# Web Engineering

Inhalt

[Web Engineering 1](#_Toc407980088)

[Entwicklung 3](#_Toc407980089)

[IPv4 3](#_Toc407980090)

[NAT (Network Address Translation) 3](#_Toc407980091)

[Client / Server Konzept 4](#_Toc407980092)

[HTML 4](#_Toc407980093)

[http 4](#_Toc407980094)

[Unidirektionale Links 5](#_Toc407980095)

[Nichtlineare Textsammlung: 5](#_Toc407980096)

[URL 5](#_Toc407980097)

[Webserver – Datenbank – CMS „Content Management System“ 5](#_Toc407980098)

[Content Management System (CMS) 6](#_Toc407980099)

[Standortbestimmung Komponenten und Templates eines CMS 6](#_Toc407980100)

[Unterschied zwischen GET und POST 7](#_Toc407980101)

[GET wird in der URL angezeigt 7](#_Toc407980102)

[GET ist begrenzt, POST nicht 7](#_Toc407980103)

[Vor- und Nachteile von post und get 8](#_Toc407980104)

[Unterschied Radio-Button und Checkboxen 8](#_Toc407980105)

[Querystring 8](#_Toc407980106)

[Aufbau 8](#_Toc407980107)

[Übertragung 8](#_Toc407980108)

[Beispiel 8](#_Toc407980109)

[Auswertung (PHP) 9](#_Toc407980110)

[Nachteile und Probleme 9](#_Toc407980111)

[Kodierung 9](#_Toc407980112)

[Sicherheitsrisiken 9](#_Toc407980113)

[Technische Beschränkungen 9](#_Toc407980114)

[Cookies und Sessions 10](#_Toc407980115)

[Anwendung Cookie: 10](#_Toc407980116)

[Definition Cookie: 11](#_Toc407980117)

[Cookie-Konzept 12](#_Toc407980118)

[Session: 13](#_Toc407980119)

[Notportion zu Sessions 13](#_Toc407980120)

[Authentisierung und dezentrale Konfiguration 14](#_Toc407980121)

[HTTP Authentisierung 14](#_Toc407980122)

[Dezentrale Konfiguration 14](#_Toc407980123)

[FTP – Server 15](#_Toc407980124)

[Aktiv/Passiv FTP 15](#_Toc407980125)

[Aktives FTP 15](#_Toc407980126)

[Passives FTP 15](#_Toc407980127)

[Mail 18](#_Toc407980128)

[Wofür brauche ich dieses POP oder IMAP? 18](#_Toc407980129)

[Was ist POP? 18](#_Toc407980130)

[Was ist IMAP? 18](#_Toc407980131)

[SMTP 18](#_Toc407980132)

[Header anzeigen 19](#_Toc407980133)

[Mime 19](#_Toc407980134)

[JavaScript 20](#_Toc407980135)

[Ajax 20](#_Toc407980136)

[JSON 20](#_Toc407980137)

**Einführung: Anbindung ans Internet**

### Entwicklung

1969 entstand das ARPANET (Advanced Research Project Agency) welche eine Abteilung des amerikanischen Verteidigungsministeriums war. Somit wurde es primär für militärische Zwecke verwendet.

Nach und nach wurde das ARPANET an verschiedenen Unis und Forschungseinrichtungen eingesetzt.

1972 gab es den ersten Boom 🡪 Das Email wurde geschaffen. Programm zum Senden und Empfangen von Emails.

1973 kam das TCP (Transmission Control Protokoll) „gibt es heute noch“.

1980er Jahre Trennung zwischen ARPANET und MILNET

1991 Grundidee für WWW wurde geschaffen 🡪 öffentlicher Zugang wurde gewährt durch ISPs

1993 Mosaic als Webbrowser lanciert 🡪 Nutzung des des graphischen WWW 🡪 Anzahl Nutzer stieg rasant an

1994 Netscape Einfach Nutzung für Amateur-User

Das Web ist ein Verbund von Netzen welche sich an dieselben Regeln halten müssen, das sogenannte Internet Protocol = IP. Die IP-Adresse ist die gemeinsame Adresslogik. Jeder hat eine IP-Adresse woran man Ihn/Sie identifizieren kann im Internet. Die IP wechselt in den meisten Fällen regelmässig.

TCP = Transfer control protokoll. Kann Datenpakete nummerieren und quittieren.

IP definiert im Wesentlichen Adress-Schema, Paketformat, etc

ICMP = Internet Control Message Protocol 🡪 Verkehrspolizei 🡪 ist ein Steuerungsprotokoll.

### IPv4

IPv4 benutzt 32-Bit-Adressen, daher sind maximal 4.294.967.296 eindeutige Adressen möglich. IPv4-Adressen werden üblicherweise dezimal in vier Blöcken geschrieben, zum Beispiel 207.142.131.235. Je Block werden 8 Bit zusammengefasst; somit ergibt sich für jeden Block ein Wertebereich von 0 bis 255. Bei der Weiterentwicklung IPv6 werden 128-Bit-Adressen verwendet.

Eine IP-Adresse unterteilt sich in einen Netzwerkteil und einen Host-(Adressen-)teil. Rechner sind im selben IP-Netz, wenn der Netzwerkteil ihrer Adresse gleich ist – das ist eine Voraussetzung, dass diese Rechner direkt miteinander kommunizieren können, also z. B. über einen Hub, einen Switch oder mittels eines Crosslink-Kabels. Im selben Netz darf keine Host-Adresse doppelt vergeben sein.

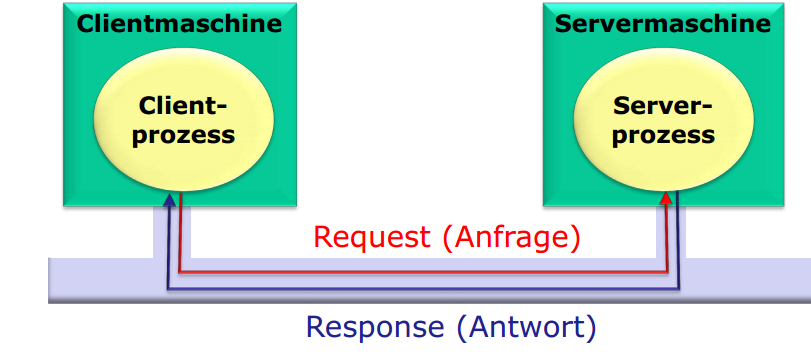
### NAT (Network Address Translation)

NAT erlaubt das Betreiben des privaten Netzes hinter derselben IP-Adresse. Jede Verbindung zum Internet wird durch den NAT-Router mit seiner Absenderadresse versehen.

Mi, 01.10.14

### Client / Server Konzept

* Server wartet auf Anfrage
* Server ist ein passiver Teil (Stand by)
* P2P ist eine Mischung von Clientprozess und Serverprozess



*Webseite ist nicht gleich Website*

Website ist der komplette Auftritt einer Website (site = Platz)

Die Webseiten sind die Seiten einer Website

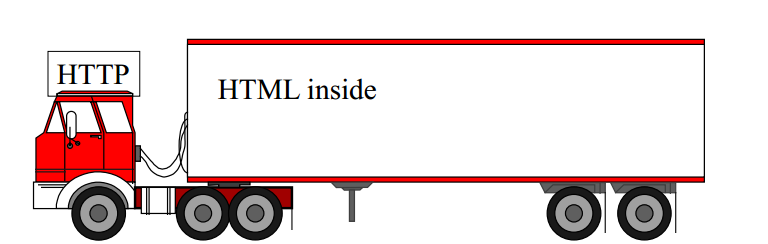
Homepage ist nur die erste Seite die ich aufrufe

### HTML

* Elemente beschreiben was danach passiert 🡪 Vorteil gegen Word 🡪 Ist kompatibel mit allen Betriebssystemen nicht wie bei Word. Ist eine klar definierte Formatierung
* Jedoch muss man Browsertests machen damit alle kompatibel sind
* Markup-Sprache: begrenzte Menge von Auszeichnungselementen
* eigentlich nicht für detaillierte Formatierungsanweisungen gedacht

