1. Kdo vynalezl www ?

**Sir Timothy John Berners-Lee**

2. Pristupnost ?

**Přístupnost = bezbariérovost**

prohlížeč, OS, rozlišení, skriptování
kultura, motorické schopnosti, vidění
extrémně handicapovanými návštěvníky jsou vyhledávací roboti

**Zásady**

validovat
používat informační strukturování
titulek, hierarchie nadpisů, oddělená navigace, linearizace tabulek
psát stručně a srozumitelně
členění textu
zpřístupnit formuláře
**vyhnout se rámcům**

*Moznosti pro pristupnosti v html ?*

**Elementy**

- h*, p; div
- em, strong, q, cite…
- fieldset, legend; optgroup; label for
- th, thead/tfoot; caption

**Atributy**

- title, lang, dir, accesskey
- **specifické**
  - table - longdesc
  - input - title, tabindex
  - img - alt, title, longdesc

3. Co znamena pojem kaskada v css ?

**Kaskáda**: prvek může mít určitou vlastnost definovánu vícekrát

**Ovlivnění kaskády**: !important

Kaskáda

1. Najdi všechny definice pro daný prvek
2. Seřaď je podle váhy (klient < uživatel < autor < autor s !important < uživatel s !important)
3. Pokud stejná váha – seřaď podle specifičnosti (specifičtější selektory mají přednost před obecnějšími)
4. Seřazení podle pořadí definování

**Pr.:**

k dokumentu tagem <link> připojíme několik stylopisů

Pokud více pravidel definuje nějakou vlastnost pro stejný prvek, nejprve se porovná konkrétnost selekce (selektoru u každého pravidla). Přednost má to pravidlo, které prvek popisuje nejkonkrétněji. Konkrétnější jsou pak třídy a pseudotřídy. Nejlépe vystihují prvek identifikátory

elementy v html ?

***Blokové*** *(zalamují odstavec , bloky, tabulky, formuláře )-* **p, h1-h6, ul, ol, li, dl, dt, dd, pre , br**
***Textové*** *(uvnitř blokových, frázové × prezentační )-* ***em, strong, b ,I ,u , sup, abbr***
***Generické*** *( kontejnery, vazba na CSS ) Bez formatovany (**div**-blokovy **span** -radkovy)*
***Obecné atributy*** *(všechny elementy ,id, class, style, title; lang, dir; onSomeEvent )*

4. znacky php ?

```
<?php …… ?>
<? ……. ?>
<script language=”php”> ……. </script>
<% …. %>
```

5. vytvorit object s konstruktorem , ktery vola sam sebe ?

```
class TestClass {
        public static $counter = 0;
        public $id;
                public function __construct() {
                $this->id = self::$counter++;
                }
}
$pom = new TestClass();

                //funkce vola sama sebe
class TestClass {
        public $id;
                public function __construct() {
                $this->id = self::$id;
                }

}
```

6. pripojeni do databaze ?

**$db1=mysql_connect(“localhost”,”root”,”heslo”)**
**$db2=mysql_pconnect(“localhost”,”root”,”heslo”)**
pconnect - persistentni pripojeni
connect – jednorazove pripojeni

## 7. nadefinovat javascript a zavolat ho do html ?

<script language="Javascript"> … </script>

Tri zpusoby jak zaclenit Javascript do HTML

1) pomoci tagu <script> **do proudu dokumentu**
   <script>
   alert('Toto napsal Javascript');
   </script>

*(muze se objevit kdekoliv v kodu,Prohlizec pak script zpracuje hned jakmile na nej narazi)*

2) pomoci tagu <script> a odkazem **na externi soubor**.V externim souboru muze byt napsano toto: document.write('Toto napsal Javascript');
   Do stranky pak bude vypadat kod:
   <script src="externi.js"></script>
   (pouziva se hlavne k nacitani stejneho souboru do vice stranek)

3) **in-line zapis**, radkovy zapis, zapisuje se to jako atribut jineho tagu
   <a href=" http://www.fav.zcu.cz" onmouseover="alert('Toto napsal Javascript')"></a>
   (po prejeti mysi se spusti javascript)

## 8. funkce v AJAXu onreadystatechange.....?

Vlastnost **onreadystatechange** – obsahuje funkci, která zpracuje odpověď serveru

Vlastnost **readyState** obsahuje stav odpovědi
**Vždy, když je hodnota změněna, spustí se** onreadystatechange
0 = požadavek nebyl inicializován
1 = požadavek byl připraven
2 = požadavek byl odeslán
3 = požadavek je zpracováván
4 = hotovo

## 9. pseudotridy ?

*A:HOVER*
*A:AKTIVE*
*A:LINK*
*A:VISITED*

## 10. rozdil HTTP a HTTPS ?

**Protokol** je soubor syntaktických a sémantických pravidel určujících výměnu informace mezi nejméně dvěma entitami spojenými například prostřednictvím počítačové sítě.

