

Webprogrammierung - Zusammenfassung

1. Rendering Engine

- zentraler Bestandteil von Browsern
- technischer Kern beziehungsweise ein Motor
- Diese Komponente interpretiert den (HTML-)Code der aufzurufenden Webseite und zeichnet ihn, stellt die Seite also dar.
- Browser & Techniken:
 - o Google mit Chrome/Chromium (hohes Tempo)->Blink (für HTML) und V8 (für JavaScript)
 - o Mozilla mit Gecko (für HTML) und SpiderMonkey/JägerMonkey (für JavaScript)

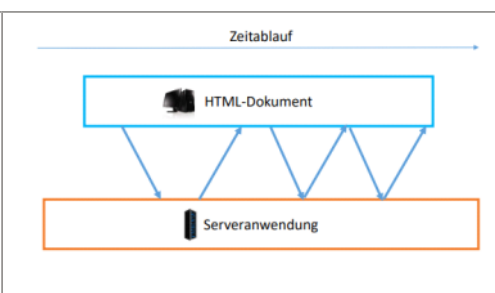
2. AJAX = Asynchronous JavaScript And XML

- Es können Asynchrone Anfragen an den Server geschickt werden.
- Möglichkeit mit dem Server Daten auszutauschen und die bestimmte Bereiche der Webseite zu aktualisieren OHNE DIE SEITE NEU ZU LADEN.
- kommuniziert über das XMLHttpRequest-Objekt
 - o können Anfragen an den SERVER geschickt werden und die Antwort entgegen genommen werden.
 - o Datenformate die ausgetauscht werden können: Text, JSON, XML,...
- XMLHttpRequest kann lediglich eine HTTP-Anfrage an einen Webserver schicken und die Rückgabe auswerten
- Nutzer merkt die Anfrage nicht

Desktop-Applikationen	Webapplikationen
<ul style="list-style-type: none"> - sind schnell - können große Datenmengen sofort verarbeiten - sind für ein spezielles Betriebssystem ausgelegt - Verarbeitung der Daten findet primär auf dem Computer statt <p>→ Interaktion mit 1 USER</p>	<ul style="list-style-type: none"> - Nicht auf dem Computer des Anwenders installiert - Basieren auf der Interaktion zwischen CLIENT und SERVER - Browser ist eine Desktop Applikation wird aber nur zur Darstellung der Inhalte eingesetzt. - Webapplikationen sind langsamer aber aktueller. <p>→ Interaktion mit 1-1000**** USER</p>

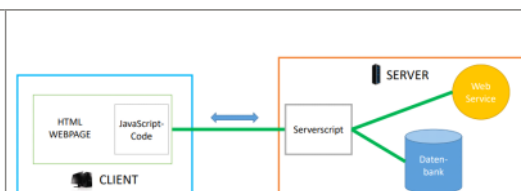
- Webapplikationen mit AJAX

- o Kommunikation muss vom CLIENT aus initiiert werden!
- o Webseite kann während des Ladens der Daten im Hintergrund vom Benutzer weiter bedient werden.
- o zu Beginn auch eine Webseite geladen. Zusätzliche Informationen können dann im Hintergrund vom Server geholt werden

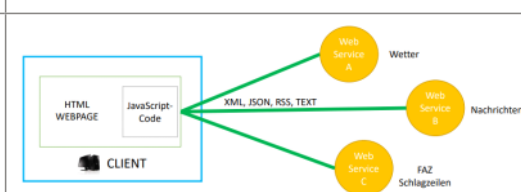


- Aufbau einer AJAX Applikation

- CLIENT Seite kommuniziert über ein Skript (JavaScript) mit dem Server (->Serverscript läuft).
- Das Serverscript kann nun Informationen aus einer Datenbank holen oder andere Quellen anzapfen wie zum Beispiel einen Web-Service.



Es können auch Dienste im Internet direkt angebunden werden.



