

1. Nejčastější bezpečnostní chyby
2. Autentizace
3. K čemu je dobré XML?
4. vysvětlete pojem Webové služby
5. Vysvětlete pojem SOAP
6. Popište XHTML
7. Co je to DTD?
8. K čemu slouží kaskádové styly?
9. Jaké 3 typy dokumentu definuje standard HTML 4.01?
10. Jaká je nejnovější verze HTML?
11. Rozdíl mezi HTML 4.0 a XHTML
12. Proč se na stránkách používají metainformace
13. Jaký by měl být dobře navržený web?
14. Co by na žádné stránce nemělo chybět?
15. Jaký je rozdíl mezi bitmapovou a vektorovou grafikou?
16. Jaký je zásadní rozdíl mezi formáty GIF, JPEG a PNG?
17. Jaké formáty obrázků lze vložit do HTML stránky?
18. Jaké formáty grafiky se na webu používají?
19. proč je dobré použít atributy width a height při vkládání obrázků?
20. Co jsou synchronizovaná média a jaký je rozdíl mezi ostatními systémy?
21. Vymenujte co nejvíce programovacích jazyků a vývojových prostředí na straně serveru
22. Popište DHTML
23. K čemu slouží rozhraní CGI?
24. Rozdíl mezi metodami GET a POST
25. K čemu se používají skriptovací jazyky na straně serveru?
26. Co je PHP?
27. Vysvětli pojem Java Server Pages
28. Vysvětlete, co je ASP.NET
29. Jaké jsou největší nedostatky současných technologií používaných pro vývoj webových aplikací?
30. Jaké jsou základní stavební prvky HTML?
31. K čemu se využívají hlavičky v html odesílaných ze strany uživatele(z prohlížeče)?
32. K čemu se využívají hlavičky v html zasílané klientovi jako součást odpovědi?
33. Jak lze nahradit stavovost např. v e-shopech?
34. Jaký je rozdíl mezi skriptováním na straně klienta a serveru?
35. Zásady správné tvorby hesel
36. Jaké funkce mělo nabízet prostředí pro vývoj webových aplikací?
37. Je možné programovat v php bez použití konkrétní databáze?.Jak?
38. Proč není dobré ke zmenšování obrázků na stránkách používat atributy width a height?
39. Jaké znakové sady používá html a xhtml?
40. 3 základní pilíře WWW

Komentář [JK1]: Tento dokument obsahuje moje komentáře ke zpracovaným otázkám k testu z IZ1228, které si můžete stáhnout ze serveru vseborec.cz.
K vytvoření komentářů mě donutil fakt, že otázky jsou zpracovány tak, že za ně spíše zápočet nedostanete.

1. Nejčastější bezpečnostní chyby

Nekontrolování vstupu od uživatele

veškerá data získaná od uživatele by měla být před použitím ověřena
musíme počítat s tím, že uživatel omylem udělá chybu nebo se někdo záměrně snaží nabourat do aplikace

Použití neinicializovaných proměnných

některé jazyky (např. starší verze PHP) automaticky načítají data z požadavku do proměnných
data zvnějšku mohou změnit obsah neinicializované proměnné
pro snížení rizika obdobných chyb nové verze PHP nenačítají data z vnějšku (metody GET a POST, cookies) do proměnných, ale jsou dostupná ve speciálních polích \$_GET, \$_POST, \$_COOKIE apod.

Ochrana session

jediná 100% spolehlivá ochrana je SSL (Secure Sockets Layer) a vypnuté posílání HTTP hlavičky Referer
útok spočívá ve špatné kontrole vstupu a v cross-site skriptování

př. na serveru, kde je uživatel přihlášen, je diskusní fórum
snížení rizika:

všechny odkazy ve vložených příspěvcích přesměrovávat přes pomocnou stránku, která již v URL nemá session-id
dodatečná kontrola session-id (kontrola shody IP adresy, session-id se mění pro každou stránku)

SQL injection

skripty často konstruují SQL dotaz dynamicky na základě vstupů
vstupy se musí pečlivě kontrolovat, aby chybný vstup neumožnil spuštění libovolného SQL příkazu

2. Autentizace

= ověření totožnosti

a) HTTP autentizace

standardní součást protokolu HTTP, nelze změnit podobu přihlašovacího okna, obtížně se řeší odhlášení a automatické odhlášení po určité době, bývá implementována na úrovni webového serveru, hesla jsou přenášena v nekódované podobě, bezpečnější metoda Digest se začíná rozšiřovat až v poslední době

b) Vlastní autentizace

využívá HTML formuláře a session proměnné

mnohem větší flexibilita oproti HTTP – vlastní přihlašovací stránka, hesla uložena na libovolném místě

v session proměnné se uchovávají informace o přihlášeném uživateli a o době jeho posledního přístupu, odhlášení – stačí zrušit session proměnnou

