

WEB Klausuraufgaben

1. TCP / IP

a. Vorteile erläutern

Funktionalitäten werden vereinfacht dargestellt;

Die Anwendungsschicht muss sich nicht um die physische Verlinkung kümmern, also Aufteilung des Problems

b. Einzelne Schichten erläutern

	Beispiel
	HTTP , UDS , FTP , SMTP , POP , Telnet
	TCP , UDP , SCTP
	IP (IPv4 , IPv6)
	Ethernet , Token Bus , Token Ring , FDDI , IPoAC

Quelle: <http://de.wikipedia.org/wiki/TCP/IP-Referenzmodell#TCP.2FIP-Referenzmodell>

2. Wieso Klassen bei IP Adressen?

Sinnvolle Unterteilung der IP-Adressen: C für kleine Unternehmen/Private Netzwerke, B für große Unternehmen (oder Unis etc.), A für Militär etc.
(--> Class A Netz für 10 - köpfiges Unternehmen... macht keinen Sinn)

3.

a. "Port" anhand der Vorlesung erklären

"Points of entry"

interagieren nicht mit echter Hardware

i.d.R. sind Services direkt mit einem Port verknüpft (z.B. http - port 80)

niedrige Ports sind für spezielle Services reserviert

b. Funktion/Sinn einer Firewall

schützt ein Netzwerk oder einzelne Computer vor unerwünschten Netzzugriffen

i.d.R. Schutz durch Prüfen der Adresse des Kommunikationspartners

c. Grenzen Sicherung über Ports

Über tunneling kann man über Ports anderer Geräte auf den gewünschten Service zugreifen;

Man kann nicht einfach alle oder einen beliebigen Port blockieren. Würde man z.B. einfach port 80 blockieren, könnte man www nicht mehr nutzen.

4.

a. Was ist HTTP?

Hypertext Transfer Protocol

Sprache, in der Web Clients und Web Server miteinander kommunizieren

- b. HTTP-GET und HTTP-POST erläutern
 - GET: Leitet Aufruf einer Website ein (fordert HTTP-Server auf ein Dokument oder Element zu senden)
 - Parameter werden in der URL übermittelt unsicher für Passwort zB
 - Aufbau: GET / pfadname /filename http-Version
 - z.B. GET / welcome.html http/1.1
 - POST: Informationen werden über den Browser am dem HTTP-Server gesendet
 - Parameter werden dem User transparent übermittelt
 - z.B. Übergabe von Formulareinträgen einer HTML-Seite
 - z.B. EINGABEFELD1=test1&EINGABEFELD2=test2
&submit=Abschicken
 - c. Wann macht was Sinn? (Bezug auf b)
 - POST: Bei Anmeldungen mit PW zB damit man das Passwort nicht lesen kann
 - GET: steht z.B. im Browserverlauf, man kann sich Lesezeichen auf Adressen mit GET-Parametern setzen (bei normalen Website Anfragen)
 - d. Wie man einen Webserver “aufsetzt” in 7 Schritten (wie Tafelanschrieb bzw. mündlicher Exkurs)
 - i. wait for connection on port 80
 - ii. read request (text)
 - iii. Analyse and interpret
 - iv. in case of Error, return message
 - v. load resource
 - vi. send request
 - vii. close connection
5. Bei E-Mail - woher bekommt SMTP die IP des Empfängers?
- Hier wird der Domain Name Service (DNS) verwendet. Dieser übersetzt für Menschen lesbare Domains wie “gmail.com” in die zugrundeliegenden IP-Adressen.

6. Ordnerstruktur → Einzelne Datei/Unterordner
 - a. URL (Uniform Resource Locator) absolut angeben
 - i. komplette URL angeben
 - b. URL relativ angeben
 - i. Name der Datei auf die zugegriffen werden soll angeben. Diese wird im aktuellen Verzeichnis gesucht, in dem auch die aktuelle HTML-Datei ist

```
root
+--docs
|   +--api
|       +--new
+--img           +file.html
|               +--logo.gif
|
+--css
    +--def.css
```

relativ: logo.gif (oder \logo.gif ???)
absolut: root\img\logo.gif

7. Adjazenzmatrix gegeben → gerichteten Graphen zeichnen
8. HTML - Code → Adjazenzmatrix angeben und gerichteten Graphen zeichnen

9. Tabelle gegeben (rechts im Bild) → HTML code (html-dateien im Ordner)

a. colspan

b.

