf Basis der PHP Datei an den Client aus
- PHP liegt aktuell in der Version 7.2 (stable) vor
- PHP ist eines der weitverbreitetsten serverseitigen Programmiersprachen im Internet
 - Facebook
 - Wikipedia
 - Flickr
 - Istockphoto
 - Yahoo



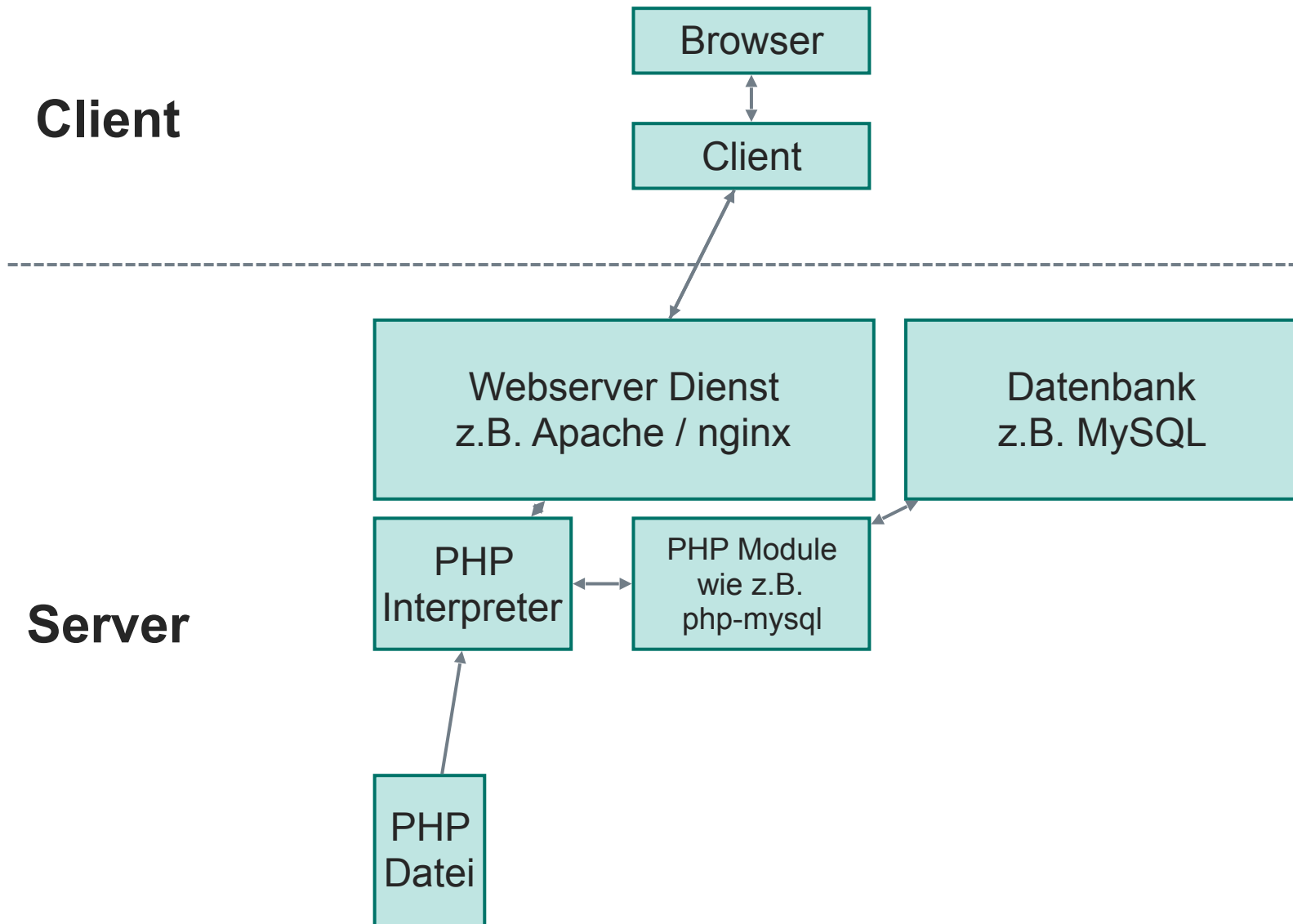
Laufzeitumgebungen

- Standardinterpreter mod-php führt die Dateien aus (ggf. Bottleneck)
- Php-fpm Interpreter (offizielles Paket) nutzt FastCGI für schnellere Ausführung (bis zu 3x schneller)
- HHVM (von Facebook) ist bis zu 3,5x schneller als php-fpm
- HHVM beinhaltet auch Architekturveränderungen
- PHP bietet Module zur Anbindung von Datenbanken
- Bekannte PHP Frameworks sind:
 - Zend
 - Laravel
 - Symfony
 - Flow
 - CakePHP
- PHP kann skriptorientiert, prozedural oder objektorientiert entwickelt werden



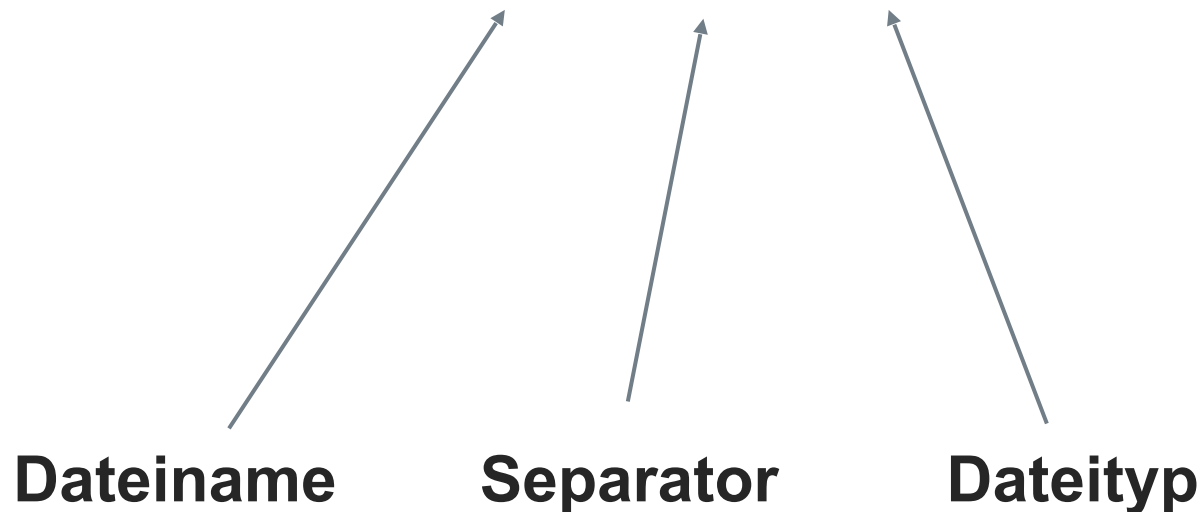
Praktische Demonstration einer webbasierten Programmiersprache z.B. PHP

Laufzeitumgebungen



Textdokument

index.php



Allgemeine Syntax

- PHP Programmiercode steht immer in `<?php ... ?>` Block
- Quellcodezeile endet immer mit „;“
- Kommentare sind mit `//` oder `/* ... */` möglich
- Variablendefinition entspricht schwacher Typisierung
- Fehlerausgabe/Debugging sollte aktiviert werden
- Per `php.ini` Datei kann PHP konfiguriert werden
- Vor Variable steht immer „\$“
- Texteditor, Notepad, PHPStorm, Atom, Notepad++ als lokale IDE (Integrated Development Environment)
- MAMP oder XAMP als Webserver, PHP Interpreter und Datenbank installieren
- Online Testumgebung: <https://wtools.io/php-sandbox>




PHP Block

```
<?php  
    [ PHP  Anweisung ]  
    [ PHP  Anweisung ]  
    [ PHP  Anweisung ]  
?>
```

```
<?php  
    [ PHP  Anweisung ]  
?>
```

PHP Block in HTML

```
<html>
  <head></head>
  <body>
    <?php
      [PHP Anweisung]
      [PHP Anweisung]
      [PHP Anweisung]
    ?>
  </body>
</html>
```

wtools.io

TOOLS

</> Sandbox

PHP popular

Paste Code

Generate

Check

Convert

Search tools

Q

Online PHP Sandbox

Form of PHP Sandbox

Enter Your PHP code here for testing/debugging in the **Online PHP Sandbox**, please include `<?php` tag at first:

1k?php

2echo "Hallo Welt!";

3echo 'Hallo Welt!';

4

5print "Hallo Welt! ";

6printf ("Hallo Welt! ");

7?>

8

9

Press F11 to switch to fullscreen mode

PHP Version

7.2.7

☒ Enable Error Reporting (`E_ALL`).

Execute

Clear

Your result can be seen below.

Save

Zuweisungsoperator

```
<?php  
    $liter = 14;  
?>
```

Ausgabe in PHP

```
<?php
    echo "Hallo Welt!";
    echo 'Hallo Welt!';

    print "Hallo Welt! ";
    printf ("Hallo Welt! ");
?>
```

Variablen

```
<?php
    $liter = 14;
    $preis = 1.35;
    $art = "Benzin";
    $zapfsaeulen_nummer = 1;
    $zapfsaeulen_nummer = 1;
    $zapfsaeule1 = "links";
    $zapfsaeule = "links";
    $!zapfsaeule = "links";
    $zapfsaeule = "links";

?>
```

Operatoren

Operator	Bedeutung
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
%	Modulo-Operation: der Rest einer ganzzahligen Division $7\%3 = 1$
**	Potenzieren $2 ** 3$ entspricht 2 hoch 3

kombinierter Zuweisungsoperator

```
<?php
```

```
$aufschlag = 5; // aufschlag ist 5
```

```
$aufschlag += 3; // aufschlag ist 8
```

```
$abschlag = 3; // abschlag ist 3
```

```
$abschlag -= 2; // abschlag ist 1
```

```
?>
```


Operatoren

```
<?php
```

```
    $liter = 14;  
    $preis = 1.35;
```

```
    $betrag = 14 * 1.35; // ohne Variablen  
    $betrag = $liter * preis; // mit
```

Variablen

```
    echo $betrag;  
    echo "Der Betrag ist: ". $betrag;  
    echo "Der Betrag ist $betrag";  
    echo 'Der Betrag ist $betrag';
```

```
?>
```

Konstanten

```
<?php
    const pi = 3.1415926;
    const anrede = "Herr";
    echo pi . "<br>";
    echo anrede;
?>
```

Operatoren Bedingungen

Operator	Bedeutung
==	gleich
!=	ungleich
>	größer als
<	kleiner als
>=	größer als oder gleich
<=	kleiner als oder gleich

Bedingungen

```
<?php
    $zahl = 3;
    $zahl2 = 5;

    if (3 < 5) {
        echo "Das ist richtig";
    }

    if ($zahl < $zahl2) {
        echo "Das ist richtig";
    }

?>
```

Bedingungen

```
<?php
    $zahl = 3;
    $zahl2 = 5;

    if ($zahl < $zahl2) {
        echo "Zahl ist kleiner Zahl2";
    } elseif ($zahl > $zahl2) {
        echo "Zahl ist größer Zahl2";
    } else {
        echo "Die Zahlen sind gleich";
    }

?>
```

Bedingungen

```
<?php
    $zahl = 3;
    $zahl2 = 5;

    if ($zahl == $zahl2) {
        echo "Die Zahlen sind gleich";
    }

    if ($zahl != $zahl2) {
        echo "Die Zahlen sind ungleich";
    }

?>
```

Rangordnung der Operatoren

Operator	Bedeutung
()	Klammern
!-	logisches Nicht, neg. Vorzeichen
* / %	Multiplikation, Division, Modulo-Operation
+ -	Addition, Subtraktion
< < = > >	Kleiner, kleiner oder gleich, größer größer oder gleich
== !=	gleich, ungleich
&&	logisches Und
	logisches Oder
=	Zuweisung

Schleifen

```
<?php
    $wiederholungen = 5;

    for ($i = 0; $i <= $wiederholungen; $i++)
    {
        echo $i . "<br>";
    }

?>
```


Schleifen

```
<?php
    $wiederholungen = 5;

    for ($i = 0; $i <= $wiederholungen; $i++)
    {
        echo $i . "<br>";
    }

?>
```

Arten von Schleifen

Operator	Bedeutung
for	Durchläuft entsprechend der Angaben im Kopf der Schleife
foreach	Durchläuft jeden Eintrag wie im Kopf der Schleife angegeben
while	Durchläuft solange bis die Bedingung im Kopf zutrifft (kopfgesteuert)
do-while	Durchläuft solange bis die Bedingungen im Fuß zutrifft (fußgesteuert)

Funktionen

```
<?php
    function addieren($zahl1, $zahl2) {
        return $zahl1 + $zahl2;
    }

    echo addieren(2,3);

?>
```

Funktionen

```
<?php
    function Begruessung($vorname, $nachname)
    {
        return "Guten Tag $vorname
$nachname";
    }

    $vorname = "Max";
    $nachname = "Mustermann";
    echo Begruessung($vorname, $nachname);

?>
```

Funktionen

```
<?php
    function Begruessung($vorname, $nachname) {
        global $anrede;
        return "Guten Tag $anrede $vorname
$nachname";
    }
    $anrede = "Herr";
    $vorname = "Max";
    $nachname = "Mustermann";
    echo Begruessung($vorname,$nachname);

?>
```

PHP Funktionen

```
<?php
    echo time();
    // siehe http://php.net/manual/de/
function.time.php

    echo round(10.54321,2);
    // siehe http://php.net/manual/de/
function.round.php
?>
```

Formulare verarbeiten

index.html

```
<body>
    <form action="eingabe.php"
method="post">
        <input type="text" name="Vorname">
        <input type="submit">
    </form>
</body>
```

Formulare verarbeiten eingabe.php

```
<?php
    echo "Guten Tag $_POST[ "Vorname" ] ";

    // Unterschied $_GET zu $_POST?
?>
```


PHP Übungsaufgabe

1. Richten Sie sich Ihre Entwicklungsumgebung ein bzw. rufen Sie <https://wtools.io/php-sandbox> auf
2. Entwickeln Sie ein PHP Programm, in dem Sie die vier Grundrechenarten jeweils per Funktionsaufruf durchführen können und lassen Sie sich das jeweilige Ergebnis ausgeben
3. Entwickeln Sie ein PHP Programm, in dem Sie ausgeben wie Sie heißen, Sie begrüßt werden, welcher Tag heute ist und wie die Uhrzeit lautet

8 Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress

Modul-/Veranstaltungsgliederung

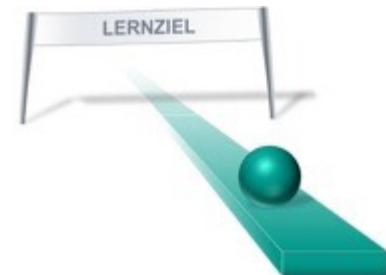
6.5.5	Web Service Basisarchitektur
6.5.6	Web Service Description Language (WSDL)
6.5.7	Simple Object Access Protocol (SOAP)
6.5.8	XML-RPC
7	Praktische Demonstration einer webbasierten Programmiersprache z.B. PHP
7.1	Laufzeitumgebungen
7.2	Allgemeine Syntax
7.3	Ausgabe
7.4	Skriptsprachenorientierte Programmierung
7.4.1	Typen und Datenstrukturen
7.4.2	Operatoren
7.4.3	Kontrollstrukturen
7.4.4	Funktionen
7.5	Request verarbeiten
8	Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress
8.1	Einrichtung von Wordpress
8.2	Backend und Frontend
8.3	Erstellen von Seiten, Blogbeiträgen und Inhalten

Modul-/Veranstaltungsgliederung

8.4	Verwalten von Assets
8.5	Themen Anpassungen
8.6	Kontaktformulare
8.7	Erstellen und Einbindung von HTML Elemente mit CSS
8.8	Plugin Funktionen

Im Anschluss an diesen Themenblock sollen Sie:

- Anhand einer praktischen Übung die Bedienung eines CMS für die Contentpflege beherrschen
- Verstehen wie Erweiterungen in CMS zu installieren sind
- Anpassungen am Layout vorgenommen werden können
- Wissen wie der Content-Workflow verfolgt wird
- Welche Rechtesysteme ein CMS beinhaltet
- Wissen wie ein Asset (Bilder, Videos etc.) Management erfolgt
- Die Publikation von Content im CMS kennen
- Die Bedienung von Wordpress kennen