Komentář [JK2]: Hned u druhé otázky mi začaly odpovědi připadat nějak povědomé. A skutečně, ve většině případů se jedná o zkopírované texty ze slidů mých přednášek. Tim se ovšem autor tohoto dokumentu dopustil porušení autorského zákona. Z pohledu studenta však bude zajímavější jiná informace – pokud někdo v testu uvádí doslovné citace textů ze slidů, vypadá to poněkud podezřele a vzbuzuje to dojem, že dotyčný tomu vůbec nerozumí, jen se nazpaměť našrotl nějaké texty.

automatické odhlášení – při každém požadavku se porovnává aktuální čas s časem posledního přístupu (ten je uložen v session proměnné)
pokud klient podporuje JavaScript, lze použít challenge-response mechanismus (heslo není přenášeno v odkrytém tvaru)
pro PHP celý problém velice elegantně řeší knihovna PHPLIB

3. K čemu je dobré XML?

– eXtensible Markup Language
odlehčená verze SGML

Řeší problémy dnešního webu – informační zahlcení, nutnost lepšího vyhledávání, nestandardní rozšíření HTML jednotlivými výrobci prohlížečů

podpora různých zařízení (PC, MT, PDA...)

snadná výměna dat libovolného druhu a spolupráce aplikací v celosvětovém měřítku
v dokumentech můžeme používat libovolně zvolené elementy a atributy

pomocí **DTD** můžeme vytvářet nové jazyky založené na XML

vhodně zvolené elementy přidávají dokumentu informační hodnotu a usnadní prohledávání Webu

XML má striktnější syntaxi než HTML (neukončené tagy, ohraničení atributů uvozovkami apod.), snazší implementace

Komentář [JK3]: Kromě DTD existuje mnoho dalších možností: W3C XML Schema, RELAX NG, ...

4. vysvětlete pojem Webové služby

- webová služba = web-service
- webové služby – souhrnné označení pro sadu technologií umožňujících komunikaci mezi aplikacemi
- webová služba je jednoduchá komponenta nabízející určitou službu
- převod měn, zjištění kurzu akcie, zpracování objednávky, překlad textu
- **webové služby vs. Corba, DCOM, RMI**
- zcela nezávislé na použité platformě
- dostupné globálně, umožňují snadnou integraci různých prostředí
- postaveny na jednoduchých technologiích jako XML a HTTP
- méně efektivní než „nativní“ protokoly
- tři základní pilíře
- SOAP
- WSDL
- UDDI

Komentář [JK4]: No, tak tomu já říkám vysvětlení.

Komentář [JK5]: A víte, co je to „nativní protokol“?

5. Vysvětlete pojem SOAP

- SOAP = Simple Object Access Protocol
- vzdálené volání služby (funkce) je definováno jako jednoduchá XML zpráva, která se obvykle přenáší pomocí HTTP protokolu
- SOAP definuje základní strukturu XML obálky a způsob mapování datových typů do XML
- existuje několik knihoven pro snadnou implementaci klienta i serveru
- <http://www.soapware.org> – seznam dostupných implementací

6. Popište XHTML

= Extensible Hypertext Markup Language

- HTML 4.01 se striktnější syntaxí XML
- XHTML dokumenty lze psát tak, aby jim rozuměly i prohlížeče bez podpory XML

- první stupeň očištění webu od syntakticky nekorektních stránek (většina kódu dnešních prohlížečů se jen stará o odstranění chyb ve špatném HTML)

- vývoj HTML skončil, v budoucnu se dál bude rozvíjet jen XHTML
- modularizace
- rozložení celého standardu XHTML 1.0 na několik základních skupin elementů
- různá výstupní zařízení podporují různé skupiny elementů
- podpora mobilních a bezdrátových zařízení
- rozdíly ve schopnostech jednotlivých koncových zařízení (PC versus mobilní telefon)
- profily – podмноžiny XHTML charakterizující jednotlivá zařízení
- WAP/WML – jednoduchý protokol a značkový jazyk pro mobilní zařízení (v budoucnu se místo WML bude používat modul XHTML Basic)
- možnost vkládání „cizích“ XML dat do XHTML – např. obrázky v SVG, matematické vzorce v MathML, ...

Praxe XHTML

- nepoužívanější prohlížeč IE nepodporuje XHTML, chová se k němu jako k trochu podivnému HTML
- W3C nebylo schopno od roku 2001, kdy uvolnilo XHTML 1.1, přijít s novou verzí XHTML, která by umožnila vytváření komponovaných dokumentů
- XHTML mělo umožnit do dokumentu vkládat další značky pro zlepšení sémantiky, ale současné specifikace to neumožňují
- návrh XHTML 2.0 je z mnoha technických důvodů špatný a výrobci prohlížečů o něj nejeví zájem

Komentář [JK6]: Když tohle napíšete do testu, asi byste měli nějaké ty technické důvody znát. Znáte nějaké?

7. Co je to DTD?

DTD = Definice typu dokumentu

V jazyce SGML můžeme vytvářet definice typu dokumentu (DTD). DTD obsahuje seznam elementů, které můžeme v dokumentu použít. U každého elementu pak definujeme, jaké další elementy může obsahovat, a jaké může mít atributy. Jazyk HTML má samozřejmě také své DTD. Tím je dáno, které elementy a kdy můžeme na webové stránce použít.