10.
HTML

The screenshot shows a code editor with a file named 'test.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     ...
5   </head>
6   <body>
7     ...
8     <table border="1px">
9       ...
10      <tr>
11        <td>A</td>
12        <td colspan="2">B</td>
13      </tr>
14      ...
15      <tr>
16        ...
17        <td>C</td>
18        <td>D</td>
19        <td>E</td>
20      </tr>
21    </table>
22  </body>
23 </html>
```

On the right side, there is a preview of the rendered HTML. It shows a table with a border. The first row has two cells: 'A' and 'B'. The second row has three cells: 'C', 'D', and 'E'. To the right of the table preview, there is a vertical text label 'row span' and a small table with the letters 'A', 'B', 'C', 'D', 'E' arranged in a grid.

Code gegeben

a. Warum werden `<object>` und `<embed>` zusammen verwendet?

ältere Browser kennen object nicht → brauchen embed

b. warum umschließt `<object>` `<embed>` und nicht anders herum?

i. Falls ein ältere Browser den `<object>`-Tag nicht unterstützt (fehlendes PlugIn etc.) wird der Inhalt trotzdem gerendert. (→ embed als Ausweichlösung), da der Browser in den `<object>`-Tag herein geht und den `<embed>` verwendet

ii. Neuere Browser würden den `<embed>`-Tag einfach überspringen und somit auch den `<object>`, der im `<embed>` ist.

11. RGB - Farbenmodell erläutern

Additives Farbenmodell, Basis: Rot - Grün - Blau, Variierungen über Intensität

der

Farben in 2-stelligen Hexzahlen (--> FF_FF_FF), RGB ist Geräteunabhängig - kann sich aber von Gerät zu Gerät in den Farben unterscheiden, bspw. Einsatz: Computermonitore

12.

- a. Tag-Hierarchie gegeben → in HTML umsetzen
- b. + c.: Codezeilen mit div, span usw. → output schreiben/zeichnen (Prioritäten erkennen) -> siehe Datei WE_Klausuren_Aufgabe_CSS.html
Lesetipp: der Workshop auf www.css4you.de
- c. *div span span {border: double;} weg lassen (es passiert nichts) html file unten ist ohne div span span {border: double;}*

body

```
+ div
  -br
  -span
  -br
  + div
    -br
    - span
    - br
    -div
      - br
      - span
      -br
```

```
<!doctype html>
<html>
<head>
<style>
/* für bessere Lesbarkeit */
* {
font-size: 30px;
line-height: 60px;
}

/* aus der Klausur
*/
div * span {border: dotted;}
div * {border: dashed;}
div {border: none ;}