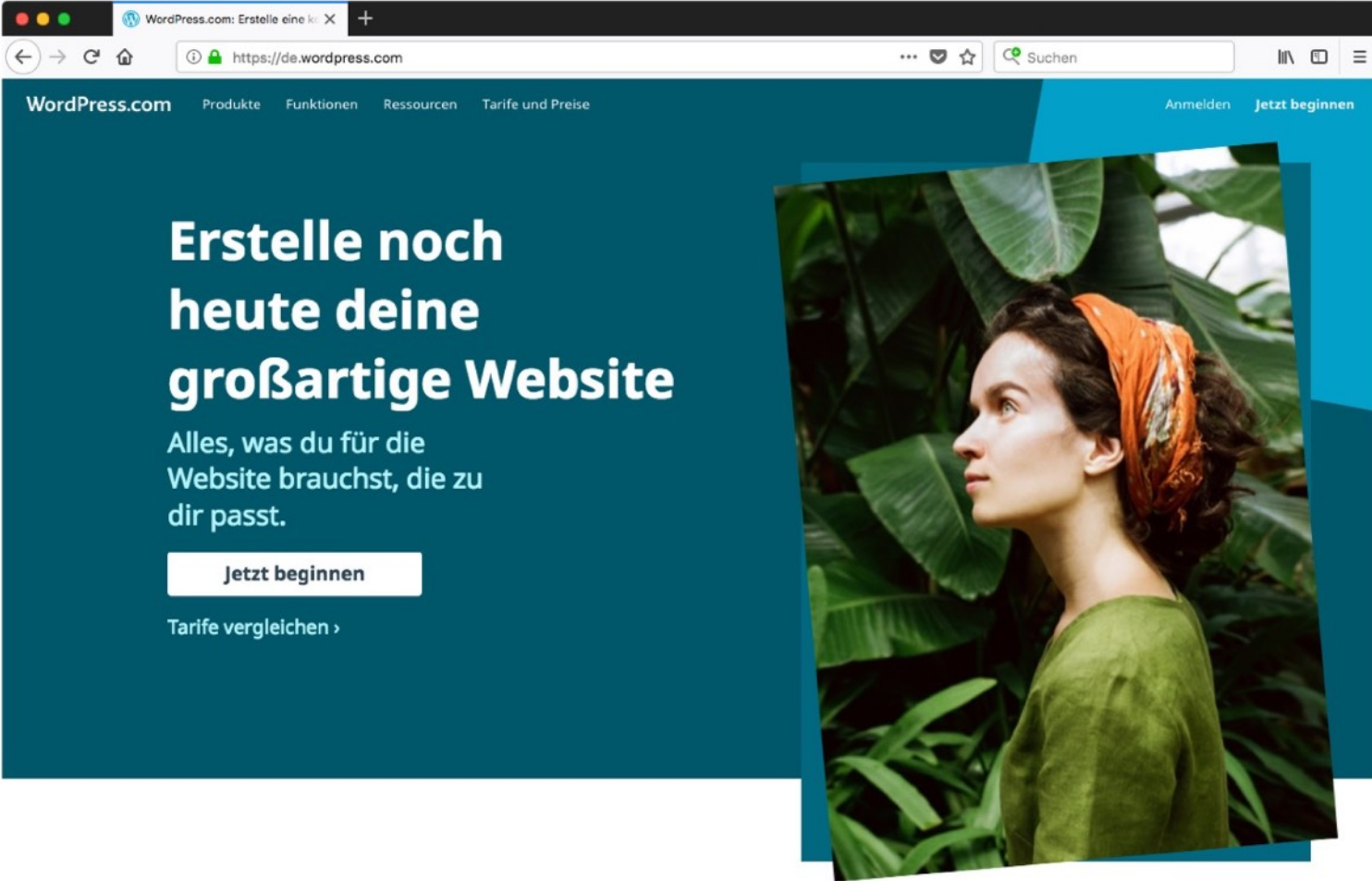


Konzept Wordpress

- Wordpress ist ein frei verfügbares Web Content-Management-System
- Verfügbar seit 2003
- Verbreitung durch über 70 Mio. Webseiten
- Unterliegt der GPL-2.0 Lizenz
- Aktuelle stable Version ist 4.9.x
- System kann auf Server installiert (**wordpress.org**) oder
- Cloud Instanz über **wordpress.com** erstellt werden
- Frontend = Ansicht für Besucher der Webseite
- Backend = Ansicht für Redakteure/Admin der Webseite
- Merkmale / Besonderheiten:
 - sehr schnelle Installation und Einrichtung (Server + Cloud)
 - Backend Bedienung ist sehr bekannt (durch die Verbreitung)
 - Große Bibliothek an Plug-ins
 - Große Bibliothek an Themes



Einrichtung einer Wordpress Instanz



WordPress.com: Erstelle eine Website

https://de.wordpress.com

WordPress.com Produkte Funktionen Ressourcen Tarife und Preise Anmelden Jetzt beginnen

Erstelle noch heute deine großartige Website

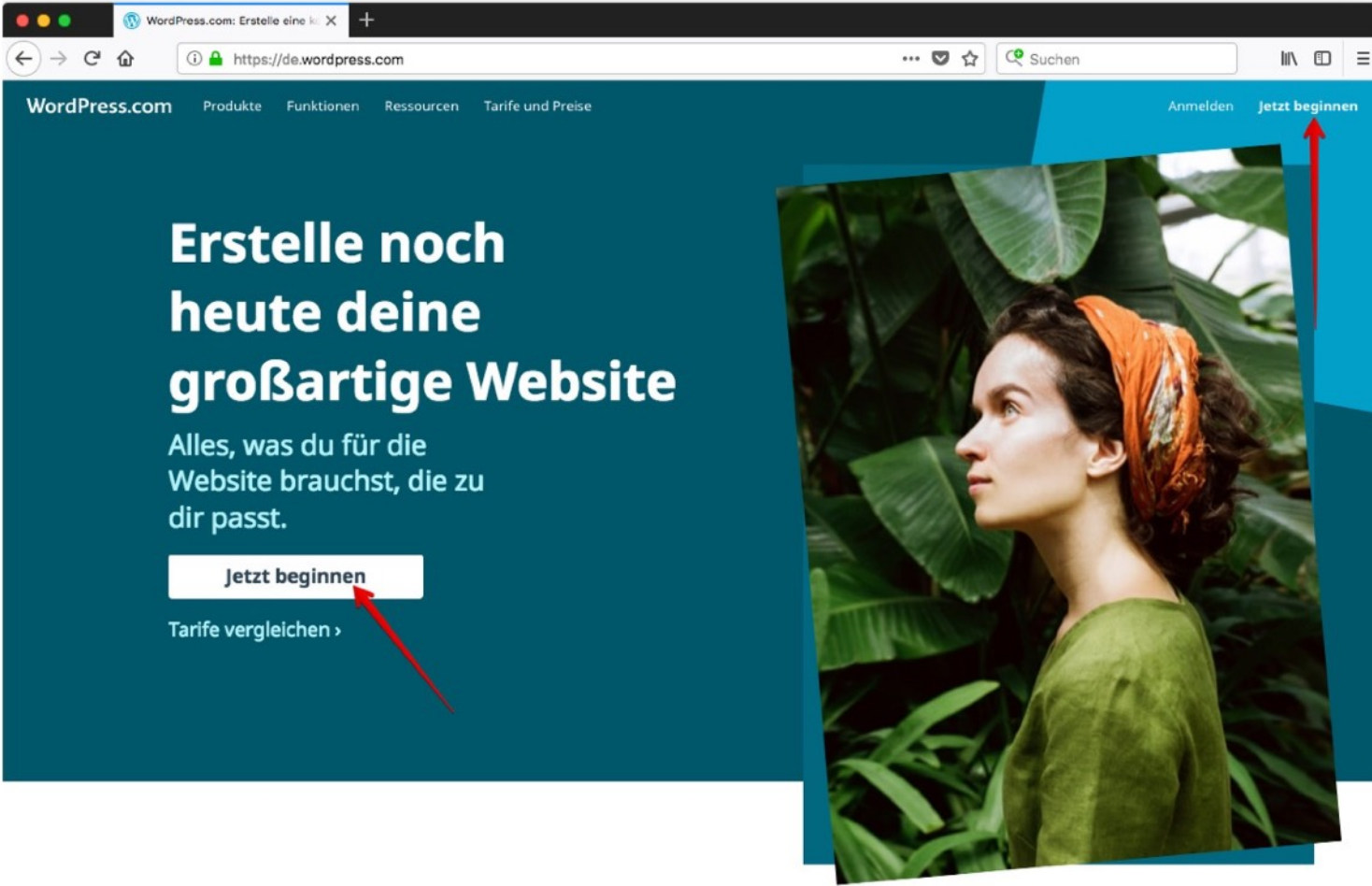
Alles, was du für die Website brauchst, die zu dir passt.

[Jetzt beginnen](#)

[Tarife vergleichen >](#)

Den Anfang macht eine Domain

Einrichtung einer Wordpress Instanz



WordPress.com Produkte Funktionen Ressourcen Tarife und Preise Anmelden **Jetzt beginnen**

Erstelle noch heute deine großartige Website

Alles, was du für die Website brauchst, die zu dir passt.

Jetzt beginnen

[Tarife vergleichen >](#)

Den Anfang macht eine Domain

Einrichtung einer Wordpress Instanz

WordPress.com Anmelden

Schritt 1 von 4

Erstellen wir eine Website

Beantworte bitte diese Fragen, damit wir dir helfen können, eine für dich passende Website zu erstellen.

Welchen Namen möchtest du deiner Website geben? ⓘ

Worum geht es auf deiner Website? ⓘ

Was möchtest du mit deiner Website erreichen?

☐ Teilen von Ideen, Erfahrungen, Neuigkeiten, Rezensionen, Geschichten, Videos oder Fotos

☐ Dein Unternehmen oder deine Organisation, deine Fähigkeiten oder Veranstaltungen bewerben

☒ Weiterbildung, Schulung oder Beratung anbieten

☐ Produkte verkaufen oder Zahlungen empfangen

☐ Dein Portfolio präsentieren

Wie erfahren bist du bei der Erstellung von Websites?

Anfänger 1 2 3 4 5 Experte

Fortfahren

Einrichtung einer Wordpress Instanz

WordPress.com Anmelden

Schritt 2 von 4

Gib deiner Website eine Adresse.

Gib den Namen deiner Website oder bestimmte Stichwörter ein, die diese beschreiben, damit es losgehen kann.

testfom

.com .net .org .blog .club .co .life .me Mehr Endungen ▾

testfom.com

In kostenpflichtigen Tarifen enthalten

Bester Treffer

- ✓ Genaue Übereinstimmung
- ✓ „.com“ ist die gängigste Endung

Auswählen

demofof.blog

In kostenpflichtigen Tarifen enthalten

Beste Alternative

- ✓ „.blog“ ist eine gängige Endung

Auswählen

testfom54707727.wordpress.com	Free	Auswählen
testfom.blog	In kostenpflichtigen Tarifen enthalten	Auswählen
testfom.org	In kostenpflichtigen Tarifen enthalten	Auswählen
demofof.com	In kostenpflichtigen Tarifen enthalten	Auswählen
testfom.net	In kostenpflichtigen Tarifen enthalten	Auswählen

Einrichtung einer Wordpress Instanz

WordPress.com

Anmelden

Schritt 3 von 4

Such dir den richtigen Tarif aus.

Free	Persönlich	Premium	Business
Ideal für Studenten	Ideal für Hobbyisten	Ideal für Profis	Ideal für Marken
<p>€0 lebenslang</p> <p>Beginne mit Free</p> <ul style="list-style-type: none"> ✓ Community-Support ✓ WordPress.com Subdomain ✓ Dutzende kostenlose Themes 	<p>€4 /Monat, jährliche oder zweijährliche Abrechnung</p> <p>Beginne mit Persönlich</p> <ul style="list-style-type: none"> ✓ Kostenlose individuelle Domain ✓ Support per E-Mail und im Live-Chat ✓ Alle kostenlosen Funktionen 	<p>€8 /Monat, jährliche oder zweijährliche Abrechnung</p> <p>Beginne mit Premium</p> <ul style="list-style-type: none"> ✓ Verdiane auf deiner Website Geld mit Werbung ✓ Unbegrenzte Premium-Themes ✓ Alle Persönlich-Funktionen 	<p>€25 /Monat, jährliche oder zweijährliche Abrechnung</p> <p>Beginne mit Business</p> <ul style="list-style-type: none"> ✓ Themes und Plugins hochladen ✓ Erweiterte SEO-Werkzeuge ✓ Alle Premium-Funktionen

Sicheres Bezahlen mit:

← Zurück

Einrichtung einer Wordpress Instanz

WordPress.com Anmelden

Schritt 4 von 4

Erstelle dein Konto.

Deine E-Mail-Adresse

Wähle einen Benutzernamen

Wähle ein Passwort

Durch das Erstellen eines Kontos über eine der folgenden Optionen stimmst du den Geschäftsbedingungen zu.

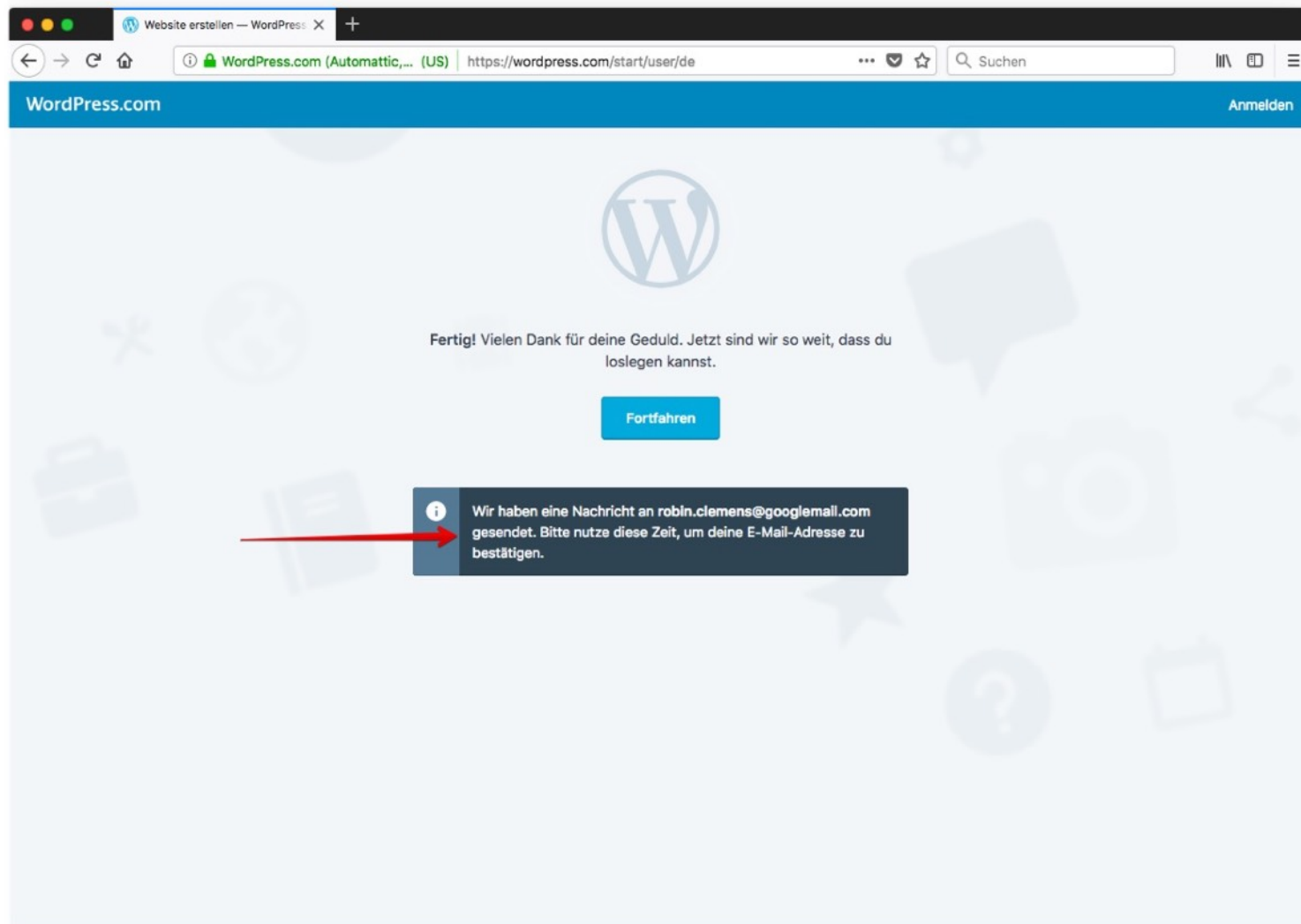
[Fortfahren](#)

Oder verbinde dein bestehendes Profil, um schneller loslegen zu können.