### http

* Hypertext transfer protocol
* Ist das Protokoll um einen Hypertext zu transferieren



### Unidirektionale Links

Links die immer nur in eine Richtung navigieren

Damit man von einer Nebenseite zurück zur Homepage kommt braucht man z.B. eine Navigationsleiste

### Nichtlineare Textsammlung:

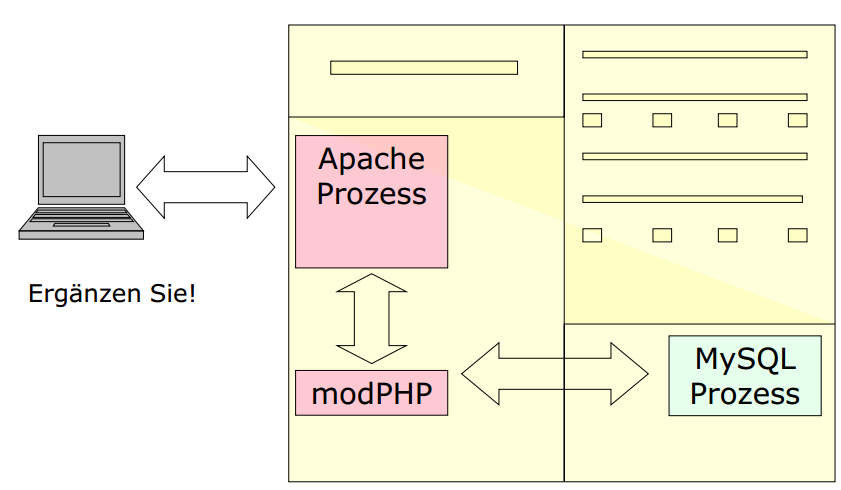
Wikipedia als Beispiel: Man kann irgendwie navigieren. Eine Buch ist linear Geschichte beginnt auf Seite 1 und endet auf Seite x. Auf Wikipedia kann man immer wieder auf Links klicken.

### URL

* Beispiel [www.google.ch](http://www.google.ch)
* Vollständige Angaben zur Lokalisierung eines Dienstes auf einem Host im Internet



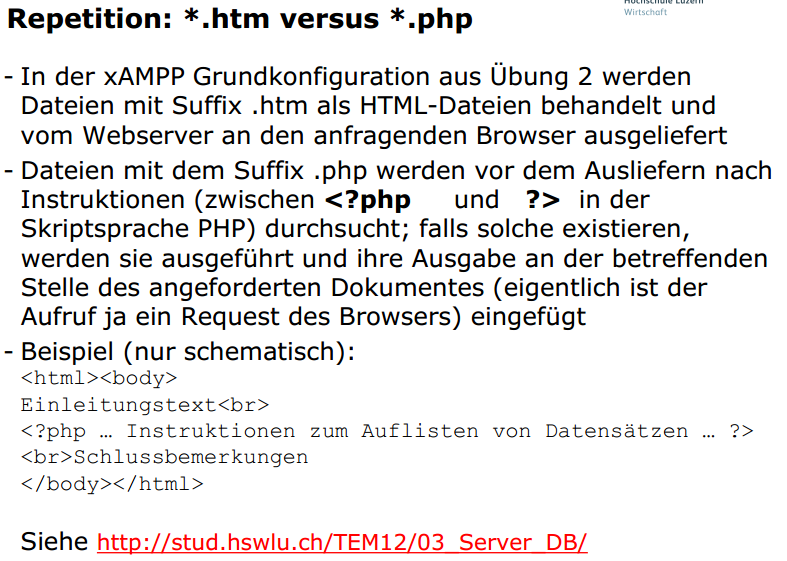
* Das was man oben sieht.



Mi, 08.10.14

### Webserver – Datenbank – CMS „Content Management System“

Ziel eines CMS ist es, dass Autoren ohne grossen Aufwand Inhalte erstellen und publizieren können. (möglichst barrierefrei.)



### Content Management System (CMS)

Eine Software zur gemeinschaftlichen **Erstellung, Bearbeitung und Organisation von Inhalten** (Content). Diese können aus **Text- und Multimedia-Dokumenten** bestehen. Ein Autor kann ein solches System ohne Programmier- oder HTML-Kenntnisse bedienen.

Je nach Systemumgebung ist es nicht effizient, Bilddateien in der DB zu speichern, da das Dateisystem des Webservers dafür meist besser geeignet ist. Joomla schlägt das Verzeichnis Images vor, in das die Dateien hochgeladen werden. Im HTML-Code können sie entsprechend referenziert werden.

Auch die Skripte und statische Seiten von Komponenten, Modulen, Plugins usw. werden im Verzeichnisbaum des Dateisystems abgelegt.

Der eigentliche «Content», d.h. der von Administratoren und Autoren eingegebene Inhalt und die zugehörigen Metadaten werden in Tabellen des Datenbankmanagementsystems gepflegt. Für einzelne Komponenten usw. werden dazu bei Bedarf weitere Tabellen im DBMS angelegt.

Mi, 15.10.14

### Standortbestimmung Komponenten und Templates eines CMS

IP-Stack ist notwendig auf dem Rechner um mit dem Internet zu kommunizieren.

http ist das Protokoll um das Internet zu verbinden.

Html ist was verschickt wird.

WWW ist ein Dienst des Internets.

Xampp ist eine Entwicklungsumgebung um eine Website zu erstellen.

Wieso müssen Konfigurationsdateien angepasst werden?

Frontend ist was der User sieht

Backend ist was der Administrator sieht (immer mit Berechtigungen)

Vererbt Berechtigung für einen Ordner + alle Unterordner.

Mi, 22.10.14

PHP auf Code Academy lösen 🡪 Prüfungsrelevant.

CGI = Das Common Gateway Interface (CGI) ist ein Standard für den Datenaustausch zwischen einem Webserver und dritter Software, die Anfragen bearbeitet. CGI ist eine schon länger bestehende Variante, Webseiten dynamisch bzw. interaktiv zu machen, deren erste Überlegungen auf das Jahr 1993 zurückgehen.

Wenn etwas sehr Zeitnahe passieren muss muss es Aktive sein, weil bei der dynamischen Seite der traffic zu gross wäre.

<http://selfphp.com/praxisbuch/praxisbuchseite.php?site=183&group=32>

* POST- und GET Methoden.

## Unterschied zwischen GET und POST

Beim Formular-Attribut “method” haben wir die Wahl zwischen “get” und “post”, je nachdem welches wir wählen, müssen wir im PHP Script auf die Variable “$\_GET” oder “$\_POST” zugreifen. Doch was ist der Unterschied zwischen den beiden?

### GET wird in der URL angezeigt

Wenn wir ein Formular mit “get” verschicken, wird in der URL-Zeile des Browser die Variablen + ihrem Wert angezeigt:

Diesen Wert kann der Benutzer manipulieren, allerdings ist $\_POST kein Schutz dagegen, aber es sieht in den meisten Fällen “schöner” aus, da die URL-Zeile nicht mit Variablen vollgepumpt ist.