</style>
</head>
<body>
<h1>WE Klausur 1. Semester TINF12-AIBC_MF</h2>

  <div>
    Line 1<br>
    <span>Line 2</span><br>
    <div>
      Line 3<br>
      <span>Line 4</span><br>
      <div>
        Line 5<br>
        <span>Line 6</span><br>
      </div>
    </div>
  </div>
</body>
</html>
```

13. Eval Funktion erklären

- a. ~~Fasst mehrere Parameter (i.d.R. String und int-var) zu einem VariablenNamen zusammen (eval("car", i) in Zählschleife → car1, car2, car3...)~~
Eval kann den Inhalt einer String-Variable als JS-Code interpretieren.
Das heißt, Quelltext kann dynamisch in einem String erzeugt werden und dann mit eval() ausgeführt. Beispiel:
for(i=1; i <= 3;i++) { eval("car"+i+" = 3;"); } → setzt 3 Variablen car1, car2, car3 mit Wert 3

14. Im Zusammenhang mit JavaScript: Was versteht man unter einem

Eventhandler/Eventhandling?

Events sind asynchrone Aktionen auf die der Browser und/oder Scripts reagieren.
Z.B. Mausklick, Seite lädt, etc.

In [computer programming](#), **asynchronous** events are those occurring independently of the main program flow. **Asynchronous** actions are actions executed in a [non-blocking](#) scheme, allowing the main program flow to continue processing.

Ein Eventhandler fängt Aktionen ("Events") ab, die meist durch den Nutzer auf der Seite ausgelöst werden, und kann nach Auslösen der Aktion JS-Code ausführen. Bsp:

<p onclick="machwas()">Hallo</p> -> Nach Klick auf "Hallo" wird die Funktion machwas() aufgerufen.

15. HTML/JS Code → Fehler finden

16. Im Zusammenhang mit JavaScript: Objektorientierung erläutern

- a. Alles ein eigenständiges Object (Funktionen sind Objectfunktionen); außer primitive Typen wie Zahlen
- b. Attribute und Funktionen (object1.essen=myFunction(par1, par2)) können jeder Zeit hinzugefügt werden (assoziative Arrays)
 - i. var a = new Object();
 - ii. z.B. Object.name = "HUGO";
- c. Klassen existieren nur in Form von "Konstruktorfunktionen", Objekte von Typ dieser Methode können erstellt werden.
- d. Prototypmethoden: Da im Gegensatz zu anderen OO Sprachen in JS keine Klassen existieren wird bei einer Funktionsdefinition im Konstruktor zu jedem Objekt eine Instanz dieses Funktionsobjekts erzeugt --> Speicher wird verschwendet.
Lösung: Erstellen einer Prototypmethode mit
Object.prototype.myFunction = function(){}
Greift man nun über eine Objektinstanz auf die Methode myFunction zu findet JS diese in der Konstruktordefinition nicht vor und sucht als nächstes in den Prototypen, wo sie gefunden und verwendet wird. (Nimmt nur an 1 Stelle den

Speicher ein)

- Prinzip Ports und Firewall erklären
 - FTP
 - HTTP
 - Warum eine Firewall nicht ausreicht
 - Prinzip früher → alle Ports auf
 - Prinzip heute → alle Ports zu (dann auf machen welche benötigt)
- RGB erklären und wofür RGB steht, Zusammensetzung der Farben (additiv) ...
- Wie man einen Webserver “aufsetzt” in 7 Schritten (wie Tafelanschrieb bzw. mündlicher Exkurs)
- Video in HTML-Seite alt/neu (Zusammenspiel von <object> und <embed>)
- IPv4 (warum Adressen knapp werden)
 - Funktion Subnetzmaske
 - DNS Server
-
-
-
- HTML Seite aufbauen
 - 1 Aufg mit Tabellen (versch. aufgebaut; einmal drei kleine nebeneinander und ein Element darunter; das ganze noch längs. Zeichnung folgt noch)
 - 2 Aufg mit Ordered List
 - 3 Aufg SPAN DIV BR: Aufbau einer verschachtelten Struktur von Tags und pro neue Zeile sollte der Text “Zeile x” eingefügt werden
- CSS
 - Es waren 4 CSS-Regeln zur 3. Aufgabe bei HTML (span/div) gegeben. (sind bekannt, folgen noch)
 - 1. Aufgabe: Darstellung des Codes mit diesen Regeln zeichnen und kurz erläutern
 - 2. Aufgabe: Darstellung zeichnen, als wenn die letzte der 4 Regeln nicht gegeben wäre und kurz erläutern
- JavaScript
 - Fehler im Code finden
 - Was macht die Funktion eval()

Vorbereitungsvorschläge für die mündliche Prüfung:

- Hier ist ein Link mit Prüfungsfragen

<http://webtech1.wikispaces.com/Klausurfragen>.

- Tipp zu Online-Recherchen: Zu dem Fach “Web Technologien” gibt es mehr Dokumente als zu Web Engineering