Weiter mit Google

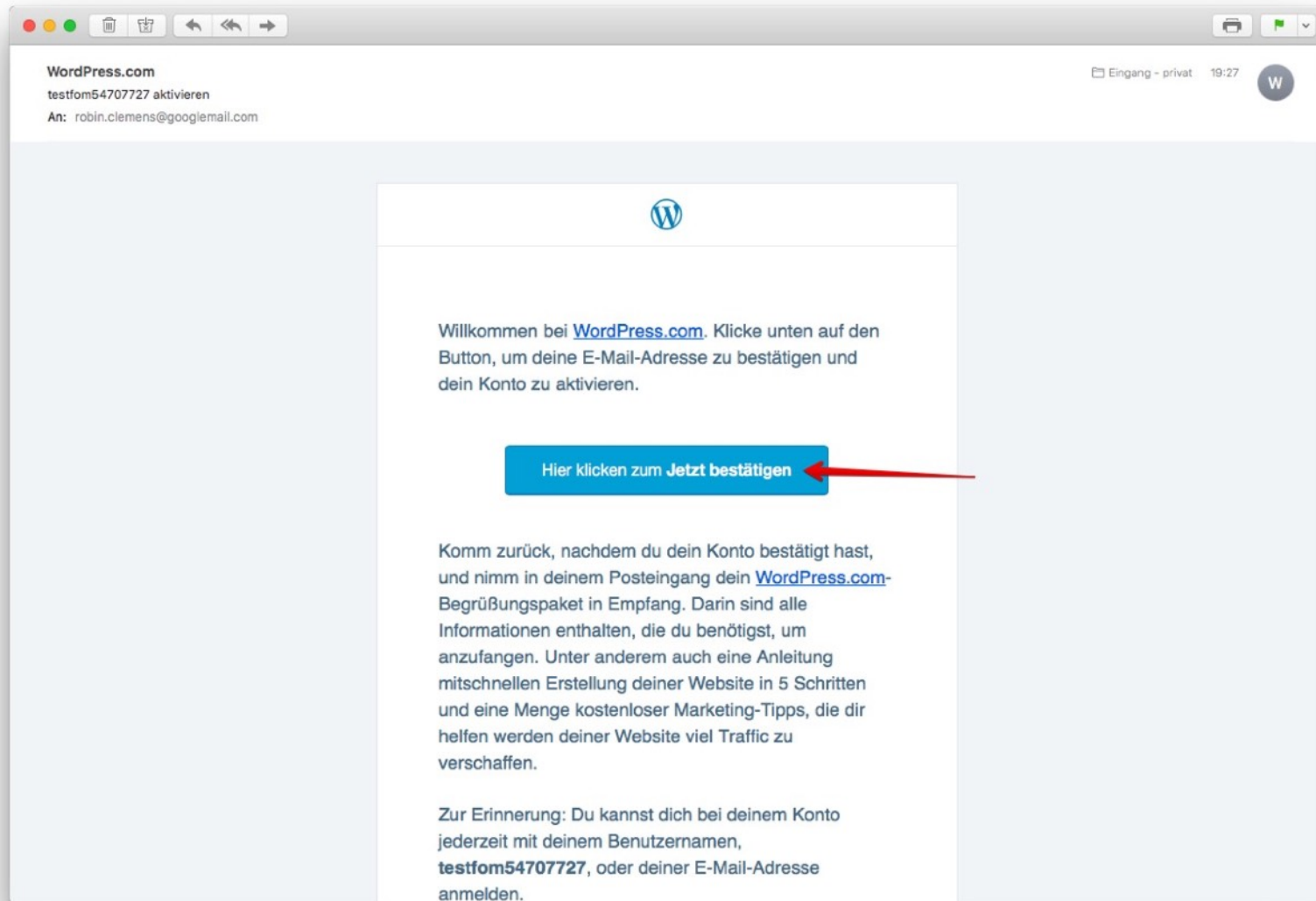
[← Zurück](#)

Einrichtung einer Wordpress Instanz



Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress

Einrichtung einer Wordpress Instanz



Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress

Einrichtung einer Wordpress Instanz

The screenshot shows the WordPress.com dashboard for a new website. The main heading is "Deine Website wurde erstellt." (Your website was created.) with a celebratory graphic. Below this, a message states: "Du hast deine Website erstellt. Jetzt ist es an der Zeit, sie mit der Welt zu teilen. Wir haben eine Liste mit Inhalten vorbereitet, mit deren Hilfe du dein Ziel schnell erreichst." (You have created your website. Now it is time to share it with the world. We have a list of content prepared to help you reach your goal quickly.)

On the left sidebar, there is a navigation menu with options like "Website ansehen", "Statistik", "Tarif", "Seiten", "Blogbeiträge", "Mediathek", "Kommentare", "Feedback", "Plugins", "Daten importieren", "Anpassen", and "Neue Website hinzufügen".

The main content area displays a "Deine Setup-Liste" (Your Setup List) with 3/9 items. The first three items are completed with green checkmarks:

- ✓ Du hast deine Website erstellt
- ✓ Du hast eine Website-Adresse ausgewählt
- ✓ Du hast deinen Website-Titel aktualisiert

The next two items are pending:

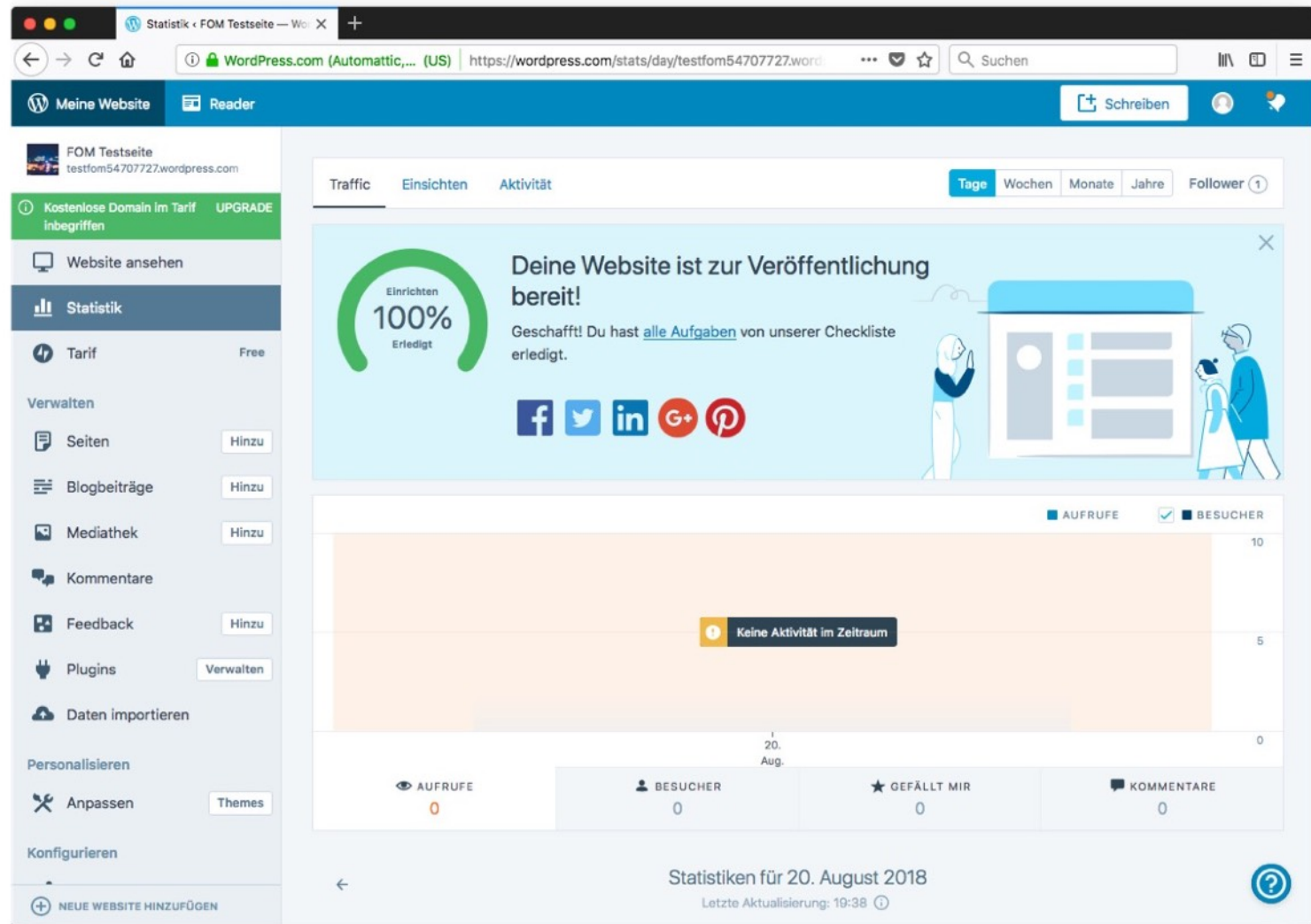
- ☐ Lade ein Website-Icon hoch
Sorge dafür, dass Besucher deine Website in Browser-Tabs wiedererkennen – wie das W von WordPress.com!
Geschätzte Dauer: 1 Minute
- ☐ Erstelle einen Untertitel
Gewinne das Interesse der Leser mit ein wenig mehr Informationen zu deiner Website.
Geschätzte Dauer: 2 Minuten

Each pending item has a "Los geht's!" (Let's go!) button. The bottom item, "Lade dein Profilbild hoch", is partially visible.

Übungsaufgaben Wordpress

- 1. Richten Sie Ihre eigene Wordpress.com Instanz ein**
- 2. Bearbeiten Sie die notwendigen Schritte (9 von 9) die Ihnen im Dashboard angezeigt werden (per Aisstent)**
 - 1. Icon hochladen (Musterbild aus Vorlagen verwenden)**
 - 2. Erstellen Sie einen Untertitel Ihrer Webseite**
 - 3. Laden Sie ein Profilbild hoch**
 - 4. Personalisieren Sie Ihre Kontaktseite in dem Sie Ihre Kontaktdaten angeben und erledigen Sie den Punkt in der Checkliste**
 - 5. Veröffentlichen Sie Ihren ersten Blogbeitrag**
 - 6. Öffnen Sie die Vorschau Ihres Blogbeitrages**
 - 7. Erledigen Sie den Punkt „Registriere eine individuelle Domain“ OHNE eine Domain zu registrieren (Klick auf die Checkbox in der Checkliste)**

Wordpress Backend Oberfläche



Wordpress Frontend Oberfläche

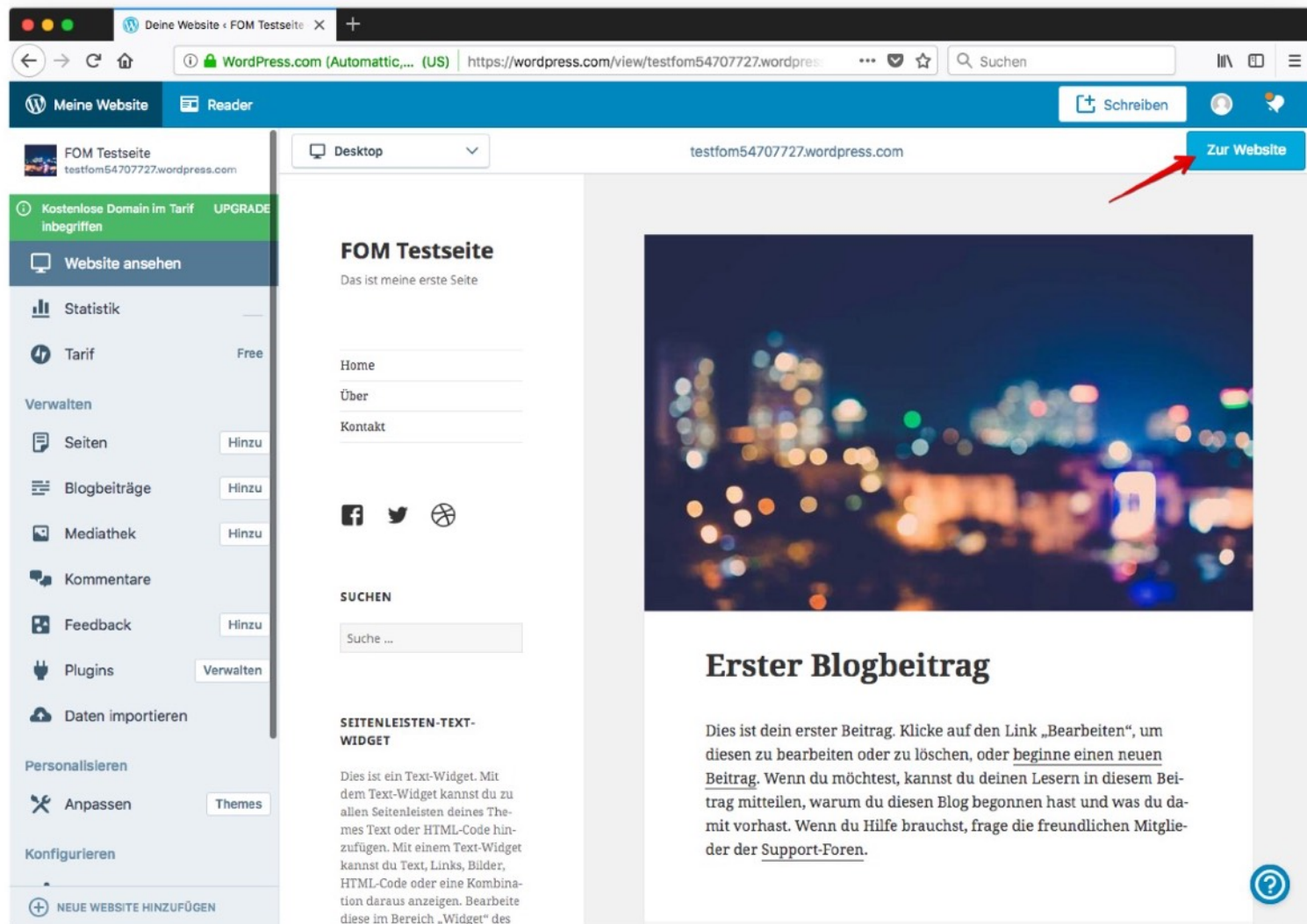
The screenshot shows the WordPress.com dashboard for a test site. The left sidebar contains a menu with the following items: 'Website ansehen', 'Statistik' (highlighted with a red arrow), 'Tarif', 'Verwalten' (containing 'Seiten', 'Blogbeiträge', 'Mediathek', 'Kommentare', 'Feedback', 'Plugins', 'Daten importieren'), 'Personalisieren' (containing 'Anpassen'), and 'Konfigurieren' (containing 'NEUE WEBSITE HINZUFÜGEN').