### GET ist begrenzt, POST nicht

Gerade wenn man ein Formularfeld hat, indem sehr sehr viele Daten reinkommen können (z.b. ein Blog-Artikel, ein Forumbeitrag, ein Wikipedia-Artikel) muss man POST verwenden, da die get-Methode begrenzt ist. Bzw eigentlich ist nicht “get” begrenzt, sondern die Url-Zeile des Browsers. Je nach Browser hat man eine maximale Länge um 1024 Zeichen und für Artikel ist das in der Regel nicht ausreichend *(für Twitter-Nachrichten schon ;) )*

### Vor- und Nachteile von post und get

Hier nochmal die Vor- und Nachteile zusammengefasst:

* Bei GET sieht der User, welche Daten übergeben werden (kann man als Vorteil oder als Nachteil sehen)
* Bei GET ist die Länge begrenzt, bei POST nicht
* Die Ergebnisseite eines GET-Formulars kann man bookmarken, da alle nötigen Informationen in der URL enthalten sind
* Die Ergebnisseite eines POST-Formulars kann man weder bookmarken noch im Browser aktualisieren, da die Daten nicht mehr zur Verfügung stehen
* File-Upload ist nur mit POST möglich

### Unterschied Radio-Button und Checkboxen

Radiobuttons sind eine Gruppe von beschrifteten Knöpfen mit gleichem Namen, von denen der Anwender einen auswählen kann. Es kann immer nur einer der Radiobuttons ausgewählt sein.

Eine Checkbox ist eine ankreuzbare (aktivierbare) Option. Sie kann entweder einzeln vorkommen, oder auch als Gruppe von Checkboxen mit gleichem Namen. Bei einer Gruppe kann der Anwender keine, eine oder mehrere Checkboxen ankreuzen.

## Querystring

### Aufbau

Diese URL enthält einen Query String: http://www.example.org/suche?stichwort=wiki&ausgabe=liste Der Query String dieses Beispiels lautet **stichwort=wiki&ausgabe=liste**, der Parameter *stichwort* enthält den Wert „wiki“ und *ausgabe* enthält den Wert „liste“. Innerhalb des Query Strings werden die Parameter-Werte-Paare (auch: *Key-Value Pairs*) mit einem kaufmännischen [&-Zeichen](http://de.wikipedia.org/wiki/Et-Zeichen) getrennt. Die Paare selbst werden durch ein [Gleichheitszeichen](http://de.wikipedia.org/wiki/Gleichheitszeichen) dissoziiert.

### Übertragung

Die Übergabe der Parameter erfolgt über das [HTTP-GET](http://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol#HTTP_GET)-Verfahren. So bildet sich beispielsweise ein Query String aus Parameter-Werte-Paaren, wenn ein [Webformular](http://de.wikipedia.org/wiki/Webformular) abgesendet wurde, das als Übertragung „GET“ benutzt. Ein anderes Verfahren zur Übertragung ist [HTTP-POST](http://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol#HTTP_POST), die Wertepaare sind in diesem Fall dann nicht Teil der URL bzw. des Query Strings.

### Beispiel

Ein Webformular ist wie folgt konzipiert:

<form action="suche.php" method="get">

<label for="stichwort">Suche nach</label>

<input type="text" name="stichwort" id="stichwort" />

<input type="hidden" name="suchdatum" value="2011-09-07" />

<input type="submit" value="Suche starten" />

</form>

Zu beachten ist hierbei das *hidden*-Feld. Es ist für den Benutzer nicht sichtbar, wird beim Absenden des Formulares aber dennoch Teil des Query Strings. Der Submit-Button wird dagegen nicht Teil des Query Strings, da diesem keine Bezeichnung (*name*) zugewiesen wurde. Gibt der Benutzer in das Textfeld „wiki“ ein und sendet das Formular ab, entsteht dieser Query String: http://server/suche.php?stichwort=wiki&suchdatum=2011-09-07

### Auswertung (PHP)

In PHP lässt sich dieser Query String abfragen. Dabei ist der Inhalt des Query Strings als [Array](http://de.wikipedia.org/wiki/Feld_%28Datentyp%29) verfügbar.

echo $\_GET['stichwort'];

gibt beispielsweise den Text aus, der in das Suchfeld eingetragen wurde.

print\_r($\_GET);

ergibt also

Array

(

[stichwort] => wiki

[suchdatum] => 2011-09-07

)

### Nachteile und Probleme

### Kodierung

→ *Hauptartikel:* [*URL-Encoding*](http://de.wikipedia.org/wiki/URL-Encoding)

Einige Zeichen dürfen nicht in einem Query String auftreten, da diese ansonsten fehlerhaft interpretiert werden können. Wird das Gleichheitszeichen beispielsweise innerhalb eines Wertes benutzt, wird dieses fälschlicherweise als Parameter-Werte-Trenner erkannt. Beim Generieren des Query Strings müssen diese Zeichen speziell codiert werden – das Leerzeichen würde dann in ein Pluszeichen (+) gewandelt werden.

### Sicherheitsrisiken

Da der Query-String Teil der URL ist, ist er für jeden Internetnutzer im [Browser](http://de.wikipedia.org/wiki/Browser) einseh- und auch modifizierbar. Der Entwickler der Applikation sollte also nur ungefährliche Parameter in dem Query-String verarbeiten, etwa die Eingabe in einem Suchfeld.

### Technische Beschränkungen

Müssen größere Daten übergeben werden, empfiehlt sich diese via POST zu senden, da für das GET-Verfahren einige Beschränkungen gelten:

* Die HTTP-Spezifikation empfiehlt aus Kompatibilitätsgründen eine Maximalgröße von 255 [Bytes](http://de.wikipedia.org/wiki/Byte) der [URLs](http://de.wikipedia.org/wiki/Uniform_Resource_Locator) bzw. [URIs](http://de.wikipedia.org/wiki/Uniform_Resource_Identifier).[[2]](http://de.wikipedia.org/wiki/Query_String#cite_note-2)
* Der [Internet Explorer](http://de.wikipedia.org/wiki/Internet_Explorer) unterstützt keine URLs, die aus mehr als 2048 Zeichen bestehen[[3]](http://de.wikipedia.org/wiki/Query_String#cite_note-3)
* [Webserver](http://de.wikipedia.org/wiki/Webserver) können die Maximallänge eines Query Strings selbst begrenzen. Bei einer Überschreitung dieser Grenze sendet der Server dann den [Statuscode 414](http://de.wikipedia.org/wiki/HTTP-Statuscode#4xx_.E2.80.93_Client-Fehler).
* Die (mittlerweile veraltete) [HTML](http://de.wikipedia.org/wiki/Hypertext_Markup_Language)-3-Spezifikation schreibt eine Maximallänge von Linkzielen von 1024 Zeichen vor.[[4]](http://de.wikipedia.org/wiki/Query_String#cite_note-4) Diese Beschränkung ist seit HTML 4 nicht mehr vorhanden.

Mi, 29.10.14

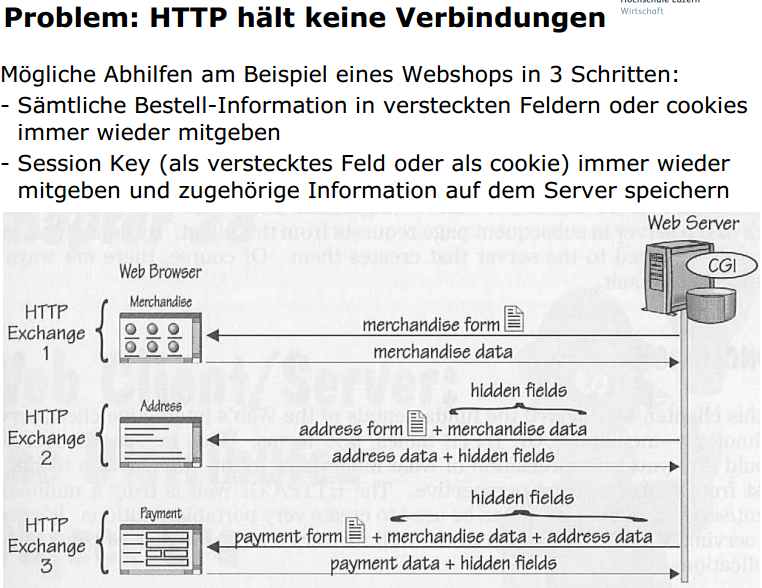
Cookies und Sessions(*Block 6*)

### Anwendung Cookie:

HTTP ist ein zustandsloses Protokoll, daher sind für den Webserver die Seitenaufrufe unabhängig voneinander. Eine Webanwendung, deren Interaktion mit dem Benutzer über mehrere Seitenaufrufe andauert, muss mit Tricks arbeiten, um den Teilnehmer über mehrere Zugriffe hinweg identifizieren zu können. Dazu kann in einem Cookie vom Server eine eindeutige Session-ID gespeichert werden, um genau diesen Client bei weiteren Aufrufen wiederzuerkennen. Aus Sicherheitsgründen wird beim Electronic Banking eher ein Einmal-Token pro Seitenaufruf eingesetzt.