- Sicherheit kann gewährleistet werden mit: Verschlüsselung, Zertifikaten, Proxyservern

3. HTML-Hypertext Markup Language

- Grundgerüst

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Titel der Einzelseite</title>
7   <meta name="description" content="">
8   <meta name="author" content="">
9   <meta name="keywords" content="">
10
11  <link href="style.css" type="text/css" rel="stylesheet">
12 </head>
13
14 <body>
15
16 </body>
17 </html>

```

4. CSS-Cascading Style Sheet

- 3 Funktions Möglichkeiten
 - o Inline Style
 - Gültigkeit bezieht sich nur auf das zugewiesene Element
 - o Eingebettet
 - Gültigkeit beschränkt sich auf die Seite, wo der Style definiert ist
 - Können im Kopfbereich platziert werden
 - o Verlinkt
 - Das Stylesheet kann in mehreren Seiten eingefügt und verwendet werden
 - Hat einen globalen Gültigkeitsbereich
 - Hat den Vorteil, dass das Erscheinungsbild komplett vom strukturellen Markup getrennt ist
- ~~Unterschied zwischen Klassen & ID's~~
 - o IDs sind leistungsfähiger (Es darf nur eine Instanz einer ID innerhalb eines Dokumentes sein!)
 - o JavaScript verwendet die IDs, um Element anzusprechen.
 - o Mehrmals die gleiche Klasse, darf auf verschiedene Elemente in einem Dokument angewendet werden
- Klassen
 - o Klassenselektoren beginnen immer mit einem Punkt!
 - o gruppieren mehrere Elemente
 - o Vorteil, dass alle Elemente einer Klasse im CSS angesprochen und einheitlich designt werden können
- ID's
 - o Klassenselektoren beginnen immer mit einem Hashtag!
 - o bestimmt ein einzelnes Element
 - o eindeutige Attribute
 - o bestimmte ID im gesamten Code auch nur einmal definiert werden

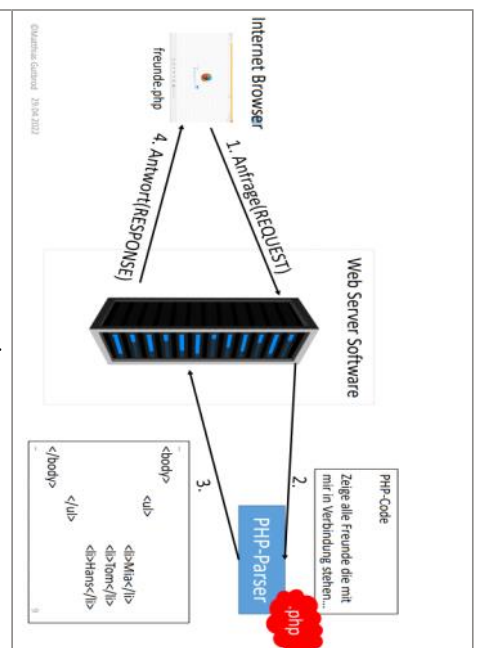
5. Dynamische Webseiten mit php

- kann ihren Inhalt je nach Struktur in Form eines Baukastensystems ändern.
- beziehen ihre Inhalte und Daten von verschiedenen Quellen
- Das erfolgt vor allem mit der Hilfe von Servern, auf denen Codes aus PHP oder JavaScript (über Node.js) liegen. Diese werden miteinander und mit relevanten Dateien verbunden.
- rufen Daten ab, während sie online sind
- Die Seite und Ihre Inhalte werden für jeden Besucher neu generiert.
- Daten in der Datenbank könnten bei jedem Aufruf verändert sein.
- Seite an sich, auf der die Daten im Internet dargestellt werden, bleibt gleich - die Daten selbst sind neu oder erweitert.
- Beispiele: Webshop, Gästebuch, Schnittstellen mit Datenaustausch und vieles mehr.