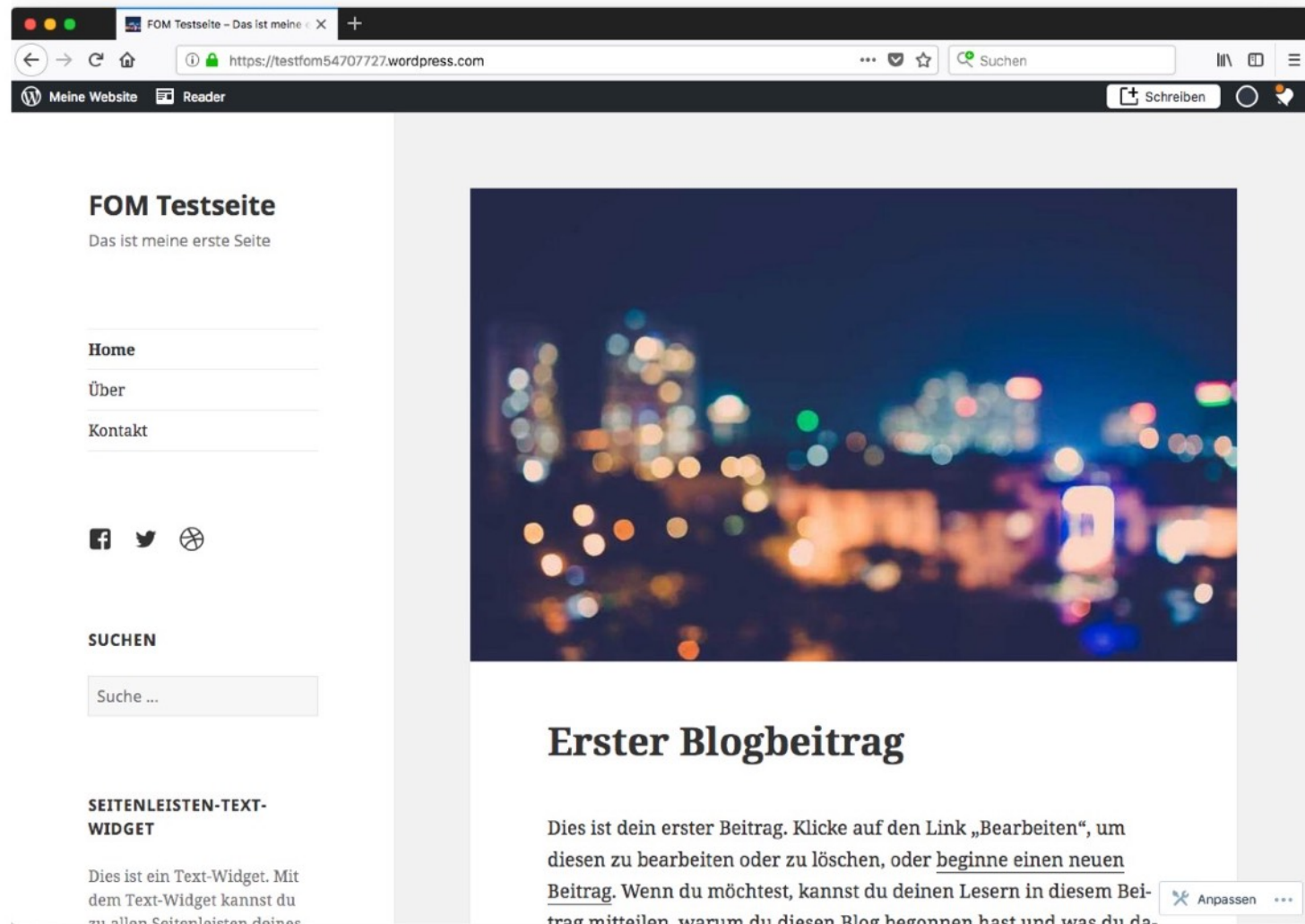
The main content area shows a 'Traffic' view for '20. Aug.' with a '100% Einrichten Erledigt' status. Below this is a table of statistics:

Statistik	Wert
AUFRUFE	0
BESUCHER	0
GEFÄLLT MIR	0
KOMMENTARE	0

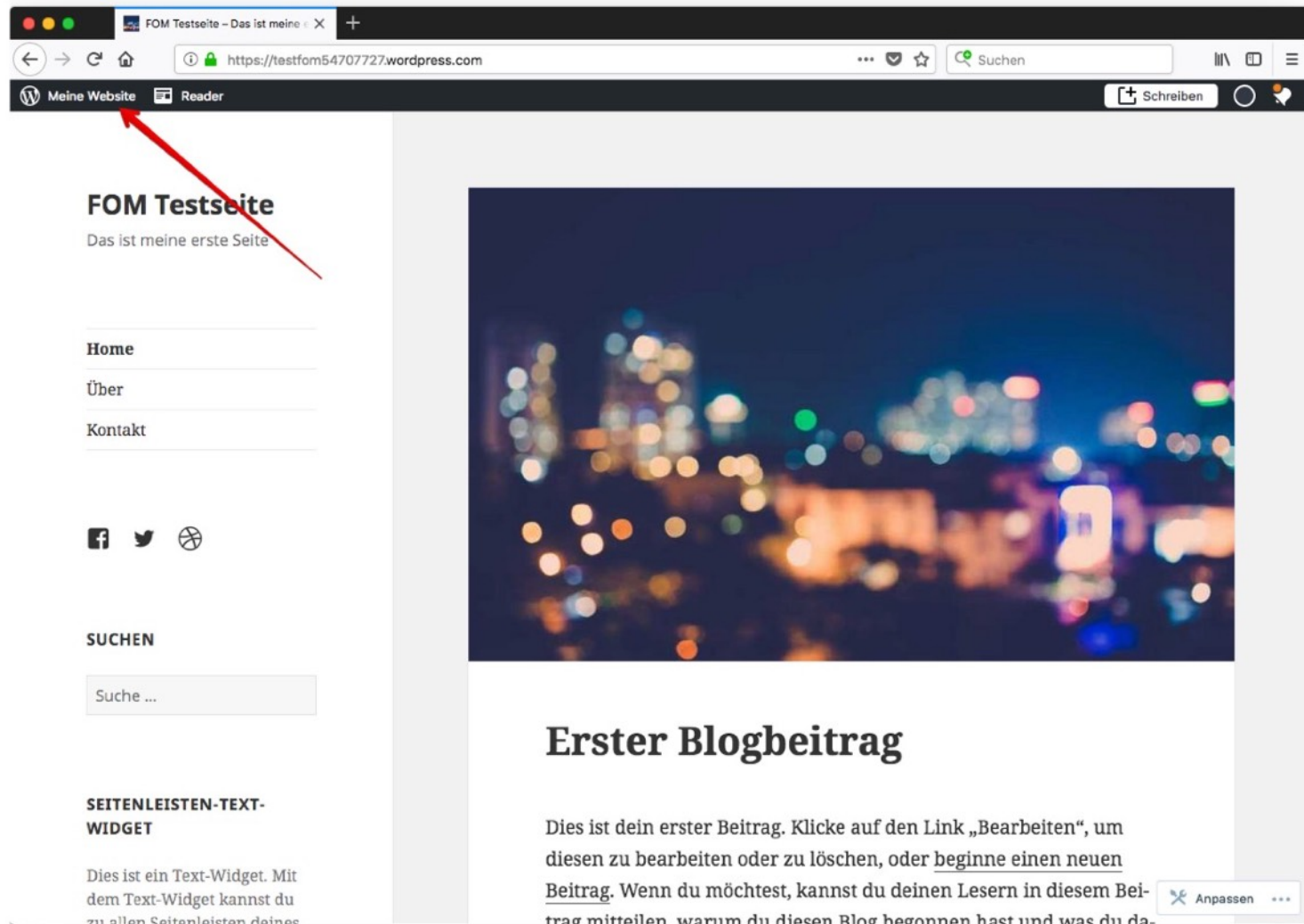
Additional statistics shown: 10 AUFRUFE, 5 BESUCHER, 0 KOMMENTARE. A message states 'Keine Aktivität im Zeitraum' (No activity in the period).