Online-Shops können Cookies verwenden, um Waren in virtuellen Einkaufskörben zu sammeln. Der Kunde kann damit Artikel in den Einkaufskorb legen und sich weiter auf der Website umschauen, um danach die Artikel zusammen zu kaufen. Die Identifikation des Warenkorbs bzw. der Session des Benutzers wird im Cookie abgelegt, die Artikel-Kennungen werden auf dem Webserver diesem Warenkorb bzw. der Session des Benutzers zugeordnet. Erst bei der Bestellung werden diese Informationen serverseitig ausgewertet.

Damit bei Webanwendungen Benutzeraktionen und -eingaben, die für den Server bestimmt sind, bei Abbrüchen der Verbindung zum Server (zum Beispiel in Mobilfunknetzen) nicht verlorengehen, können Cookies zur Zwischenspeicherung eingesetzt werden. Bei Wiederherstellung der Verbindung werden sie vom Server abgefragt. Die Webanwendung erkennt dabei die Reihenfolge, in der die Cookies erzeugt wurden, und markiert bereits verarbeitete Cookies oder löscht deren Inhalt. Weil bei dieser Verwendung unter Umständen viele Cookies erzeugt werden, die frühestens beim Schließen des Browsers gelöscht werden, der Speicherplatz des Browsers für Cookies aber beschränkt ist, muss die Webanwendung Vorkehrungen gegen einen Cookie-Überlauf treffen.



06\_Folie 2

### Definition Cookie:

Eine Definition: http://www.socialnet.de/materialien/internetglossar.html

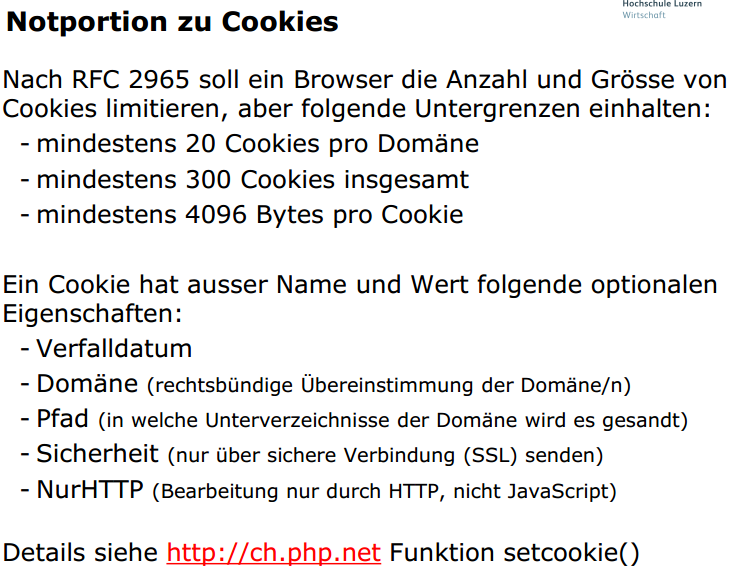
Cookie nennt man kleine Dateien, die ein Browser auf Veranlassung des Webservers auf dem lokalen Rechner speichert, um Informationen über die aktuelle Sitzung bis zum nächsten Besuch des gleichen Webservers aufzubewahren. Sinnvolle Anwendungen sind z.B. Kundennummern oder Präferenzen des Besuchers, um diese bei dem nächsten Besuch automatisch berücksichtigen zu können. Einige Online-Shops funktionieren nur richtig, wenn ein Cookie mit Informationen zur aktuellen Bestellung gespeichert werden kann. In den Sicherheitseinstellungen des Browser kann festgelegt werden, ob Cookies immer, nur auf Nachfrage oder nie zugelassen werden. Zusatzprogramme erlauben eine differenziertere Cookie-Verwaltung. Normalerweise kann nur der Webserver die Informationen in dem Cookie auswerten, der den Cookie angelegt hat. Allerdings haben sich teilweise Betreiber von Webservern zusammengeschlossen, um serverübergreifend Cookies gemeinsam zu nutzen. Da Cookies keine ausführbaren Programme enthalten, bedrohen sie nicht die Sicherheit des lokalen Systems. Sie ermöglichen nur, den Besucher eines Servers mit früheren Besuchen – und den bei der früheren Sitzung preisgegebenen Informationen – in Verbindung zu bringen.

### Cookie-Konzept

* HTTP ist zustandslos: Der Server kann sich nicht an den vorherigen Request eines Browsers „erinnern″. Dank Cookies wird es möglich, verschiedene Requests dem gleichen Browser zuzuordnen
* Cookies sind kleine Informationseinheiten (keine Programme!), die ein Webserver „S″ bei einer HTTPResponse dem Browser zusammen mit der angeforderten Webseite übermittelt
* Browser speichern die Cookies während der aktuellen Sitzung oder bis zum angegebenen Ablaufdatum. Bis dann werden sie jeder weiteren Anfrage an den Server „S″mitgegeben
* In der Regel enthält das Cookie nur eine eindeutige Identifikationsnummer einer „Session″; die zugehörige Information speichert der Server

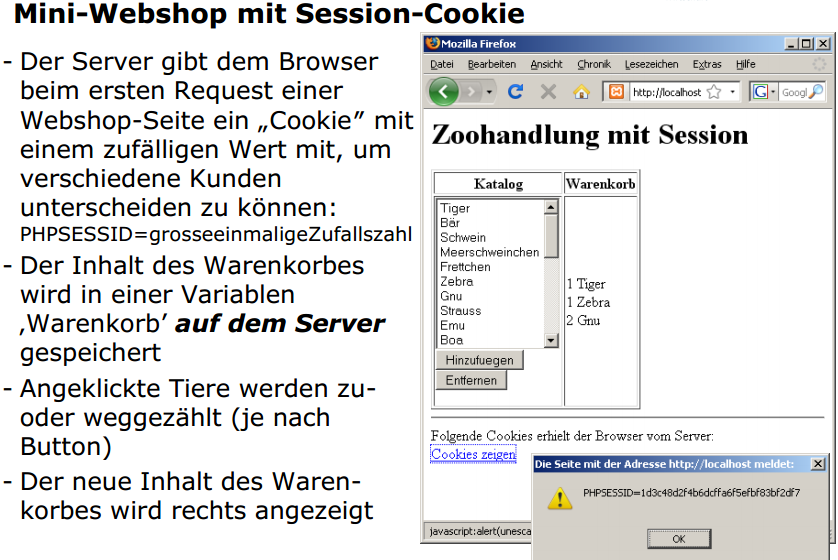
Cookies können das Serve-Verhalten kann ausspioniert werden. (Spam, Marktforschung)

Cookies speichern häufig auch einen Warenkorb.