```

<?php
    ....
?>

```



6. ~~Unterschied zwischen dynamischen & responsiv Webseite~~

- das Layout einer Website so flexibel gestaltet, dass dieses auf dem Computer- Desktop, Tablet und Smartphone eine gleichbleibende Benutzerfreundlichkeit bietet
- Inhalts- und Navigationselemente & struktureller Aufbau einer Website passen sich der Bildschirmauflösung des mobilen Endgeräts an
 - es reagiert auf und korrespondiert mit der Auflösung des mobilen Endgeräts

7. Begriffe erklären

- Angular JS
 - o Ein clientseitiges JavaScript-Webframework für dynamische Webapplikationen
 - o Für die Entwicklung von Single-Page-Apps gedacht
 - o Softwareentwicklung und das Komponententesten können damit vereinfacht
 - o Entwickelt von Google und dort im Produktiveinsatz
- React
 - o React ist eine JavaScript-Softwarebibliothek, die ein Grundgerüst für die Ausgabe von User-Interface-Komponenten von Webseiten zur Verfügung stellt.
 - o clientseitig
- WordPress
 - o ist ein freies Content-Management-System (CMS)
 - o bietet jedem Einsteiger die Möglichkeit vollkommen ohne Programmierkenntnisse Inhalte im Web zu veröffentlichen
 - o serverseitig
- Node JS
 - o Node.js ist eine plattformübergreifende Open-Source-JavaScript-Laufzeitumgebung, die JavaScript-Code außerhalb eines Webbrowsers ausführen kann
 - o Damit kann zum Beispiel ein Webserver betrieben werden
 - o severseitig

8. Was ist eine Session

- Sessions bieten eine einfache Möglichkeit, Daten für individuelle Benutzer unter einer eindeutigen Session-ID zu speichern. Sie können verwendet werden, damit Daten zwischen Seitenanforderungen bestehen bleiben.
- Unterschied zu Cookies-> Daten werden nicht auf dem Client gehalten! sondern auf dem Server !
- In Sessions speichert man Daten während einer Browsersitzung
- SessionId wird im Cookie gespeichert!
 - > bei vollständiger Deaktivierung von Cookies funktionieren Sessions nicht!

9. Wie funktionieren SESSIONS

- Zu jeder Session => eindeutige Session-ID.
- Diese wird per Cookie beim Besucher hinterlegt oder als Parameter an die URL angehängt.
- Damit kannst Du verschiedene Besucher unterscheiden bzw. kriegt jeder seinen eigenen Wert zugewiesen.
- Im Hintergrund werden alle Daten, die zu der Session gehören in Dateien oder in einer Datenbank gespeichert.
- Am Anfang des Scripts muss session_start() aufgerufen werden. Sonst kann man nicht drauf zugreifen.
- Die Session wird dabei nur auf dem Server gespeichert, wo PHP läuft.
- Der Client kriegt davon nichts mit. Nur was Dein Script draus macht, kriegt der geliefert.

10. ~~Unterschied: Const und let~~

- LET Variablen sind nur in Blöcken zugreifbar, aber auch in Unterblöcken
 - > let foo = 1; // Deklaration + Initialisierung
- CONST verhindern, dass ein Wert überschrieben werden kann:
 - > const a = 5; // Nur Deklaration + Initialisierung möglich!

11. ~~JSON~~ - JavaScript Object Notatio

- Beschreibungssprache -> man kann Informationen mit dieser Sprache speichern und von Computer/System

- zu Computer/System senden
- ist von Programmiersprachen unabhängig
- Wird zur Übertragung und zum Speichern strukturierter Daten eingesetzt
- Dient als Datenformat bei der Datenübertragung (Serialisierung)
- Insbesondere bei Webanwendungen und mobilen Apps wird es in Verbindung mit JavaScript, Ajax oder WebSockets zum Übertragen von Daten zwischen dem Client und dem Server häufig genutzt

11. Was ist ein Hashwert

- Prüfsumme: Hashwerte sind Prüfsummen, die für die Verschlüsselung von Nachrichten mit variabler Länge verwendet werden. So wird jeder Nachricht ein ganz bestimmter Hashwert zugeordnet. Hashwerte sind Fingerabdrücke eines sehr langen Datensatzes.