Wordpress Frontend Oberfläche

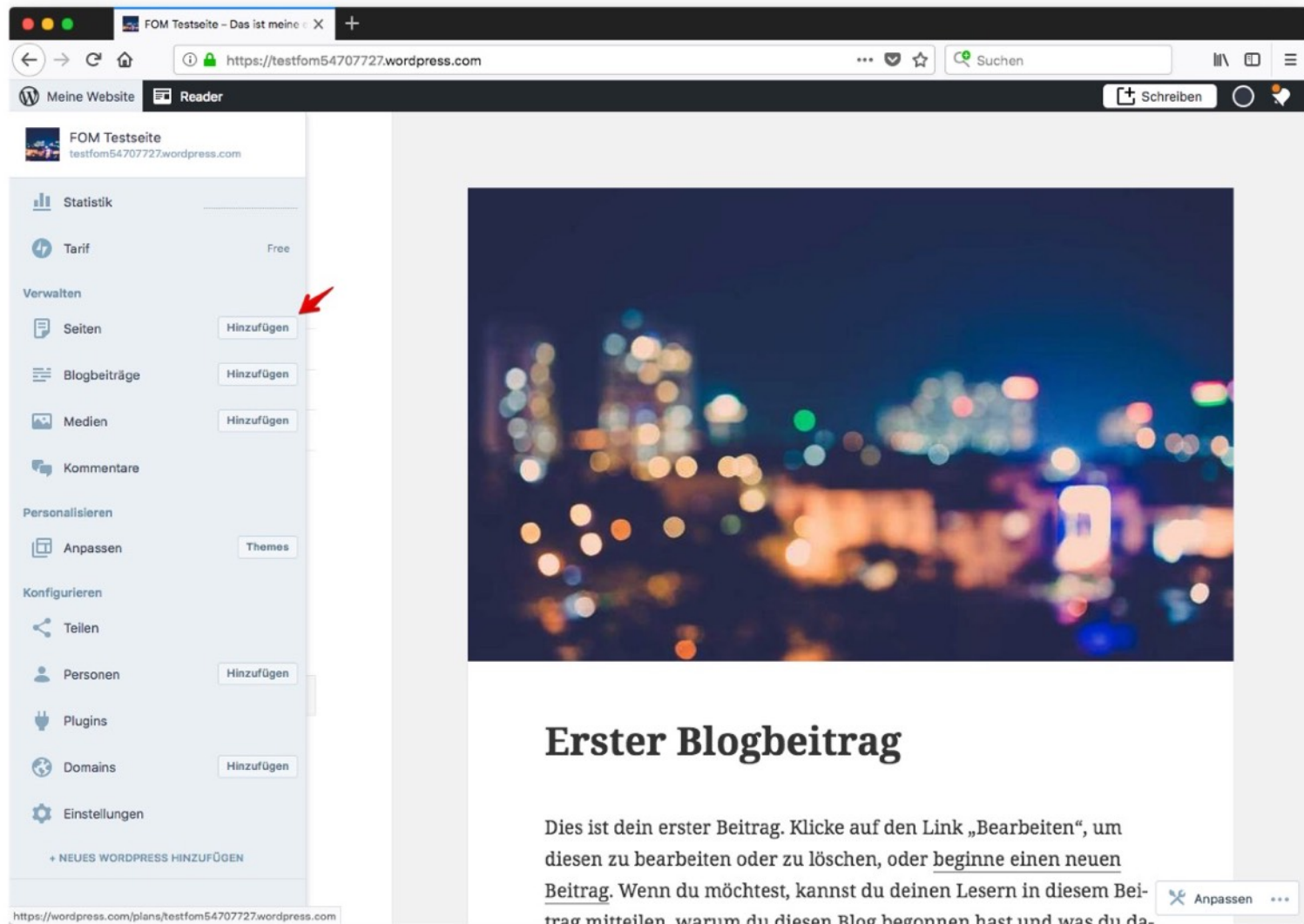




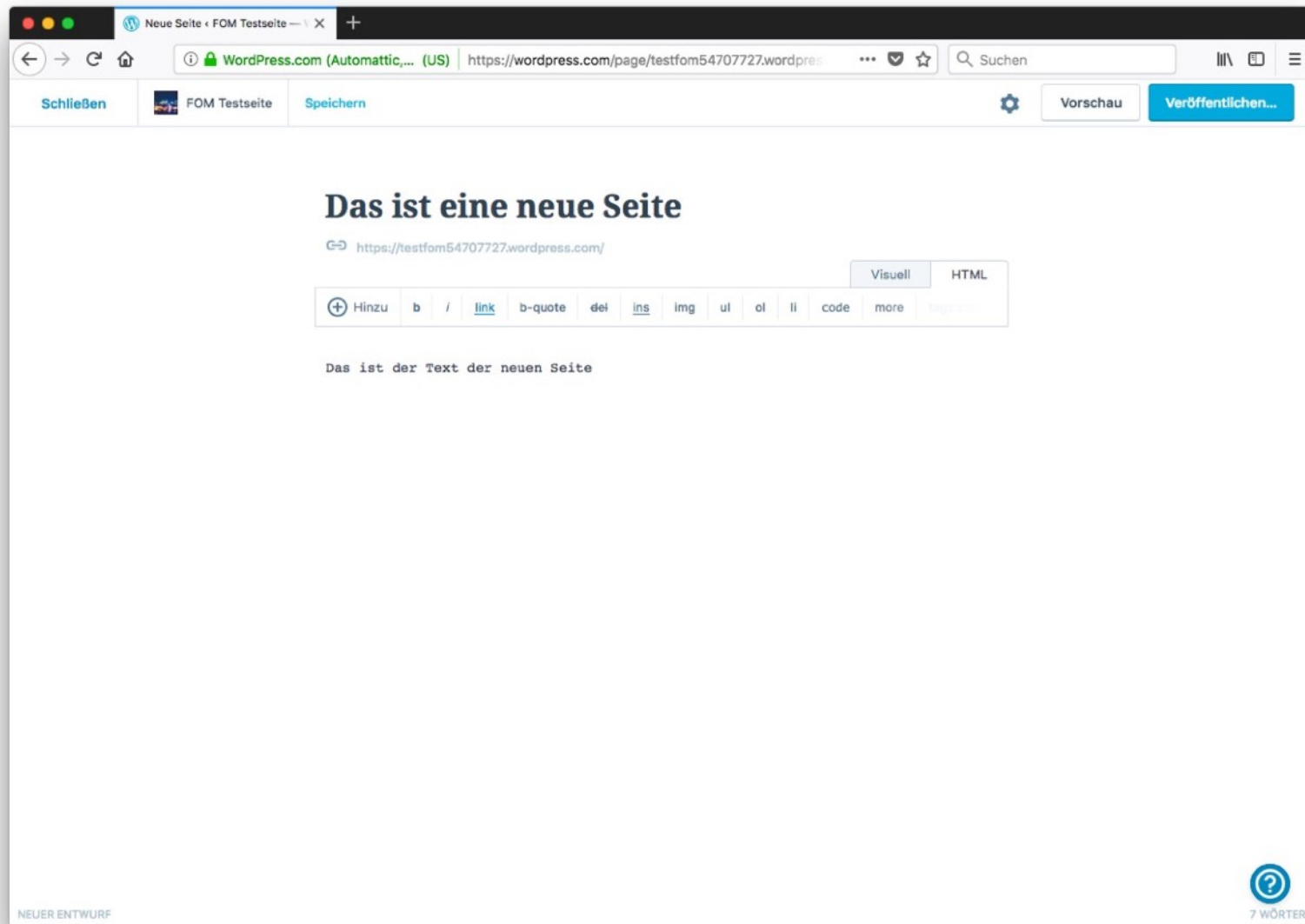
Seiten anlegen



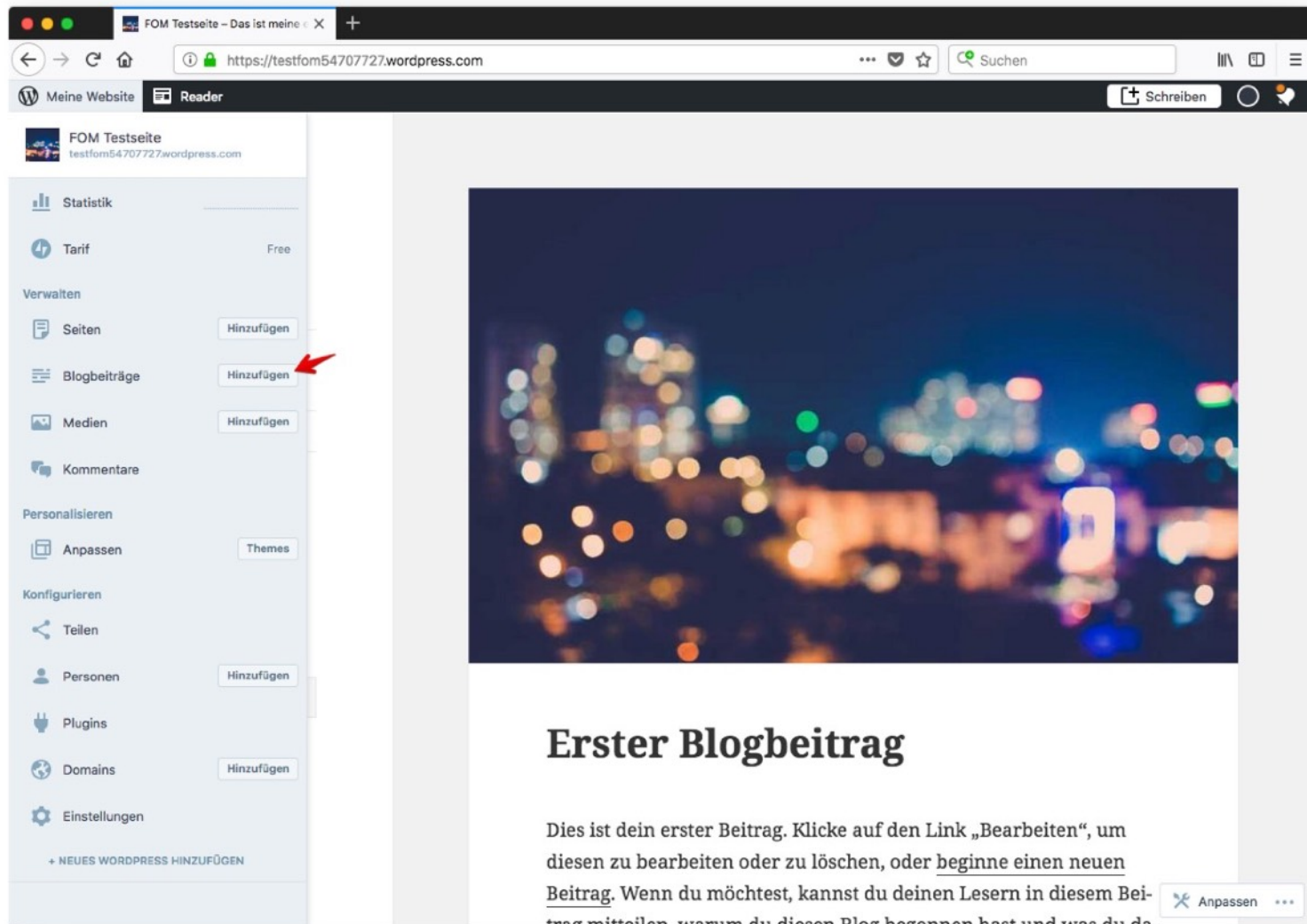
Seiten anlegen



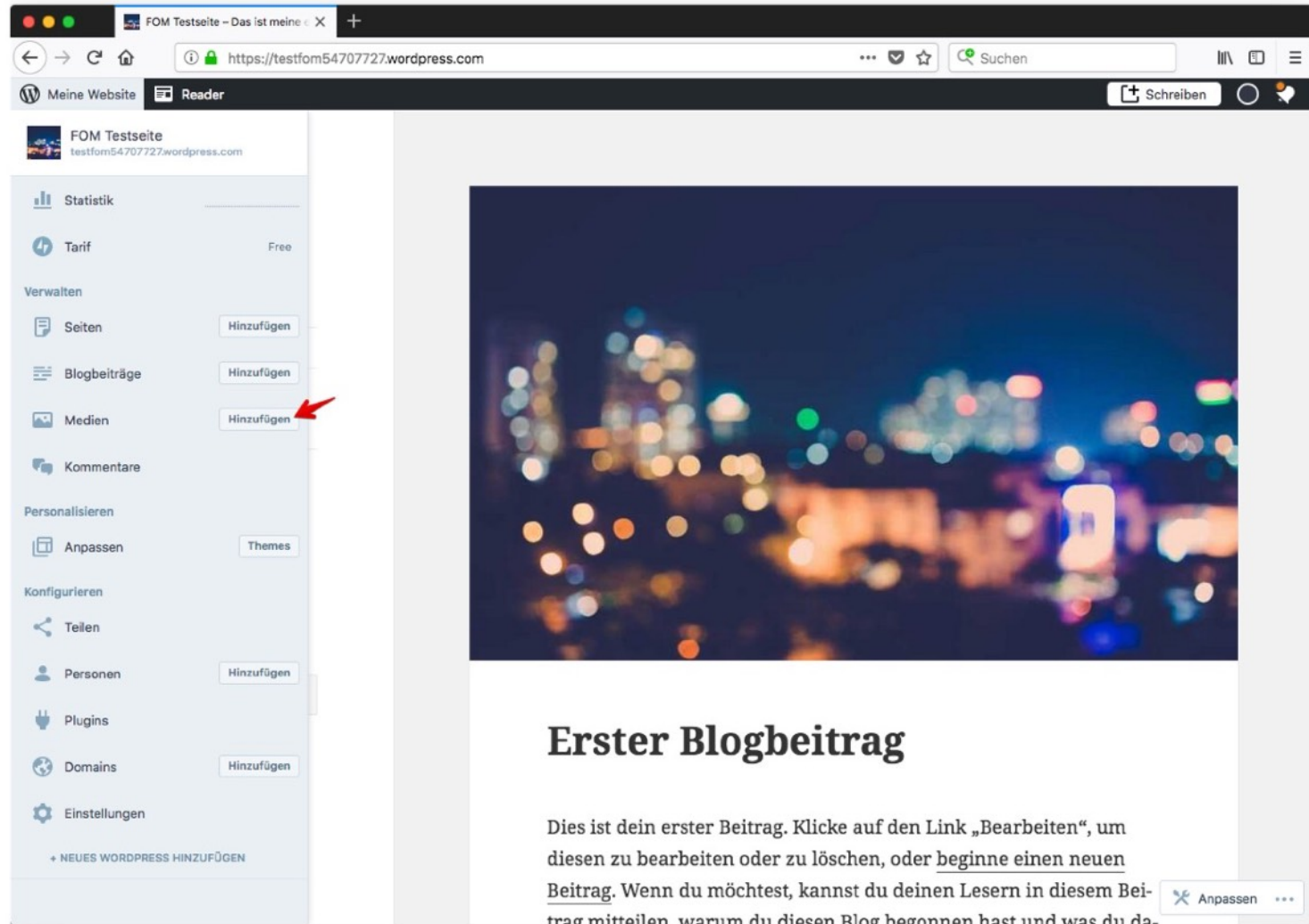
Seiten anlegen



Blogbeiträge anlegen



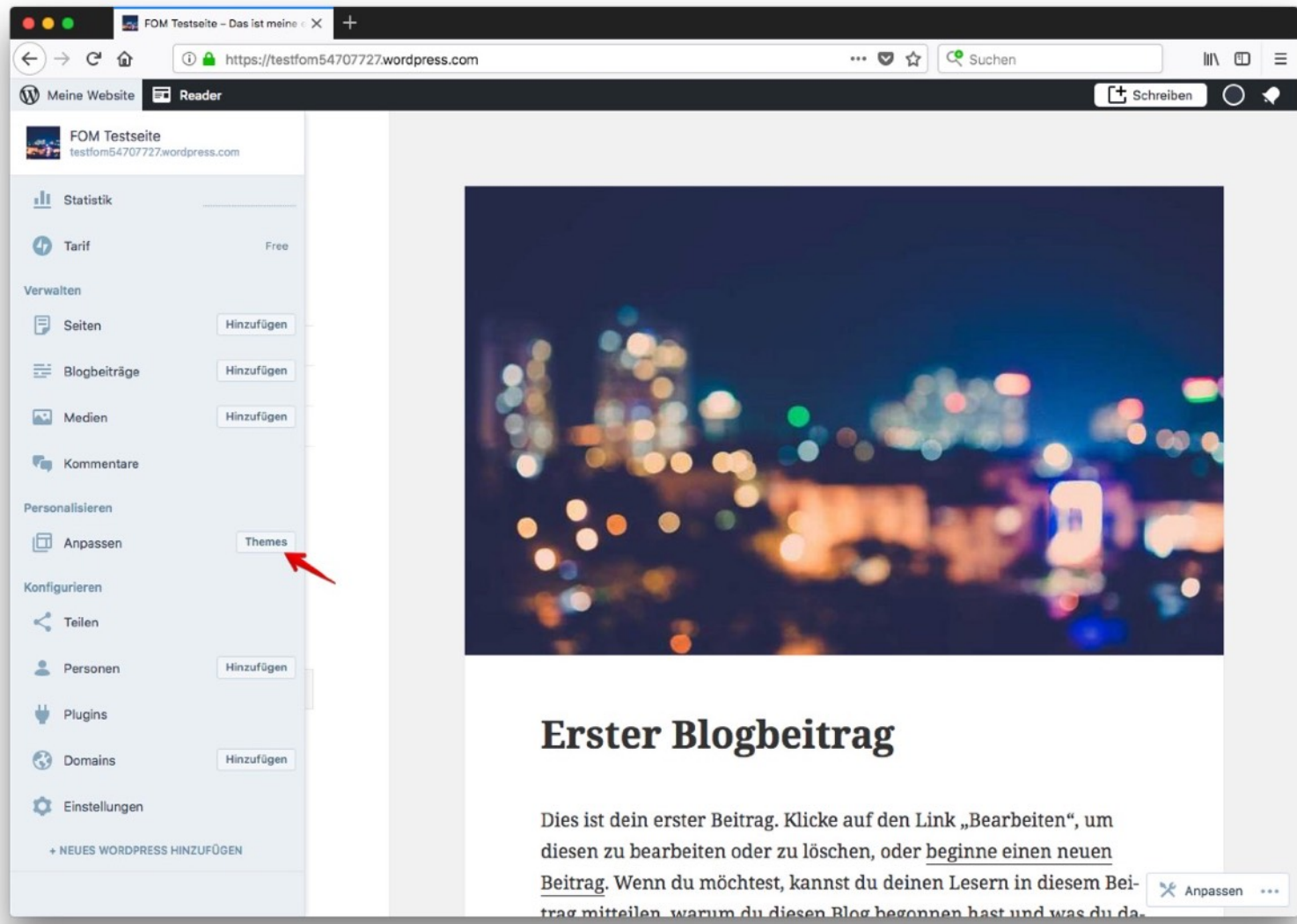
Medien hochladen



Übungsaufgaben Wordpress

- 1. Erstellen Sie zwei neue Seiten „Ueber mich“ und „Mein Beruf“ und beschreiben Sie diese Themen kurz**
- 2. Denken Sie daran, Ihre Veränderungen über den Button „Aktualisieren“ abzuspeichern**
- 3. Erstellen Sie zwei neue Blogbeiträge zu Ihren Hobbies und beschreiben Sie diese kurz**
- 4. Binden Sie eine PDF Datei Ihrer Wahl in einer der angelegten Blogbeiträge ein, so dass der Besucher darauf zugreifen kann**

Anpassen des Themes aus der Vorschau



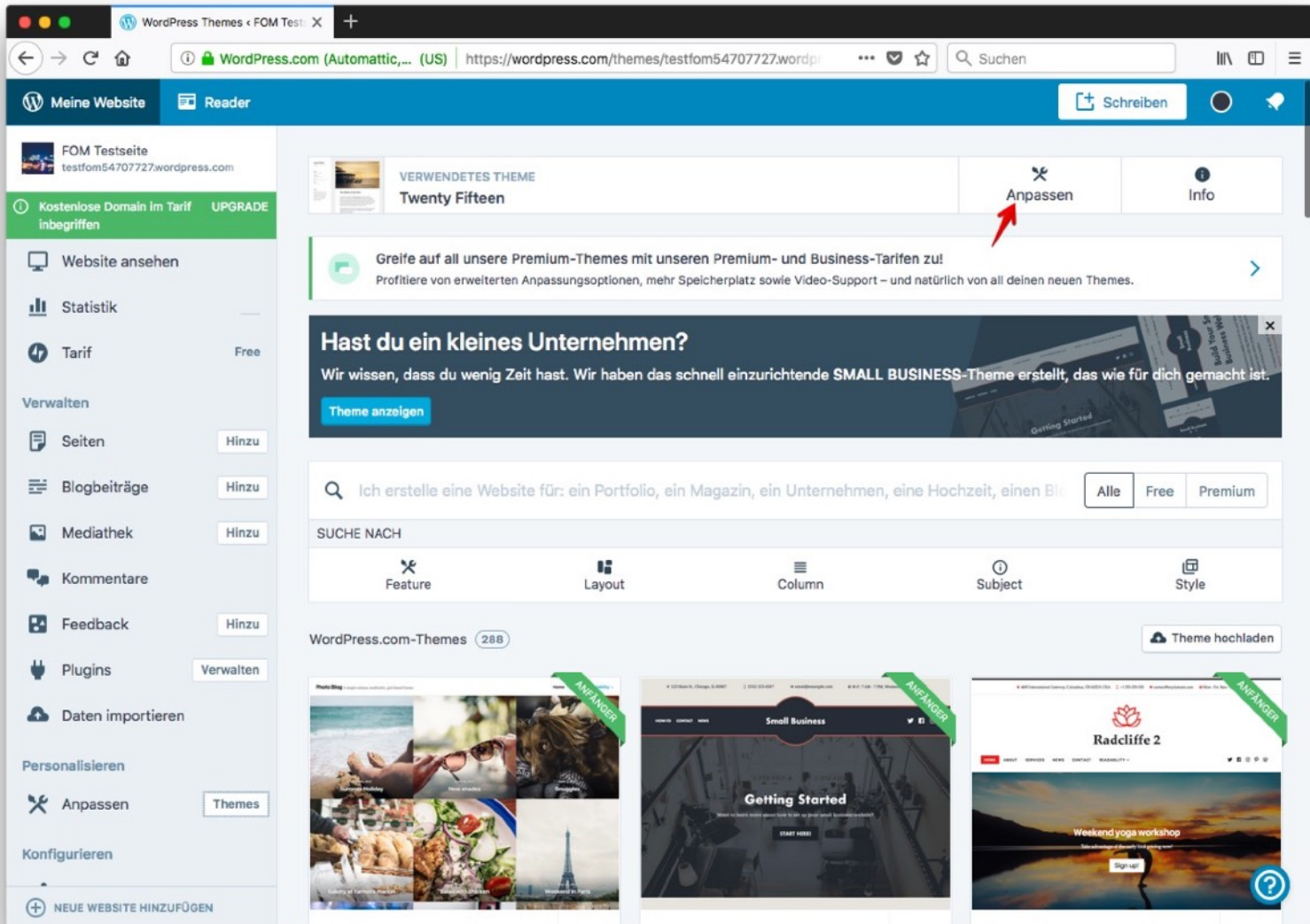
Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress

Anpassen des Themes aus dem Backend

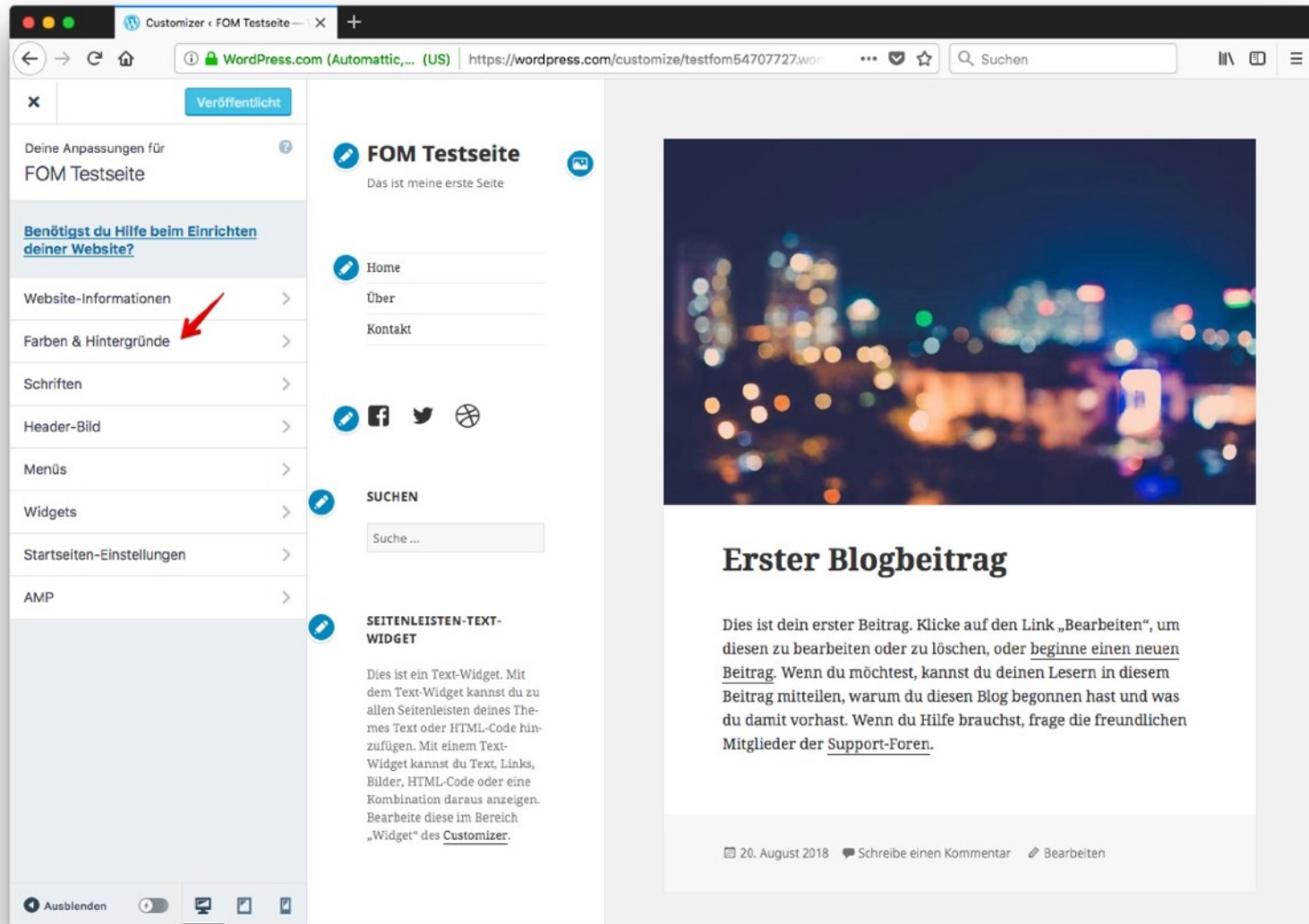
The screenshot shows the WordPress dashboard for a site named 'FOM Testseite'. The left sidebar contains a menu with the following items: 'Meine Website', 'Reader', 'Statistik', 'Tarif', 'Verwalten', 'Seiten', 'Blogbeiträge', 'Mediathek', 'Kommentare', 'Feedback', 'Plugins', 'Daten importieren', 'Personalisieren', 'Anpassen', and 'Konfigurieren'. The 'Anpassen' item is highlighted with a red arrow. The main content area displays a 'Traffic' overview for the date '20. Aug.' with a message 'Deine Website ist zur Veröffentlichung bereit!' and a '100% Erledigt' status. Below this, there is a table showing statistics for 'AUFRUFE' (0), 'BESUCHER' (0), 'GEFÄLLT MIR' (0), and 'KOMMENTARE' (0). The footer indicates 'Statistiken für 20. August 2018' and 'Letzte Aktualisierung: 19:56'.