Request for comment = RFC

### Session:



### Notportion zu Sessions

Sessions:

* Sie dienen zur Überwindung der Zustandslosigkeit von HTTP. Eine Folge von Aufrufen kann vom Server einem bestimmten Browser zugeordnet werden
* Eine eindeutige Session-ID charakterisiert die Beziehung zwischen Browser und Website
* Die Session-ID wird via Cookie oder im Query-String des URL übermittelt
* Die Daten der Session werden auf dem Server gespeichert
* In den Skripts werden sie mit dem Hash $\_SESSION angesprochen:

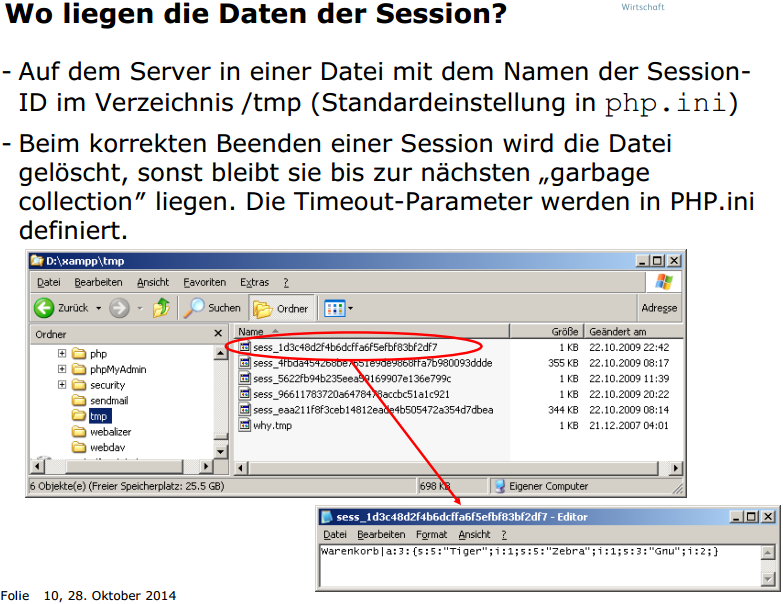
$\_SESSION[`Warenkorb’] = …

* -Nicht alle Datentypen lassen sich in Session-Variablen speichern

Weitere Quellen:

Cookies: <http://www.xirrus.ch/ecm/ecm.cgi?c=topic_cookies.html&l=de>

Sessions: <http://ch.php.net/manual/de/book.session.php>



### Authentisierung und dezentrale Konfiguration

**Authentisierung ist die Antwort auf die Frage „wer bin ich?“**

**Autorisierung ist die Antwort auf die Frage „was darf ich?“**

Selber Codieren bedeutet riesigen Aufwand und beschränkte Sicherheit. Rollenkonzepte werden schnell komplex.

Vom Server beziehen ist selten portabel und u.U. ein Lizenzproblem. Vorteile sind ausgefeilte Log-Mechanismen und Auditing sowie die Integration von Rollenkonzepten aus dem Server-OS.

Joomla ist selber codiert; das ist nötig, wenn die Applikation plattformunabhängig sein soll. In einem CMS ist ein Rollenkonzept User – Autor – Editor – Administrator ein zentrales Thema

Policy 🡪 Auf Computer wird bestimmt wer Zugriff hat und was er machen darf.

### HTTP Authentisierung

* HTTP definiert 2 Authentisierungsverfahren (Basic / Digest)
* Die Konfiguration erfolgt bei Apache i.d.R. in httpd.conf
* Festlegen, welche Zugriffsbeschränkungen für welche Container (z.B. <Directory …> gelten sollen
* Benutzer- und Gruppendateinamen festlegen. Sie liegen in der Regel ausserhalb des Dokumentenbaums, da der Zugriff auf sie für Webuser nicht möglich sein darf
* Passworte werden nicht im Klartext gespeichert, sondern nur ein Hashwert davon (z.B. mit htpasswd.exe) (Hashwert Stichwort 🡪 Quersumme)
* Die Benutzerverwaltung wird damit vom Betriebssystem und dessen Userlizenzen unabhängig
* Eine „dezentrale″ Konfiguration ist sinnvoll, wenn mehrere Websites auf dem gleichen Server laufen (z.B. Hoster); sie wird durch „.htaccess – Dateien″ realisiert (Linux 🡪 Punkt am Anfang heisst versteckt!)

### Dezentrale Konfiguration

Die Datei htaccess.sic definiert den Zugriff und die Konfiguration für das Verzeichnis geheim\_bas und seine Unterverzeichnisse

* Die Datei Benutzer.sic enthält Benutzernamen und PasswortHashes
* Diese beiden Dateien müssen unbedingt vor dem Zugriff durch Web-Benutzer geschützt sein!

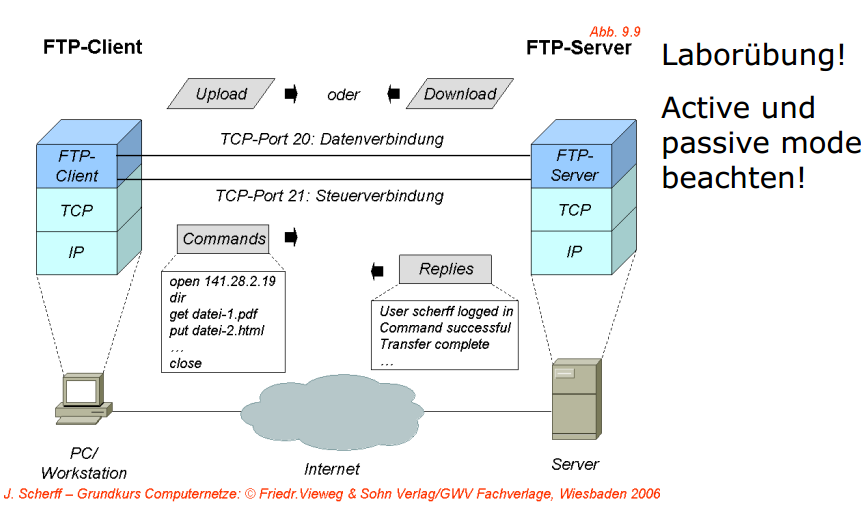
(das Beispiel hier ist rein didaktisch!)

* Achtung:

Bei Basic Authentication wird das Passwort unverschlüsselt (Base64-codiert) übertragen!

### FTP – Server

* Anwendung durch TCP/IP (verbindungsorientiert)
* Verursachte vor 1995 die meisten IP Pakete im Internet
* Zum Transfer von Text- und Binärdaten geeignet
* Ausgeprägtes Client / Server Konzept (in BEIDEN Richtungen!)
* Umfangreicher Befehlssatz (Kommandozeilen)
* Gängige Browser decken einen Teil des Protokolls ab:
  + Syntax: ftp://user:passwort@server.ch/startverzeichnis
  + (leider) immer weniger wegen Missbrauchsmöglichkeiten
  + besser einen FTP-Client verwenden (z.B. FileZilla Projekt)
* Keine Verschlüsselung der übertragenen Befehle / Daten
  + heute oft durch SFTP (Secure FTP) oder SCP (Secure Copy) abgelöst



### Aktiv/Passiv FTP

### Aktives FTP

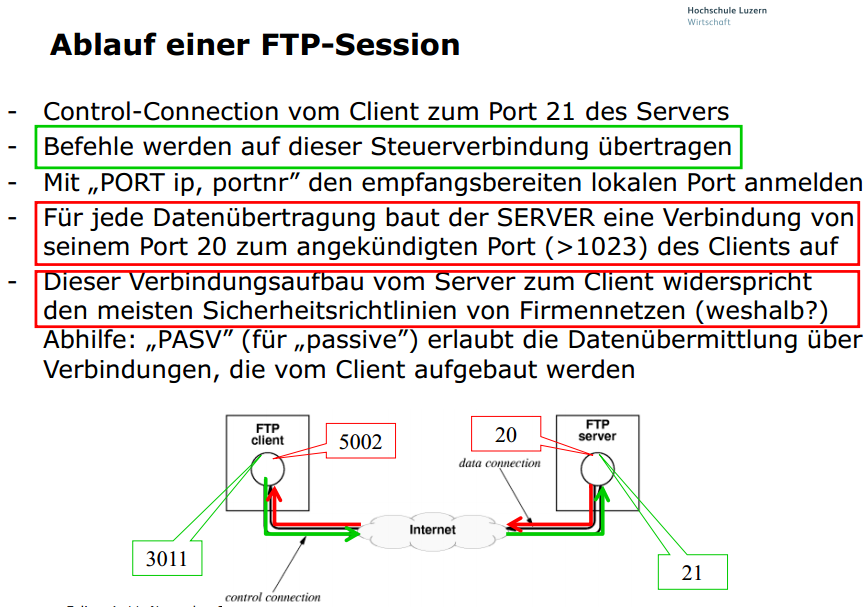
Der Client baut eine Verbindung zum Server auf Port 21 auf. Über diese Verbindung laufen dann die Steuerkommandos wie "welche Dateien sind auf dem Server" u.ä. Das klappt dann auch sehr gut, bis man anfangen möchte, Daten zu übertragen. Dazu teilt der Server dem Client mit "gib mir mal einen freien IP-Port und lausche darauf", um dann eine Verbindung zum Client-Rechner aufzubauen. Die dortige Firewall sieht eine eingehende Verbindung die neu aufgebaut werden soll und verwirft diese, was grundsätzlich keine schlechte Idee ist. In diesem Fall führt es aber dazu, dass die Verbindung "hängt", weil eine zwischengeschaltete Firewall die Daten verwirft.