- Kollision: Es dürfen keine zwei verschlüsselte Nachrichten vorkommen, die denselben Hashwert besitzen. In diesem Fall liegt eine Kollision vor.
- Rückschlüsse: Vom Hashwert aus können keine Rückschlüsse auf die gegebene Nachricht schließen. Wenn zum Beispiel Passwörter abgespeichert werden, kann man nicht mehr auf die Passwörter schließen, da deren Hashwerte abgespeichert werden.
- Vergleichen: Man können mit dem Hashwert vergleichen, ob zwei Datensätze identisch sind oder nicht. Dabei wird überprüft, ob ein Datensatz von einer Person geändert wurde. So kann ein Virens Scanner feststellen, ob eine Datei von einem Virus infiziert wurde und entsprechend eine Änderung der Datei vorliegt

12. ~~Wie funktioniert Hash~~

- Hashwert beschreibt die Verarbeitung des Inhaltes einer Datei, um diesem einen eindeutigen numerischen Wert zuzuweisen. Anhand des numerischen Wertes können Suchalgorithmen den Inhalt einer Datei identifizieren. Jeder Wert ist also einmalig und kann deswegen eindeutig einer Datei zugeordnet werden.

13. POST & GET(Unterscheiden & Verstehen)

GET	POST
Werte in der URL sichtbar.	Werte in der URL nicht sichtbar.
GET hat eine Begrenzung der Länge der Werte, im Allgemeinen 255 Zeichen.	POST hat keine Begrenzung der Länge der Werte, da sie über den Hauptteil von HTTP gesendet werden.
Die Leistung besser, da die Werte einfach an die URL angehängt werden.	Geringere Leistung, da Zeit für das Einfügen von POST-Werten in den HTTP-Text aufgewendet wird.
Unterstützt nur Zeichenfolgendatentypen.	Unterstützt verschiedene Datentypen wie String, Numerisch, Binär usw.
GET-Ergebnisse können mit einem Lesezeichen versehen werden.	POST-Ergebnisse können nicht mit einem Lesezeichen versehen werden.
GET-Anforderungen können häufig zwischengespeichert werden.	Die POST-Anforderung ist kaum zwischenspeicherbar.
GET-Parameter bleiben im Verlauf des Webbrowsers.	Parameter werden nicht im Verlauf des Webbrowsers gespeichert.
Überträgt die Variablen in Adresszeile (URL)	Die Variablen und deren Werte werden über die Adresszeile übergeben.
<code>GET/RegisterStudent.asp?user=value1&pass=value2</code>	<code>POST/RegisterStudent.asp HTTP/1.1</code> <code>Host: www.guru99.com</code> <code>Benutzer=Wert1&pass=Wert2</code>

14. ~~Was ist TypeScript~~

- TypeScript ist eine Obermenge von typisiertem JavaScript (optional), die bei der Erstellung und Verwaltung umfangreicher JavaScript-Projekte helfen kann. Es kann als JavaScript mit zusätzlichen Funktionen wie starker statischer Typisierung, Kompilierung und objektorientierter Programmierung betrachtet werden.

15. PHP

Gibt einen oder mehrere Strings aus	<code>echo "Fehler".\$e;</code>
Baut eine Verbindung zur Datenbank auf -neues Objekt der Klasse PDO-> drei Parameter	<code>\$pdo=new PDO("mysql:host=localhost;dbname=". \$dbDatabasename,\$dbLoginUsername,\$dbPassword);</code>
Sql Befehl wird definiert	<code>\$sql="SELECT * FROM fpfriendss WHERE uid=". \$_SESSION['id'];</code>
Hier wird mit dem Ergebnis der Sql-Abfrage durch die Datensätze iteriert und die Daten werden in die Variable \$row gespeichert. Also werden hier der Vorname, Nachname, die Stadt, ... der Freunde ausgegeben deren UserId mit der jetzigen SESSION_ID übereinstimmt	<pre>foreach(\$pdo->query(\$sql) as \$row) { \$sFirstname=\$row['firstname']; \$sLastname=\$row['lastname']; \$sCity=\$row['city']; \$sStudyCourses=\$row['studycourse']; \$iSemester=\$row['semester']; }</pre>