Fallstudie: Umsetzung anhand eines WCMS Systems z.B. Wordpress

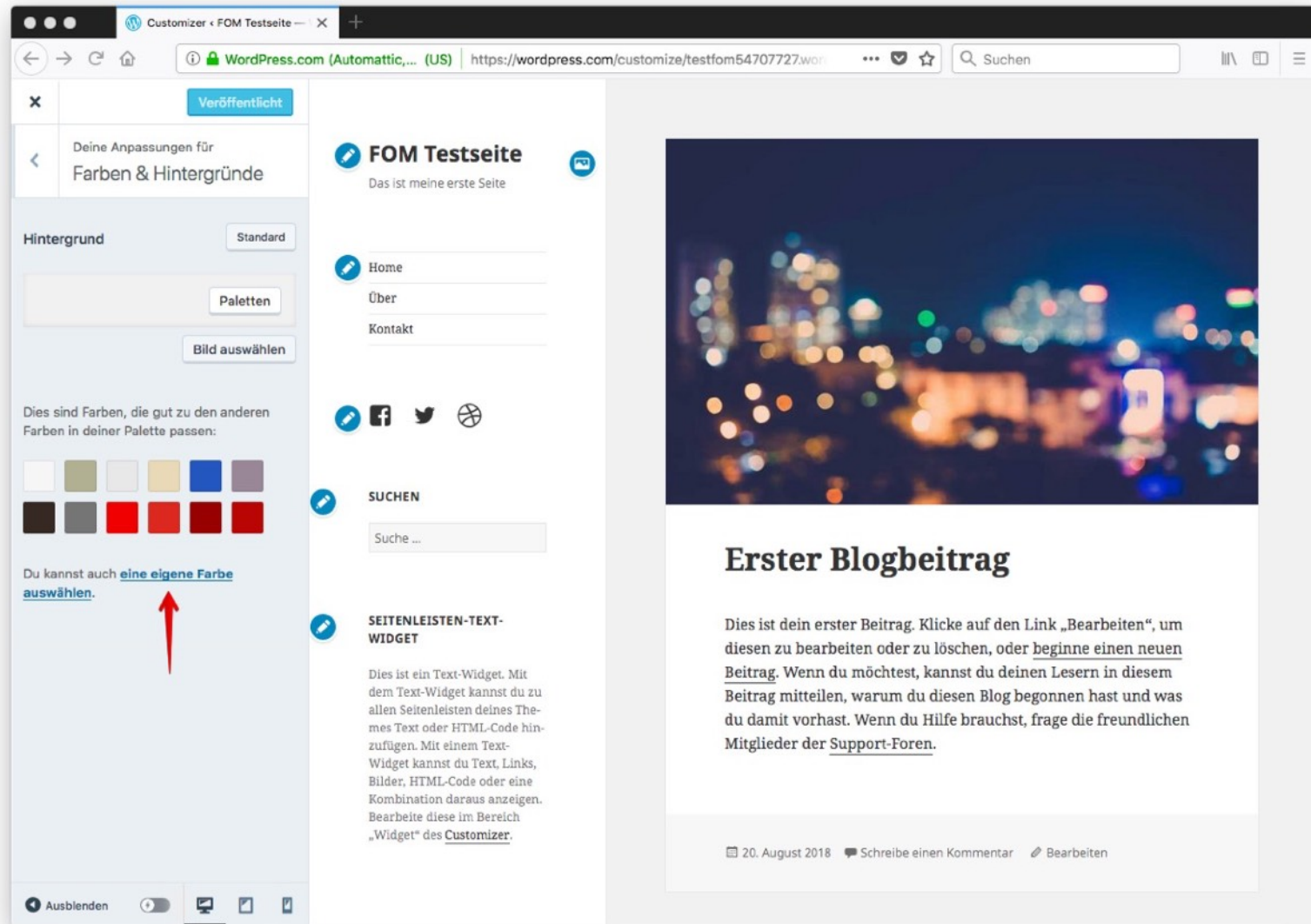
Anpassen des Themes



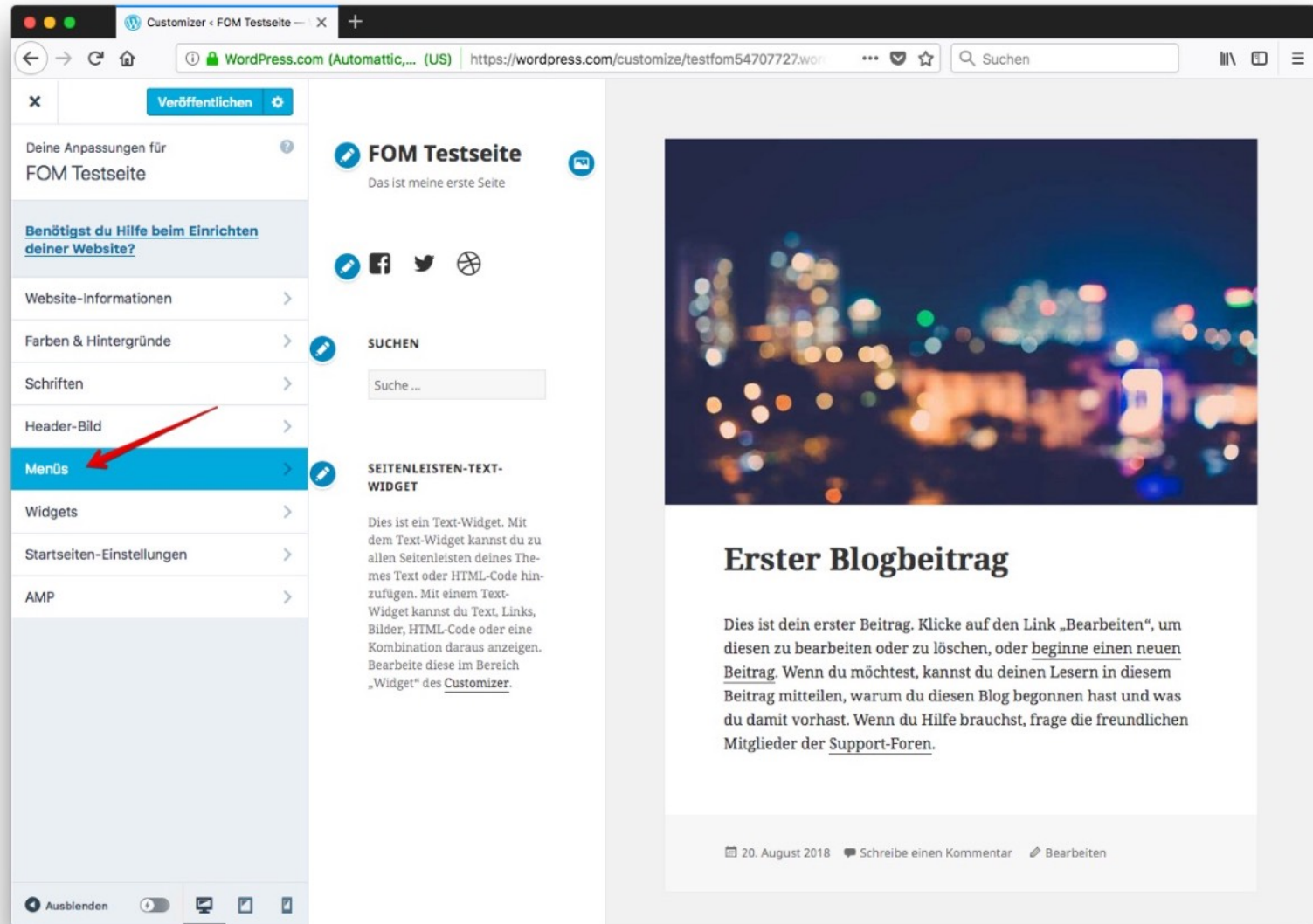
Anpassen des Themes



Anpassen des Themes



Erstellen eines eigenen Menüs



Erstellen eines eigenen Menüs

The screenshot displays the WordPress Customizer interface for creating a new menu. The left sidebar contains the following sections:

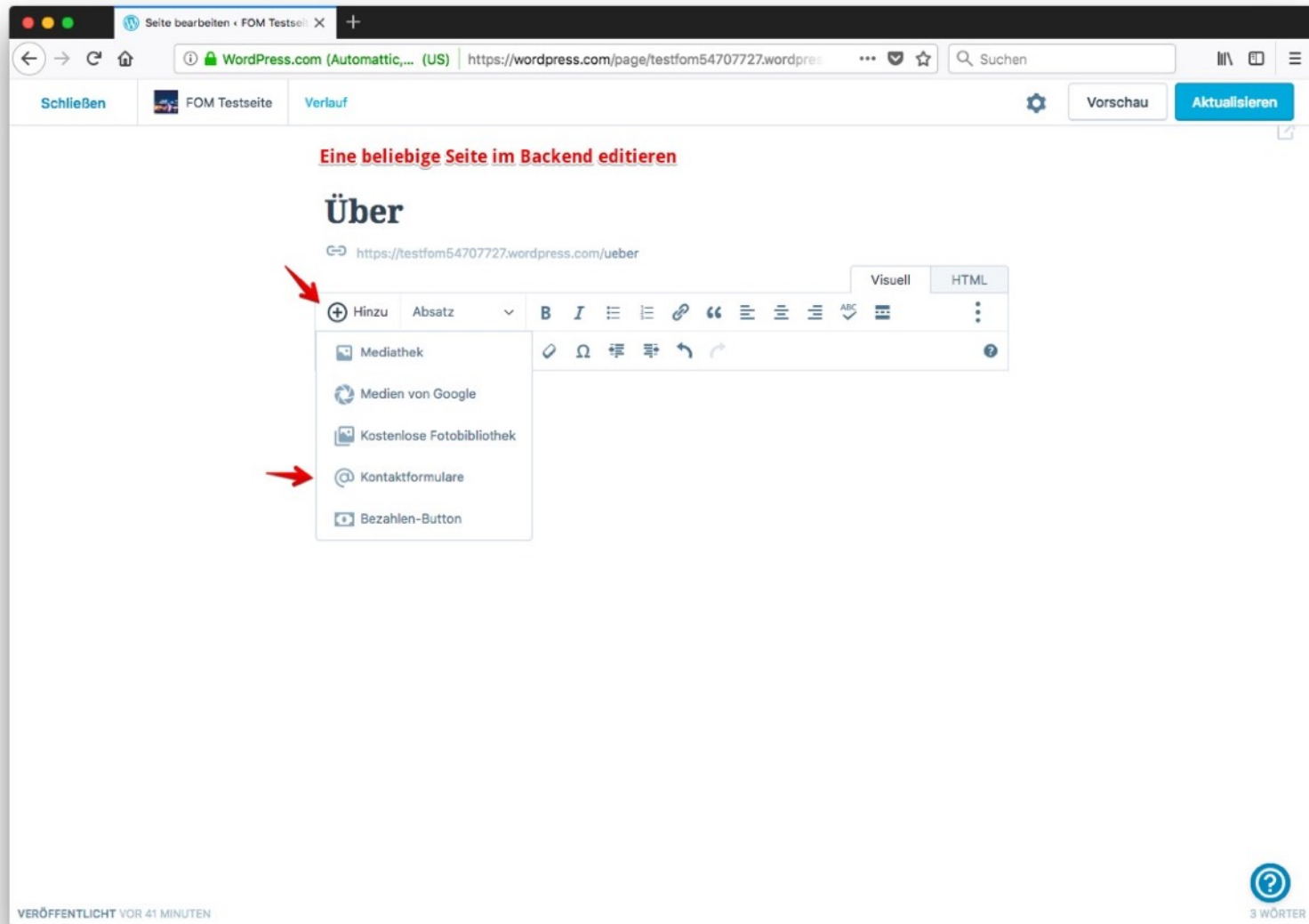
- Name des Menüs:** A text field containing 'Test'.
- Höchste Zeit, ein paar Links hinzuzufügen!** (Highest time to add a few links!): A paragraph of instructional text.
- Einträge hinzufügen:** A button to add items to the menu.
- Menü-Positionen:** A section where 'Primäres Menü' is selected with a checkbox.
- Menü-Einstellungen:** A section where 'Neue Seiten der ersten Ebene automatisch zum Menü hinzufügen' is checked.

The right sidebar shows a list of pages to be added to the menu:

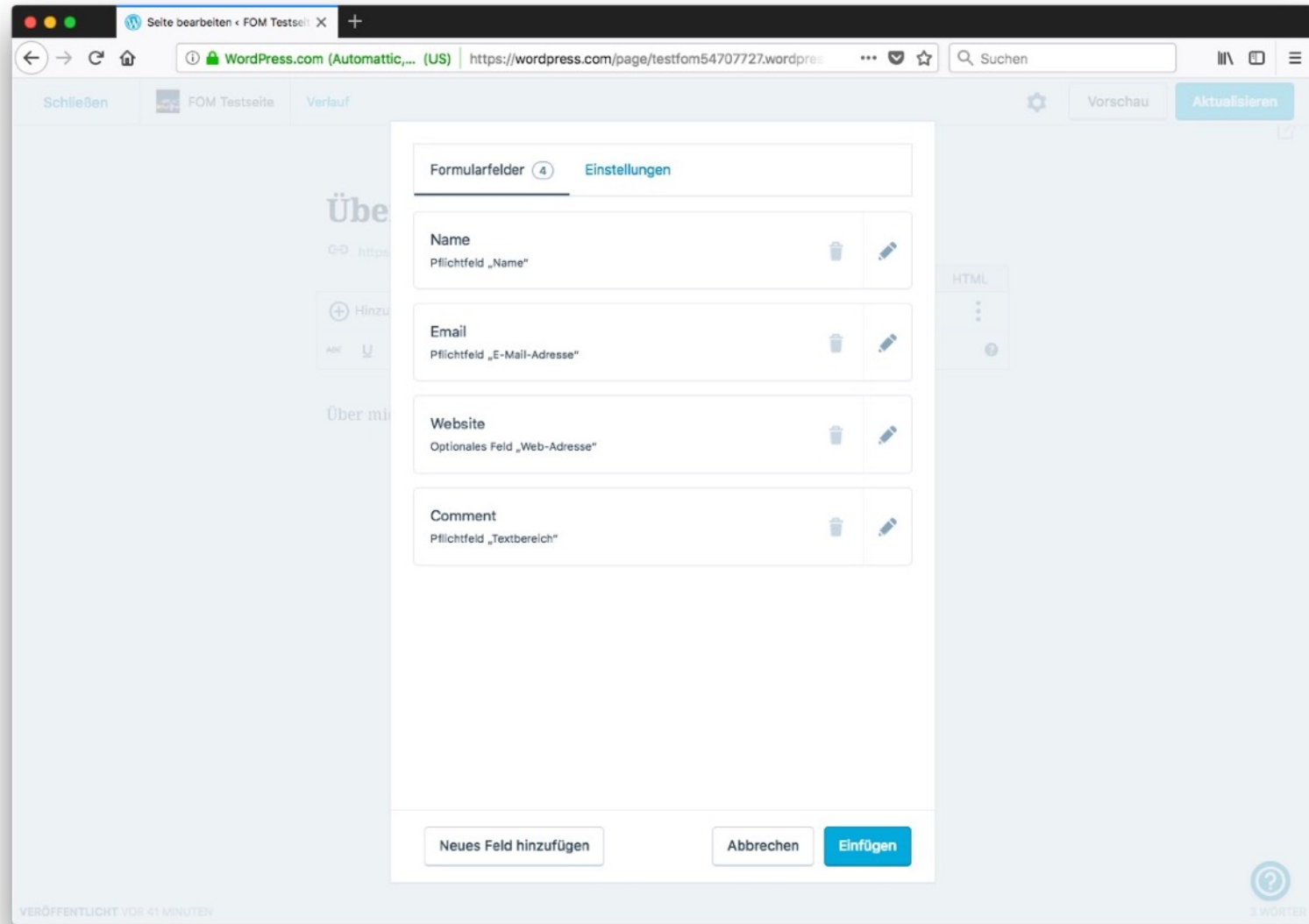
- Startseite (Home)
- Über (About)
- Kontakt (Contact)

The background preview shows a website layout with a header, a search bar, and a blog post titled 'Erster Blogbeitrag'.