HTTP - určený původně pro výměnu hypertextových dokumentů ve formátu HTML

Používá se společně s formátem XML pro tzv. webové služby (spouštění vzdálených aplikací)

Aplikačními branami zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP

HTTP používá jako některé další aplikace tzv. jednotný lokátor prostředků (URL, Uniform Resource Locator), který specifikuje jednoznačné umístění nějakého zdroje v Internetu.

K protokolu HTTP existuje také jeho bezpečnější verze HTTPS, která umožňuje přenášená data šifrovat a tím chránit před odposlechem či jiným narušením.

**Nadstavba HTTPS**

Poskytuje zvýšenou bezpečnost před odposloucháváním či podvržením dat

Data přenášena protokolem HTTP

Jsou šifrována pomocí SSL nebo TLS, což zaručuje ochranu proti *packet-sniffingu* i *man-in-the-middle* útokům

Implicitní port 443 (u HTTP je to 80)

Pro komunikaci pomocí HTTPS musí nejdříve server vlastnit certifikát – digitálně podepsán certifikační autoritou

## 11. vytvorit asociativni pole ?

$ceny = array(‘sako’>=1000,’kalhoty’>=500,’kravata’>=300)

## 12. Stavove kody v HTTP ? (1xx,2xx,3xx,4xx,5xx) ?

1xx – informační

100 continue – klient je informován, že server přijal část požadavku, a může pokračovat při jeho odesílání

2xx – úspěch – požadavek bych přijat a akceptován

200 OK – server vrací data dle metody požadavku

3xx – přesměrování – klient musí provést další akci, aby byl splněn požadavek

4xx – chyba klienta

401 Unauthorized – požaduje se autentizace uživatele

402 Payment Required – rezervováno pro budoucnost

403 Forbidden – server zakazuje vstup na stránku

404 Not Found – server nenalezl pro URI v požadavku žádný dokument

5xx – chyba na straně serveru

500 Internal Server Error – např. vyhození neodchycené výjimky

503 Service Unavailable – dočasné přetížení serveru nebo jeho údržba

# 13. Co je HTML DOM ?

**HTML Document Object Model (HTML DOM)** definuje standardní způsob pro přistup a manipulaci s HTML dokumenty
**DOM** bere **HTML** dokument jako stromovou strukturu s elementy, atributy a textem
**DOM** je W3C standard, 3 úrovně (jedna z nich je HTML DOM)
-Definuje objekty, jejich vlastnosti a metody pro přistup k nim
-Celý dokument je prvek *document*
-Každý HTML tag je prvek *element*
-Text uvnitř HTML elementu je prvek *text*
-Rodiče, potomci, sourozenci

## HTML DOM - vlastnosti

x = document.getElementById("id_elementu")
x.innerHTML – vnitřní text (HTML) elementu
x.nodeName – Název uzlu/elementu
x.nodeValue – hodnota elementu (vlastní text pro textový uzel)
x.parentNode – nadřazený uzel
x.childNodes – pole uzlů-potomků
x.attributes – pole atributů
style objekt – lze měnit css styl elementů

## HTML DOM - METODY

x.getElementById(id) – vrací element se specifikovaným ID
x.getElementsByTagName(name) – vrací pole elementů se specifikovaným názvem tagu
Obě funkce lze kombinovat
x.appendChild(node) – vloží potomka uzlu

x.removeChild(node) – vymaže potomka uzlu

## 14. Objekty prohlizece ?

**window – hlavní a nadřazený objekt všech ostatních**

window.alert(“Text vypsaný metodou alert”);
setTimeout() – poté co uplyne určený čas, provede zadaný kód
open(), close(), prompt(), confirm()
status – obsah (řetězec) stavového řádku
navigator – info o prohlížeči
location – info o url stránky
history – seznam navštívených stránek

**window.document…**

Lze psát pouze document…

**Obsahem je stránka zobrazená v okně**

**Vlastnosti a hodnoty ze zdrojového HTML**

**Formulář se jménem form1 bude přístupný přes document.form1**

**Obsahuje pole jako např. links nebo forms**

**Vlastnosti**

referer - odkud byl dokument načten
title – název stránky
location – url dokumentu

**document.write()**

**Součást objektu dokument**

document.form1.input1.value

**Vlastnosti**

action – kam má být formulář poslán
method – get/post
metody – reset(), submit()

## 15. Datove typy v PHP ?

Integer, Double, String, Boolean, Array, Object

## 17.Napiste alespon 3 priklady konfigurace v .ini ?

*max_execution_time = 30*
*memory_limit = 8M*
*essionauto_start Off*

## *18.jaky konfigurace znate ?*

U každého doménového jména webu se dají na serveru nastavit některé věci. Hlavní ale je, že se to v každé verzi softwarových serverů dělá jinak.