* Sicher hängt jedoch auch oft. (langsamer, weil die Firewall immer wieder überprüft wer zugreifen weil.)

### Passives FTP

Um dieses Problem (Firewall auf Clientseite) zu lösen, schuf man die Form des passiven FTP, bei der der Client Kontroll- UND Datenverbindung aufbaut, so dass es keine eingehende Verbindung auf den Client mehr gibt. Dazu teilt der Server dem Client mit "ich warte jetzt auf Port xy darauf, dass du eine weitere Verbindung zu mir aufbaust". Dies klappt dann, wenn nicht wieder eine Firewall, diesmal auf Serverseite, zuschlägt, und Verbindungen zu dem Port xy verbietet.

* Ist schneller, jedoch Sicherheitstechnisch eher fragwürdig. (Man umgeht das Problem der Firewall von Fall mit aktivem FTP.)

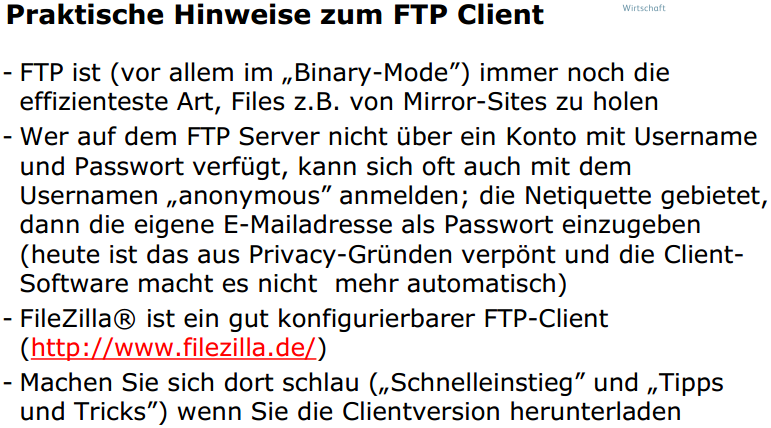
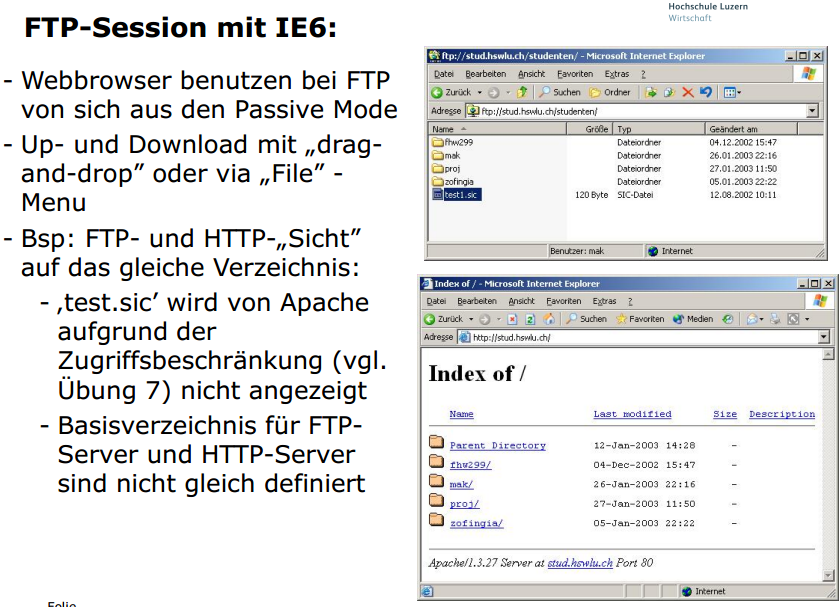