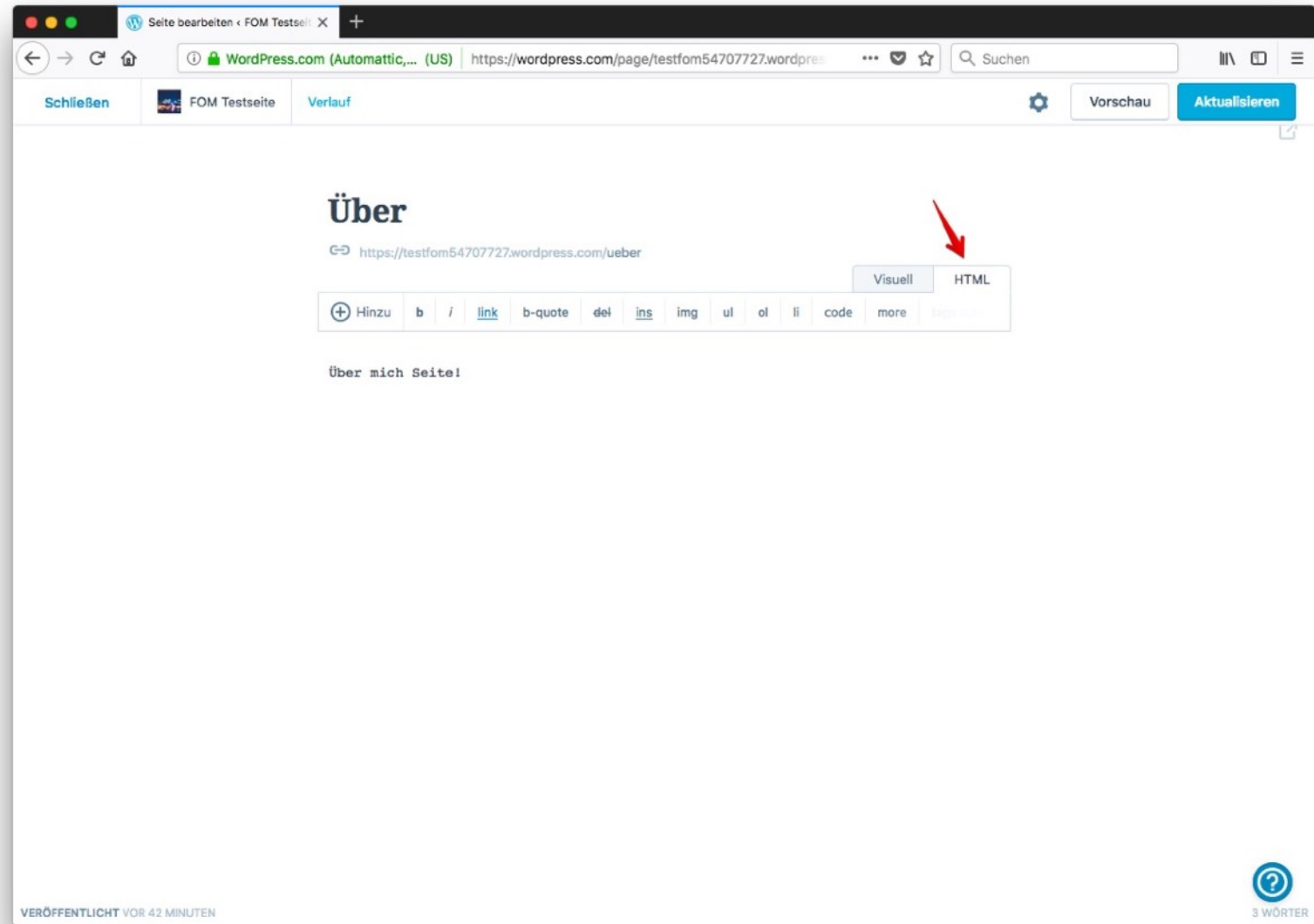
Kontaktformular anlegen



Kontaktformular anlegen



HTML Inhalt einbinden



HTML Inhalt einbinden

Seite bearbeiten < FOM Testseite X

WordPress.com (Automattic, Inc.) (US) https://wordpress.com/page/testfom54707727.wordpress.com/ueber

Schließen FOM Testseite Verlauf

Vorschau Aktualisieren

Über

<https://testfom54707727.wordpress.com/ueber>

Visuell HTML

Hinzu b / link b-quote del ins img ul ol li code more

Über mich Seite!

```
<h2>Das ist eine Überschrift</h2>
```

```
<p>Das ist ein Paragraph</p>
```

```
<ul>
```

```
<li>Aufzählungspunkt 1</li>
```

```
<li>Aufzählungspunkt 2</li>
```

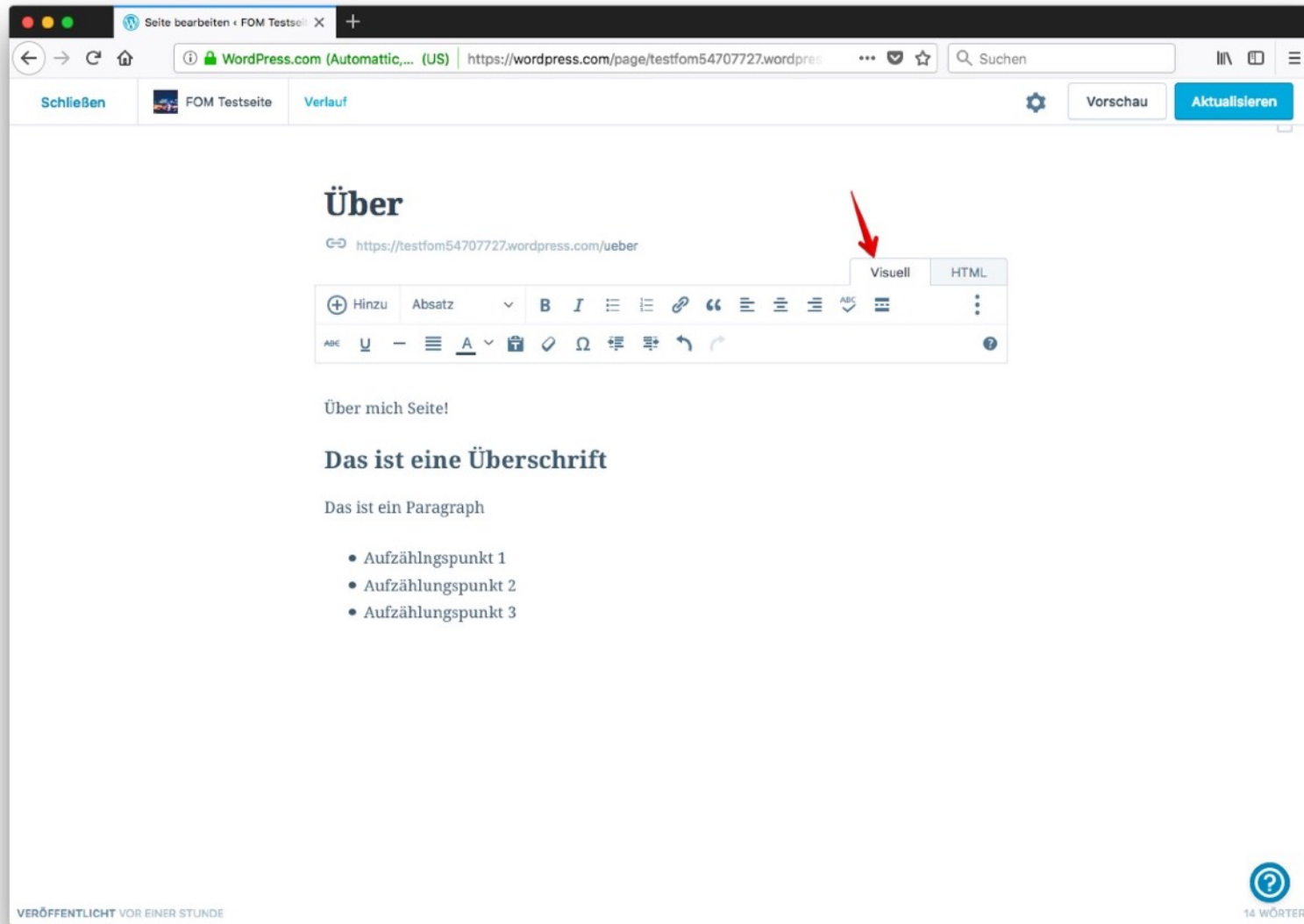
```
<li>Aufzählungspunkt 3</li>
```

```
</ul>
```

VERÖFFENTLICHT VOR EINER STUNDE

14 WÖRTER

HTML Inhalt einbinden



The screenshot shows the w3schools.com website. The browser address bar displays 'https://www.w3schools.com/html/default.asp'. The website header includes the w3schools.com logo and the tagline 'THE WORLD'S LARGEST'. A navigation menu lists various web technologies: HTML, CSS, JAVASCRIPT, SQL, PHP, BOOTSTRAP, HOW TO, JQUERY, W3.CSS, MORE, and REFERENCE. The 'HTML' tab is selected. On the left, a sidebar lists 'HTML5 Tutorial' topics, with 'HTML HOME' highlighted. The main content area features a banner for 'duda' with the text 'GOODBYE WORDPRESS, HELLO DUDA' and a 'See Why' button. Below the banner is the 'HTML5 Tutorial' section, which includes a 'Home' link and a green box containing the text: 'With HTML you can create your own Website. This tutorial teaches you everything about HTML. HTML is easy to learn - You will enjoy it.' Further down, the 'Examples in Every Chapter' section states: 'This HTML tutorial contains hundreds of HTML examples. With our online HTML editor, you can edit the HTML, and click on a button to view the result.' An 'Example' section shows a code snippet for a basic HTML document structure.

← → ↻ Sicher | <https://www.w3schools.com/html/default.asp>

w3schools.com THE WORLD'S LARGEST

🏠 **HTML** CSS JAVASCRIPT SQL PHP BOOTSTRAP HOW TO JQUERY W3.CSS MORE ▾ REFERENCE

HTML5 Tutorial

HTML HOME

- HTML Introduction
- HTML Editors
- HTML Basic
- HTML Elements
- HTML Attributes
- HTML Headings
- HTML Paragraphs
- HTML Styles
- HTML Formatting
- HTML Quotations
- HTML Comments
- HTML Colors
- HTML CSS
- HTML Links
- HTML Images
- HTML Tables
- HTML Lists
- HTML Blocks
- HTML Classes
- HTML Id
- HTML Iframes
- HTML JavaScript
- HTML File Paths
- HTML Head
- HTML Layout
- HTML Responsive
- HTML Computercode
- HTML Entities

duda GOODBYE WORDPRESS, HELLO DUDA [See Why](#)

HTML5 Tutorial

[← Home](#)

With HTML you can create your own Website.

This tutorial teaches you everything about HTML.

HTML is easy to learn - You will enjoy it.

Examples in Every Chapter

This HTML tutorial contains hundreds of HTML examples.

With our online HTML editor, you can edit the HTML, and click on a button to view the result.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
```

Übungsaufgaben Wordpress

- 1. Passen Sie die Farbgebung der Seite zu der von Ihnen gewünschten Farbe an**
- 2. Tauschen Sie das Headerbild gegen eines Ihrer Wunschbilder**
- 3. Bearbeiten Sie die Seite „Kontakt“ und fügen Sie dort ein Kontaktformular mit den entsprechenden Feldern ein**
- 4. Erstellen Sie ein eigenes neues Menü, welches die Hauptnavigation ersetzt und die Ihnen gewünschten Seiten auflistet**
- 5. Legen Sie eine neue Seite mit dem Namen „HTML“ an und erklären Sie dort mindestens 5 unterschiedliche HTML Elemente und deren Funktionsweise, in dem Sie die Elemente selbst über die “HTML Ansicht“ von Wordpress verwenden**
- 6. Verwenden Sie mindestens 5 unterschiedliche CSS Funktionen (per Inline Style) in den von Ihnen erstellen Elementen und erläutern Sie diese ebenfalls im Inhalt der Seite**

HTML Elemente einbinden - Youtube

← → ↻ Sicher | <https://www.youtube.com/watch?v=UB1O30fR-EE>

☰ YouTube html tutorial 🔍

What is HTML?

- Hyper Text Markup Language
- **NOT** a programming language
- Markup Language for creating webpages / documents
- Building blocks of the Web

▶ ⏮ 🔊 3:44 / 1:00:41

HTML Crash Course For Absolute Beginners

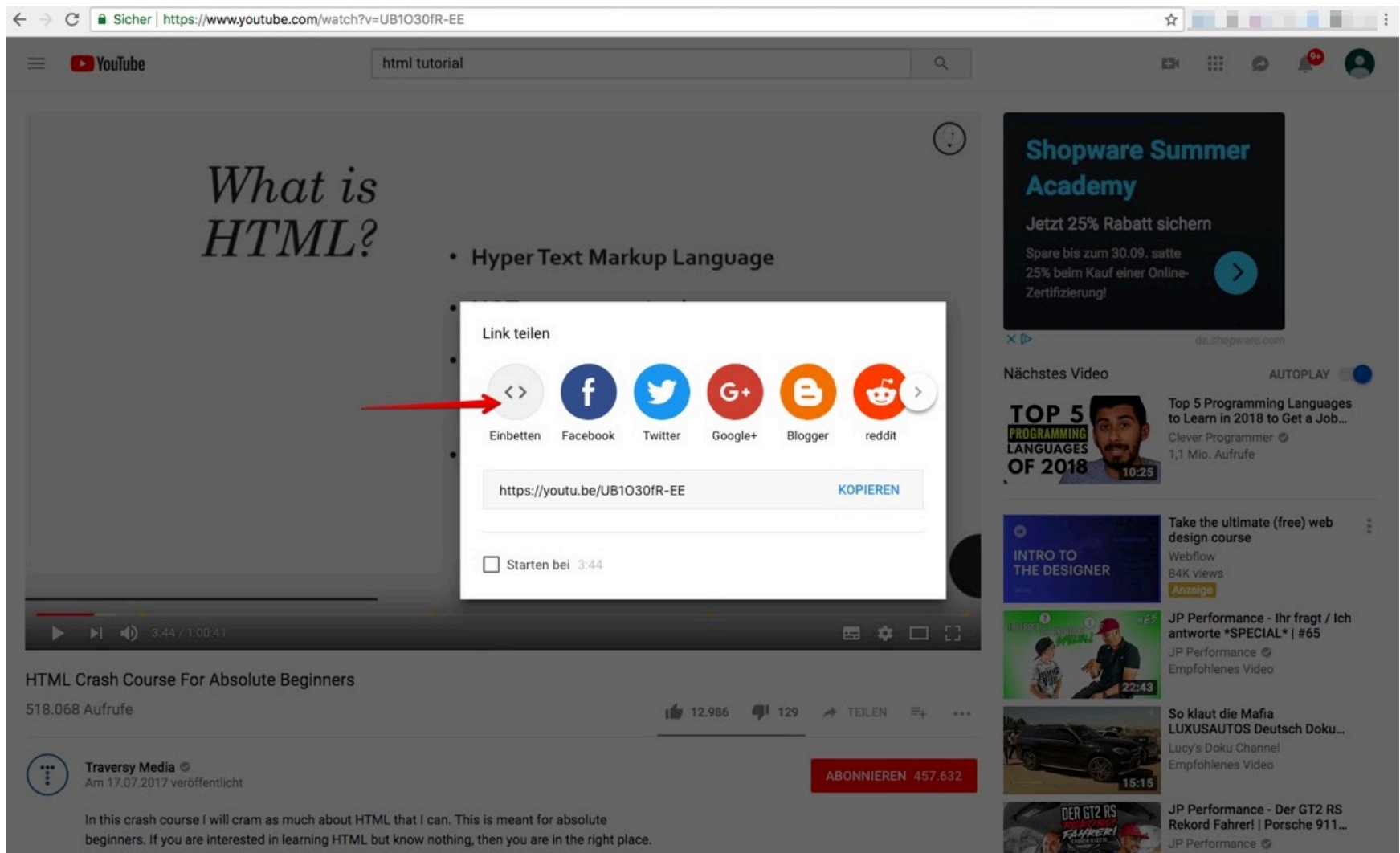
518.068 Aufrufe

👍 12.986 💬 129 ➦ TEILEN ➦ ...

👤 Traversy Media ✓
Am 17.07.2017 veröffentlicht

ABONNIEREN 457.632

HTML Elemente einbinden - Youtube



HTML Elemente einbinden - Youtube

← → ↻ Sicher | https://www.youtube.com/watch?v=UB1O30fR-EE

html tutorial

YouTube

What is

HTML Crash Course For Absolute Beginners

HTML Crash Course Absolute Beginners

100
101
102
103
104
105