Verbindung wird geschlossen	<code>\$pdo=null;</code>

16. Button: Submit und Onclick

Onclick event <ul style="list-style-type: none"> ◦ Ereignisse werden ausgelöst, wenn der Benutzer mit der Maus auf etwas klickt 	<code><button onclick="myFunction()">Click me</button></code>
Submit <ul style="list-style-type: none"> ◦ Der Submit-Button ist das finale Element eines Formulars: Der Klick übermittelt die Daten des Formulars an die Anwendung – fast immer auf eine serverseitige Anwendung. ◦ Das Attribut type="submit" macht aus einem Eingabefeld einen graphischen Button. Wenn der Nutzer auf den Button klickt, leitet der Browser die eingegebenen Daten zu der URL aus dem action-Attribut des form-Tags. 	<code><button type="submit" value="Submit"> Submit</button></code>
Onsubmit event <ul style="list-style-type: none"> ◦ Wenn der Benutzer ein Formular absendet, feuert der Browser ein submit-Event. Javascript kann das submit-Event abfangen und die Übermittlung der Daten verhindern. ◦ So kann das Script die Formulareingaben vor der Übertragung der Daten prüfen und die Übertragung aussetzen, bis alle erforderlichen Felder korrekt ausgefüllt wurden. 	<code><form onsubmit="myFunction()"> Enter name: <input type="text"> <input type="submit"> </form></code>

17. Java Script Funktion(Wert irgendwo rausholen/löschen)

- Vorteile von Javascript:
 - Die Funktionen können auch von Stellen aus ausgerufen werden, die vor dem Funktionsdeklarations-Statement stehen.
 - Das Script wird nicht auf dem Webserver ausgeführt, sondern auf den Seiten des Internetnutzers durch einen Browser. Somit werden Rechenleistung beim Server und Kommunikationswege gespart.

Dadurch fällt auch das ständige Anfragen und Antworten zwischen Client und Server weg.

<p>Using pop () (Es entfernt das letzte Element aus einem Array und gibt dieses Element zurück.) ->Um das letzte Element zu entfernen, müssen Sie folgende Schritte durchführen</p> <p>Das Ergebnis ist: ["1", "2", "3"]</p>	<pre><!DOCTYPE html> <html> <body> <p id="element"></p> <script> varmyArray = ["1", "2", "3", "4"]; myArray.pop(); document.getElementById("element").innerHTML = myArray; myArray.pop(); </script> </body> </html></pre> <p><i>document.getElementById("element").innerHTML = myArray.pop();--> um gelöschtes Element zurückzugeben</i></p>
<p>Using shift() (Es entfernt das erste Element eines Arrays und kehrt dieses Element zurück.) Um das erste Element eines Arrays zu entfernen, verwenden Sie folgenden Code</p> <p>Das Ergebnis wird folgende sein: [2,3,4]</p>	<pre><!DOCTYPE html> <html> <body> <p id="element"></p> <script> var myArray = ["1", "2", "3", "4"]; myArray.shift(); document.getElementById("element").innerHTML = myArray; </script> </body> </html></pre> <p><i>document.getElementById("element").innerHTML = myArray.shift();--> um gelöschtes Element zurückzugeben</i></p>
<p>Using splice() (Hinzufügen/Entfernen von Elementen zu/von einem Array)</p> <p>An 2. Position 1 Item entfernen</p> <p>Das Ergebnis wird folgende sein: [a,b,d].</p>	<pre><!DOCTYPE html> <html> <body> <p id="element"></p> <script> var myArray = ['a', 'b', 'c', 'd']; myArray.splice(2, 1); document.getElementById("element").innerHTML = myArray; </script> </body> </html></pre>
<p>splice() zusammen mit einem indexOf() entfernt die Item(s) aus dem Array.</p> <p>Das Ergebnis ist [a,b]</p>	<pre><!DOCTYPE html> <html> <body> <p id="element"></p> <script> var myArray = ["a", "b", "c", "d"]; myArray.splice (myArray.indexOf('c'), 2); document.getElementById("element").innerHTML = myArray; </script> </body> </html></pre>