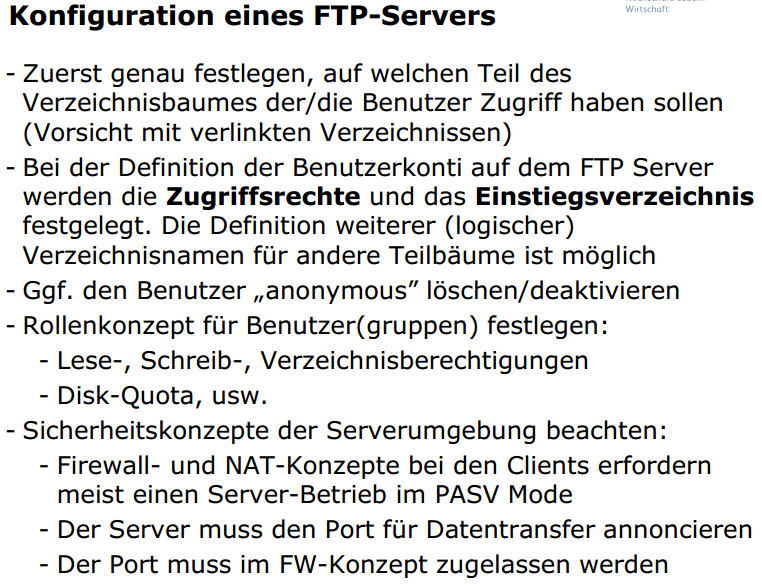
Port = Kanal

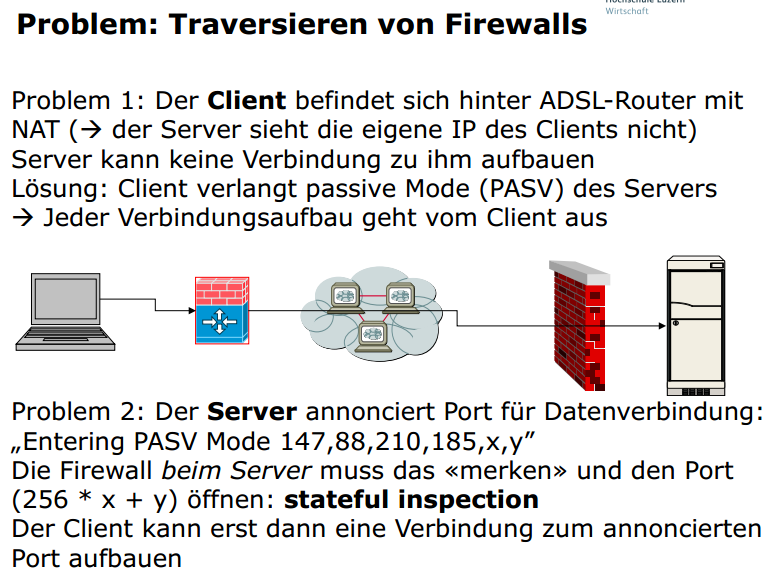
Port verbindet Services

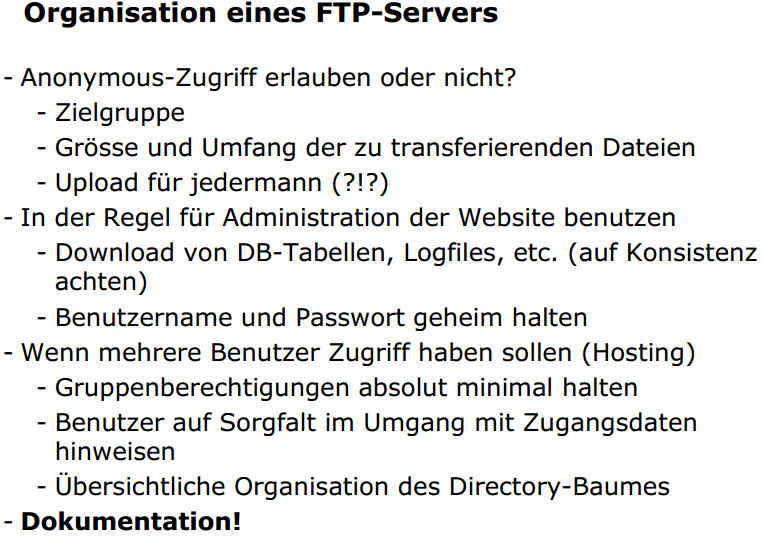
Port 80 🡪 http

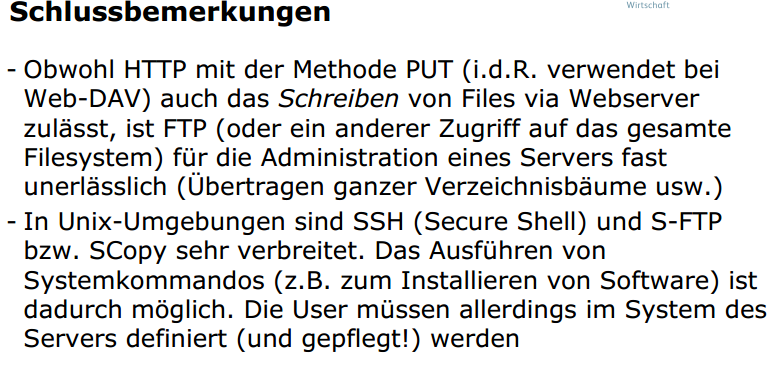
Port443 🡪 SSL



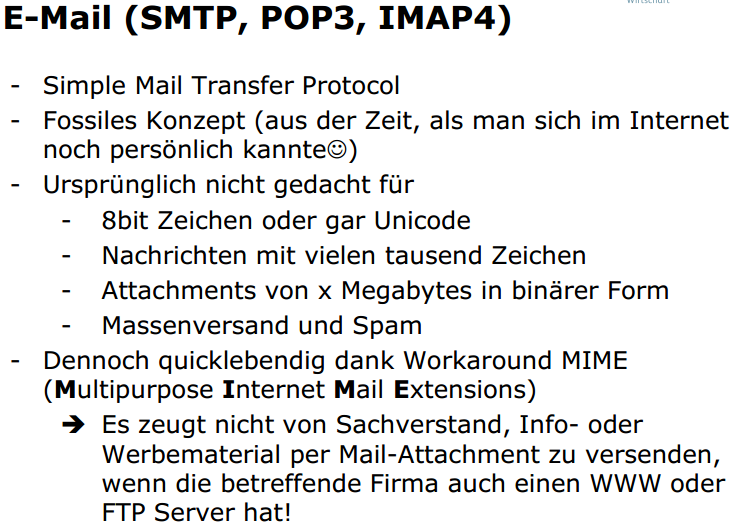








### Mail



### Wofür brauche ich dieses POP oder IMAP?

Wenn man ein E-Mailkonto auf dem PC einrichten will, kommt schnell die Frage, mit welchem Verfahren die E-Mails vom Server heruntergeladen werden sollen. Hier gibt es verschiedene Varianten, aber die gängigsten sind sicher POP und IMAP. Über diese Protokolle kommunizieren die Clients (Outlook, Thunderbird, usw.) mit den Servern (Bluewin, GMX usw.).

### Was ist POP?

POP heisst ausgeschrieben Post Office Protocol. Es ist das Übertragungsprotokoll für E-Mails. Der Client, beispielsweise Microsofts Outlook, ruft die E-Mails mit diesem Protokoll von einem entfernten Mailserver, beispielsweise Bluewin, ab. Diese Technik ist, was die Funktionen betreffen, sehr eingeschränkt. Es dient lediglich dazu, E-Mails abzurufen. Das Protokoll stammt aus dem Jahr 1996 und wird immer mehr durch IMAP abgelöst.

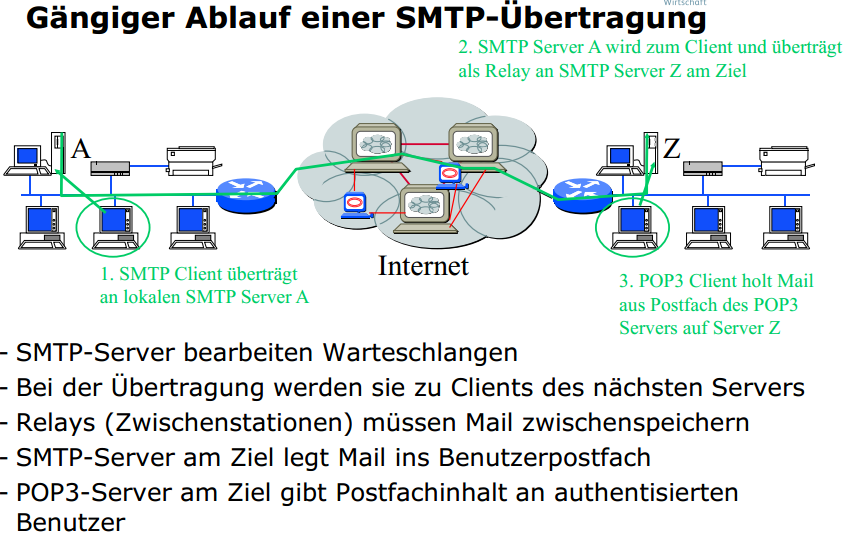
### IMAPWas ist IMAP?

IMAP ist auch nicht mehr ganz neu, es stammt aus 2003. Mit IMAP werden, im Gegensatz zu POP, die E-Mails auf dem Server belassen. Somit ist es möglich, von mehreren Geräten auf die gleichen E-Mails zuzugreifen.  
Hier ein kleines Diagramm, das zeigen soll wie IMAP funktioniert:

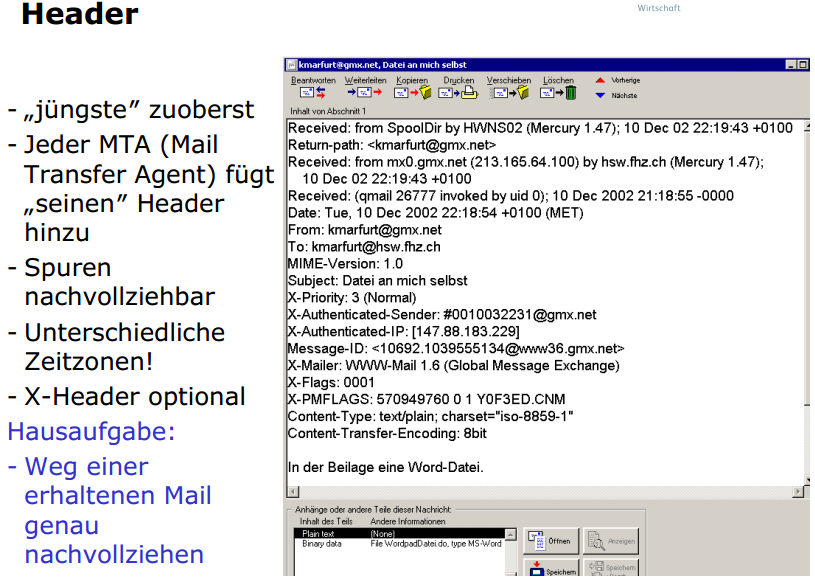
### SMTP

SMTP (Simple Mail Transfer Protokoll) ist ein Übertragungsprotokoll, dass im Unterschied zu POP und IMAP nicht zum Empfangen, sondern zum Senden von E-Mails benötigt wird. Mit SMTP werden E-Mails von Ihrem PC/Laptop, Smartphone oder Tablet zum Sunrise mail Server übertragen und von dort aus versendet.