Na Apachi si správce upravuje soubor httpd.conf(*Pro zviditelnění změn se musí server restartovat,* Listen 80 -Port, na kterém webserver poslouchá, ServerName localhost-Adresa serveru (www.kiv.zcu.cz),DocumentRoot - Cesta ke kořenovému adresáři webu ) a uživateli někdy dovolí některá nastavení přenést do soboru .htaccess, který je vedle souborů stránek a autor si je spravuje sám. ***Soubor je skrytý (. na začátku)***

## 19.popiste vlastnost display v css-u ?

block - el. se zobrazí jako blokový
inline - el. se zobrazí jako řádkový
inline-block - el. se zobrazí jako blokový ale bez zlomu na konci. Jdou mu nastavit rozměry
none - el. se nezobrazí

## 20) HTML x XHTML

| | |
|---|---|
| HTML = aplikace SGML<br>Značky<br>◦ case insensitive<br>◦ možno vynechat uzavírací<br>Atributy<br>◦ atribut=hodnota<br>◦ atribut="hodnota s mezerou"<br>◦ atribut<br>Ne-SGML data<br>◦ <![CDATA[ … ]]><br>◦ obvykle stačí komentáře<br>Renderování<br>◦ volná interpretace, tolerance | XHTML = aplikace XML<br>Značky<br>o case sensitive: malými<br>o uzavírací povinně<br>( ...  <img ... />)<br>Atributy<br>o povinné uvozovky<br>o žádná minimalizace<br>Ne-XML data<br>o povinně CDATA sekce<br>o styly, JavaScript atd lépe do externích souborů<br>Renderování<br>o striktní chování |

## 1) K čemu je DOCTYPE v HTML

říká o jakou verzi html se jedná, popřípadě jestli jde o Strict či Transitional
<!DOCTYPE html>

## 2) Novinky v HTML 5

nové elementy- section, article, footer, header
vložení videa do stránky <video>
hudby `<audio>`

## 3) K čemu je label ve formuláři

popisek formulářového pole

## 5) MPO (MVC)

MPO – Model, Pohled, Ovladač
Model - reprezentuje data a práci s nimi
View - slouží k prezentaci dat uživateli
Controler - reaguje na chování uživatele a podle toho mění Model a View

## 6) Rozdíl použití "" a '' v php

'' se zpracovává rychleji, "" mohu psát proměnné do řetězce a php za ně vloží jejich obsah

## 7) Stupně chyb a varování v php

## 8) Cookies / Sessions

cookies se ukládají na straně klienta do prohlížeče, session na server

## 9) Stručně download souboru v php

```
$file_url = 'http://www.myremoteserver.com/file.exe';
header('Content-Type: application/octet-stream');
header("Content-Transfer-Encoding: Binary");
header("Content-disposition: attachment; filename=\"" . basename($file_url) . "\"");
readfile($file_url);
```

## 10) 3 pilíře webu při jeho vzniku

http, html, url

## 12) Jak lze vložit CSS do HTML

<style></style>

atribut style="" u elementu

```
<link rel="stylesheet" type="text/css" href="pokus.css">
```

## 14) Volání normální vs statické metody v php

-> vs :: normální volám nad instancí, statické nad třídou

## 15) Jaká funkce AJAX zpracuje odpověď serveru

onreadystatechange

## 16) Popsat Cross-site request forgery

http://www.zdrojak.cz/clanky/co-je-cross-site-request-forgery-a-jak-se-branit/

## 17) Princip kontroly formuláře přes JavaScript

```
<form name="osobniUdaje" ... onsubmit="return validate();">
<script>
function validate() {
var jmeno = document.osobniUdaje.jmeno.value;
if (jmeno == "") {
alert("zadejte jmeno");
}
</script>
```

## 18) Jak lze uchovat stav mezi požadavky klienta.

session, cookie, proměnné v url

## 19) Kam se v JavaScriptu ukládá odpověď AJAXU (xml, plain text).

responseText

## 20) K čemu je factory funkce + příklad použití.

vytváří objekty (nette) až když je potřebujeme, jde o návrhový vzor továrnička