```

</div>
<a clas
<spa
<spa
</a>
<a class=

```

HTML Crash Course For Absolute Beginners

518.068 Aufrufe

Traversy Media

Am 17.07.2017 veröffentlicht

In this crash course I will cram as much about HTML that I can. This is meant for absolute beginners. If you are interested in learning HTML but know nothing, then you are in the right place.

12.986 129 TEILEN

ABONNIEREN 457.632

Embed Video

```

<iframe width="560" height="315"
src="https://www.youtube.com/embed/UB
1030fR-EE" frameborder="0"
allow="autoplay; encrypted-media"
allowfullscreen></iframe>

```

☐ Starten bei 0:00

OPTIONEN ZUM EINBETTEN

- ☒ Nach Ende des Videos vorgeschlagene Videos anzeigen.
- ☒ Player-Steuerelemente anzeigen.
- ☒ Videotitel und Player-Aktionen anzeigen.

KOPIEREN

22:43

15:15

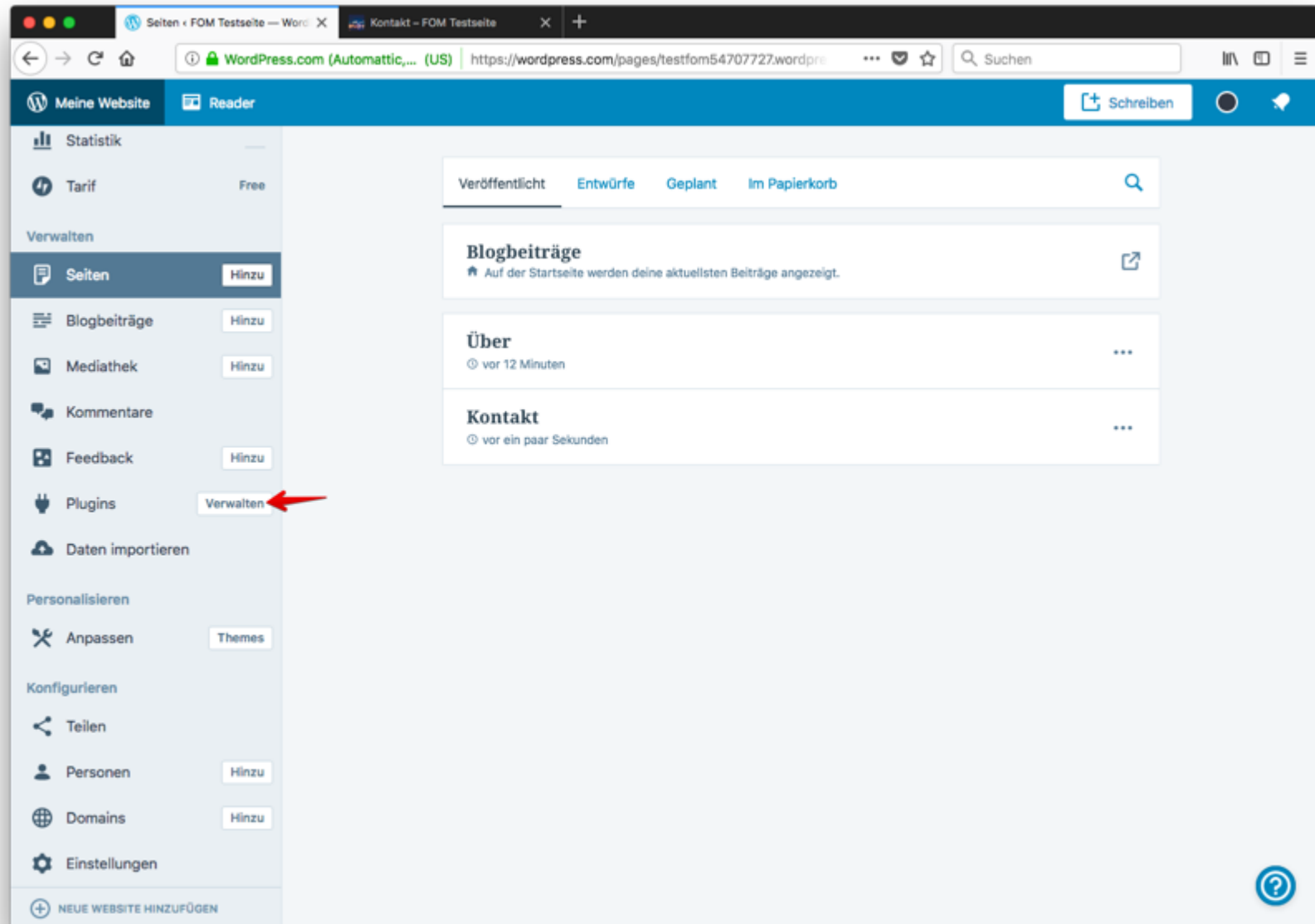
So kauft die Mafia LUXUSAUTOS Deutsch Doku...
Lucy's Doku Channel
Empfohlenes Video

DER GT2 RS
FAHRT
JP Performance - Der GT2 RS
Rekord Fahrer! | Porsche 911...

Übungsaufgaben Wordpress

- 1. Binden Sie mindestens ein Youtube Video auf der von Ihnen angelegten Seite „HTML“ ein und passen Sie die Darstellung nach Ihren Vorstellungen an (<https://www.youtube.com>)**
- 2. Was stellen Sie bei der Einbindung des Videos im HTML Editor fest?**
- 3. Binden Sie eine Google Maps Karte mit Ihrem Standort der FOM ein (<https://maps.google.com>)**
- 4. Was stellen Sie bei der Einbindung der Google Maps Karte im HTML Editor fest?**

Plugins einbinden



Plugins einbinden

The screenshot shows the WordPress.com interface for a user named 'FOM Testseite'. The browser address bar shows the URL 'https://wordpress.com/plugins/testfom54707727.wordpr...'. The left sidebar contains a menu with options like 'Statistik', 'Verwalten', 'Plugins', and 'Einstellungen'. The main content area displays a list of recommended plugins under the 'Vorgestellt' section. A red arrow points to a notification banner at the top right that says 'Upgrade NICHT durchführen'.

Plugin Name	Developer	Rating	Status
WooCommerce	Automattic	★★★★★	Not Installed
Yoast SEO	Team Yoast	★★★★★	Not Installed
BuddyPress	The BuddyPress Community	★★★★★	Not Installed
TinyMCE Advanced	Andrew Ozz	★★★★★	Not Installed
Google Analytics for WordPress	MonsterInsights	★★★★☆	Not Installed
Page Builder by SiteOrigin	SiteOrigin	★★★★★	Not Installed
Contact Form 7	Takayuki Miyoshi	★★★★★	Not Installed
Yoast SEO	Team Yoast	★★★★★	Not Installed
Akismet Anti-Spam	Automattic	★★★★★	Not Installed
Jetpack by WordPress.com	Automattic	★★★★☆	✓ Installiert
WordPress Importer	wordpressdotorg	★★★★☆	Not Installed
WooCommerce	Automattic	★★★★★	Not Installed

Plugin Beispiel Yoast SEO

The screenshot shows the WordPress admin interface with the Yoast SEO plugin active. The left sidebar contains the standard WordPress menu items, with 'Plugins' highlighted. The main content area shows the post editor for a post titled 'Die Forderung des DIW ist folglich, den Sozialen Wohnungsbau zu stärken.' The Yoast SEO meta box is expanded, showing the 'Snippet Preview' section. Below this, the 'Lesbarkeit' (Readability) section is visible, indicating that the text is not readable and suggesting improvements. The 'Fokus-Keyword' (Focus Keyword) section is also visible, showing the keyword 'Mieter und Käufer in Not!'. The right sidebar contains the 'Sichtbarkeit' (Visibility) section, showing the post is set to 'Öffentlich' (Public). Below this, the 'Kategorien' (Categories) section is visible, showing the category 'allgemein' selected. The 'Schlagwörter' (Tags) section is also visible, showing a list of tags.

Post Title: Die Forderung des DIW ist folglich, den Sozialen Wohnungsbau zu stärken.

Wortanzahl: 415

Zuletzt geändert von: am 11. Juli 2018 um 13:26 Uhr

Yoast SEO

[Hilfe benötigt?](#) [Entscheide dich für Premium](#)

Snippet Preview

This is a rendering of what this post might look like in Google's search results. [Mehr über die Code-Schnipsel-Vorschau erfahren](#)

Mieter und Käufer in Not! -

[allgemein](#) [mieter-und-kaeuer-in-not](#)

Bitte bearbeiten Sie das Codeschnipsel und richten Sie eine Meta-Beschreibung ein. Wenn Sie dies nicht tun, wird Google selbständig versuchen, einen relevanten Teil Ihres Beitrags in den Suchergebnissen anzuzeigen.

[Code-Schnipsel bearbeiten](#)

Lesbarkeit

Fokus-Keyword

Add additional keyword

Cornerstone-Inhalt

Sichtbarkeit: Öffentlich [Bearbeiten](#)

Revisionen: 3 [Anzeigen](#)

Veröffentlicht am: 10. Jul 2018 um 13:19 [Bearbeiten](#)

[Duplicate This](#)

Lesbarkeit: Verbesserung nötig

SEO: Nicht verfügbar

[In Papierkorb legen](#) [Aktualisieren](#)

Kategorien

Alle Kategorien **Häufig genutzt**

☒ Allgemein

☐ Aktuelles

☐ Einzugsgebiet

☐ Ratingen

☐ Referenzen

[+ Neue Kategorie erstellen](#)

Schlagwörter

[OK](#)

Schlagwörter durch Kommas trennen.

[Wähle aus den meistgenutzten Schlagwörtern](#)

Theme wechseln

The screenshot shows the WordPress dashboard for a user named 'FOM Testseite'. The left sidebar contains various management options, with 'Themes' highlighted by a red arrow. The main area displays a confirmation message that the website is ready for publication, along with social media icons and a traffic statistics chart. The chart shows zero activity for the selected date, August 20th, 2018.

Statistik	Wert
AUFRUFE	0
BESUCHER	0
GEFÄLLT MIR	0
KOMMENTARE	0

Statistiken für 20. August 2018
Letzte Aktualisierung: 19:56

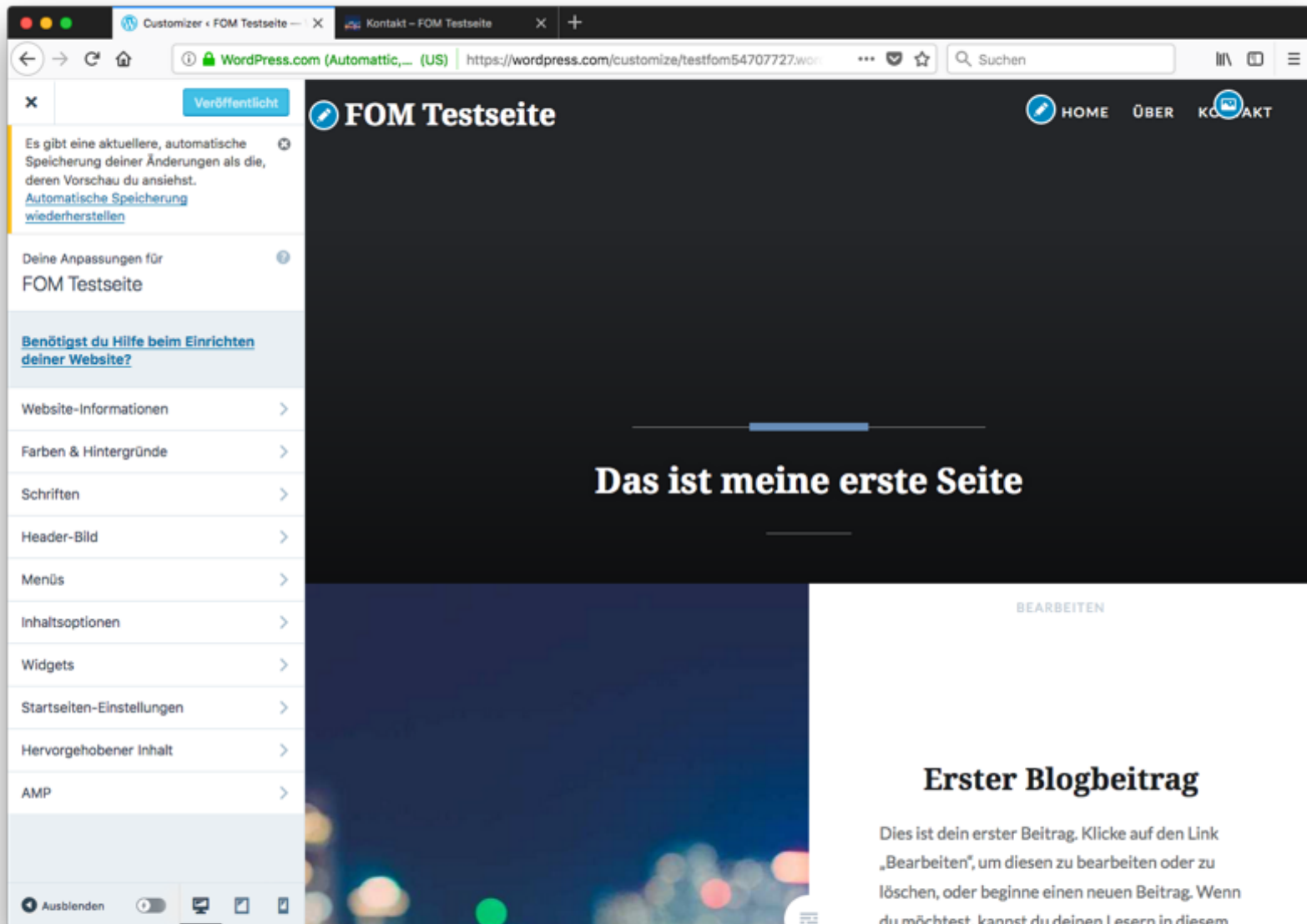
Theme wechseln

The screenshot shows the WordPress Themes directory interface. On the left is a sidebar with navigation options: 'Meine Website', 'Reader', 'Kostenlose Domain im Tarif inbegriffen', 'UPGRADE', 'Website ansehen', 'Statistik', 'Tarif', 'Verwalten', 'Seiten', 'Blogbeiträge', 'Mediathek', 'Kommentare', 'Feedback', 'Plugins', 'Daten importieren', 'Personalisieren', 'Anpassen', 'Konfigurieren', and 'NEUE WEBSITE HINZUFÜGEN'. The main area displays a grid of theme preview cards. The cards shown include 'Apostrophe 2', 'AltoFocus', 'INTERGALACTIC 2', 'Independent Publisher 2', 'Dyad 2', 'Penscratch 2', 'Backsville 2', and 'SPATIAL'. A red arrow points to the 'Dyad 2' theme card, and a red text label 'Dieses Theme "Dyad 2" suchen' is placed next to it. The top of the browser window shows the URL 'https://wordpress.com/themes/testfom54707727.wordpress.com/' and the search bar.

Theme wechseln

The screenshot displays the WordPress theme preview for 'Dyad 2' by Automattic. The interface is divided into two main sections. On the left, there is a sidebar with the title 'Dyad 2 von Automattic' and a button 'Dieses Design aktivieren FREE'. Below this, there are tabs for 'Überblick' and 'Support', and a link 'Live Demo öffnen'. The main content area on the left shows a description of the theme: '(Dyad 2 is a refreshed version of the Dyad theme, with more features and added flexibility.) Dyad 2 displays words and images in perfect balance: a post's featured image and text excerpt get equal billing on the main blog page, while single posts and pages size and position images dynamically, depending on the screen size. The homepage shows a Featured Content slider, that will display up to 6 posts, with a large Featured Image.' Below the text, there are three device mockups showing the theme's responsive design: a smartphone, a laptop, and a tablet. The laptop and tablet mockups show a post titled 'Making and drying pasta'. The smartphone mockup shows a post titled 'This is a sticky post'. On the right, there is a larger preview of the theme's homepage. It features a large featured image of yellow flowers and a bowl of cookies, with the title 'Almond lemon cookies'. Below this, there are two smaller featured images: one of a dish of food titled 'This is a sticky post' and another of a bowl of tomatoes titled 'Heirloom tomato salsa'. The bottom right corner of the preview has a blue question mark icon.

Thema wechseln



Übungsaufgaben Wordpress

- 1. Passen Sie Ihre Webseite nach der Umstellung auf „Dyad 2“ so an, dass diese Webseite Ihren Vorstellung entspricht und alle leeren Elemente befüllt sind (wie z.B. neues Headerbild)**

THE END
;-)