***Carbon Copy = CC***

**Store and Forward:**

Wird bei jeder Station immer zwischengespeichert.



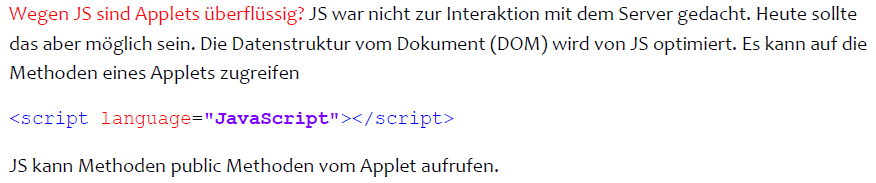
### Header anzeigen

### Mime

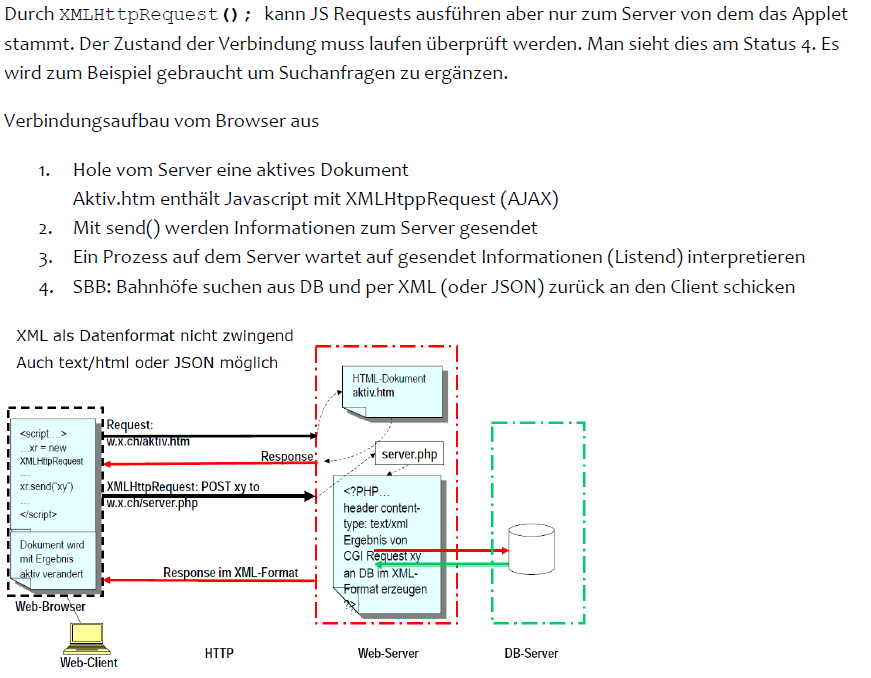
Mime sagt einfach, welches Format im Anhang mitgeschicht wurde. (Bild 🡪 JPEG, GIF/ Dok. 🡪.docx…)

30 % 🡪

### JavaScript



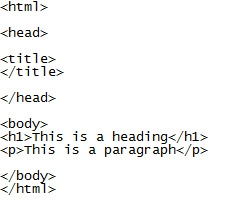
### Ajax



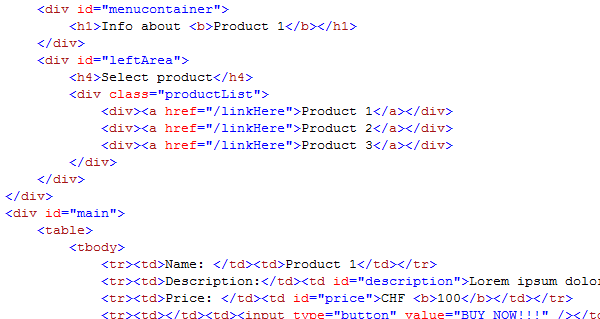
### JSON

Es gibt Webservices die im JSON Format kommunizieren. Generell ist es einfacher zu handhaben als XML.

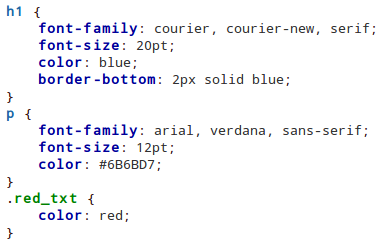
Die **JavaScript Object Notation**, kurz JSON, ist ein kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen. Jedes gültige JSON-Dokument soll ein gültiges JavaScript sein und per „eval()“ interpretiert werden können. Aufgrund kleiner Abweichungen in der Menge der erlaubten Unicode-Zeichen ist es jedoch möglich, JSON-Objekte zu erzeugen, die von einem normkonformen JavaScript-Interpreter nicht akzeptiert werden. Davon abgesehen ist JSON aber unabhängig von der Programmiersprache. Parser existieren in praktisch allen verbreiteten Sprachen.



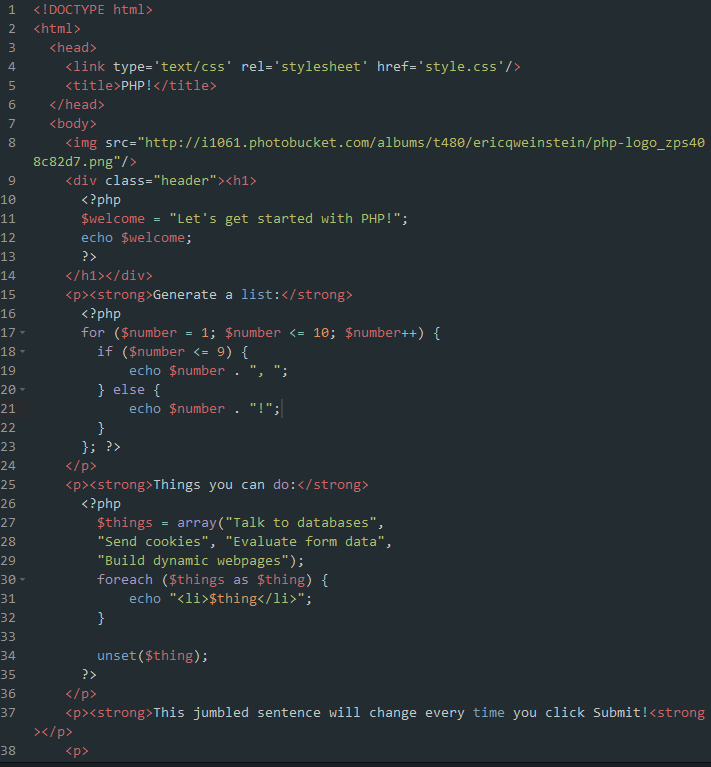
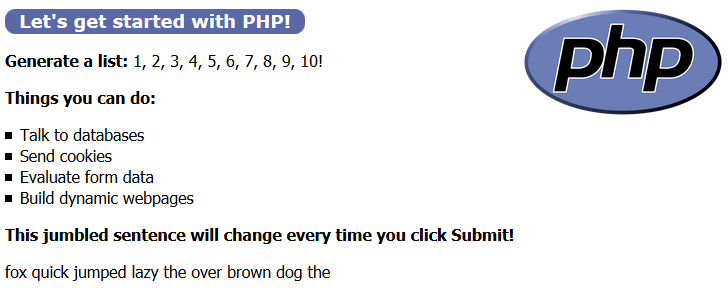
### HTML Basic

Html div 

### CSS Example



### PHP example



**So sieht es aus.**

### Dazu gehört CSS