Pokud máme k dispozici DTD můžeme pomocí speciálního programu – tzv. parseru – kontrolovat, zda dokument tomuto DTD vyhovuje.

To jakému DTD SGML dokument (tedy i HTML stránka) odpovídá, se udává pomocí deklarace typu dokumentu

Pro odkaz na DTD se nepoužívá přímo jméno souboru, ale tzv. veřejný identifikátor. Ten identifikuje některé z běžně používaných DTD.

DTD pro HTML 4.01

Standard HTML 4.01 definuje tři DTD:

Transitional DTD obsahuje všechny elementy a atributy HTML kromě těch, které se používají pro rozdělení okna prohlížeče na rámy. Veřejný identifikátor je: `-//W3C//DTD HTML 4.01 Transitional//EN`.

Strict DTD je již striktnější než Transitional. Neobsahuje žádné elementy a atributy, které slouží pro definici vizuálního vzhledu dokumentu (např. elementy font, center a atributy jako align a color).

Veřejný identifikátor je `-//W3C//DTD HTML 4.01 Strict//EN`.

Frameset DTD se používá na stránkách, které rozdělují okno prohlížeče do několika rámců pomocí elementu frameset.

Veřejný identifikátor je:

`-//W3C//DTD HTML 4.01`

Frameset//EN.

DOCTYPE by na stránkách nemělo chybět. Identifikuje použitou verzi HTML a umožňuje automaticky zkontrolovat syntaxi stránky pomocí parseru.

8. K čemu slouží kaskádové styly?

- CSS umožňují oddělit vzhled a obsah stránky
- vzhled jednotlivých elementů je úsporně definován odděleně od HTML kódu
- jeden styl může být sdílen více stránkami
- jednotný vzhled
- rychlé změny designu
- V našem případě je celý styl tvořen pouze jedním pravidlem. Každé pravidlo má dvě části – selektor (v našem případě h1) a deklaraci (color: blue).
- Selektor určuje elementy, na které bude deklarace aplikována.
- Každá deklarace se skládá ze dvou částí – z vlastnosti a její hodnoty. Deklarace můžeme sdružovat dohromady, pokud je oddělíme pomocí středníku

9. Jaké 3 typy dokumentu definuje standard HTML 4.01?

DTD pro HTML 4.01

Standard HTML 4.01 definuje tři DTD:

Transitional DTD obsahuje všechny elementy a atributy HTML kromě těch, které se používají pro rozdělení okna prohlížeče na rámy. Veřejný identifikátor je: -//W3C//DTD HTML 4.01 Transitional//EN.

Strict DTD je již striktnější než *Transitional*. Neobsahuje žádné elementy a atributy, které slouží pro definici vizuálního vzhledu dokumentu (např. elementy font, center a atributy jako align a color).

Veřejný identifikátor je -//W3C//DTD HTML 4.01 Strict//EN.

Frameset DTD se používá na stránkách, které rozdělují okno prohlížeče do několika rámců pomocí elementu frameset.

Veřejný identifikátor je:

-//W3C//DTD HTML 4.01

10. Jaká je nejnovější verze HTML?

XHTML, viz otázka 6

11. Rozdíl mezi HTML 4.0 a XHTML

viz otázka 6

12. Proč se na stránkách používají metainformace

- snazší nalezení stránky
- metadata = data o datech ;)
- v našem případě obsahují přídavné (sekundární) informace o dokumentu
- usnadňují prohledávání, katalogizaci, navigaci – zkrátka automatické zpracování
- různé slovníky metadat
- standard HTML
- časem zažité

Komentář [JK7]: Tak to asi ne přátelé. XHTML není verzí jazyka HTML. Nejnovější verzí jazyka HTML je 4.01, u XHTML to je 1.1.

Běžně používaná metadata:

- autor dokumentu

```
<meta name="author" content="Jan Novák">
```

- klíčová slova

```
<meta name="keywords" content="www, design, metadata, HTML">
```

- stručný popis stránky

```
<meta name="description" content="Stránka o designu stránek. . Základní pravidla návrhu stránky.">
```

- zejména klíčová slova a popis stránky je vhodné uvádět. Používá je mnoho vyhledávacích služeb – přesnější vyhledávání, lepší zobrazení výsledku dotazu

13. Jaký by měl být dobře navržený web?

- design by měl být střízlivý – méně je někdy více
- bez zbytečně velkých animací a obrázků, které jsou k ničemu
- vhodné kombinace barev
- barva pozadí a textu musí být zvolena tak, aby se text dobře četl
- obrázky na pozadí je lepší vůbec nepoužívat (téměř žádná profesionální stránka je nepoužívá)
- jako barva pozadí je vhodná zejména bílá a velice světlé odstíny dalších barev na obrazovce jsou čitelnější bezpatková písma (Arial, Verdana, Helvetica)
- nekombinovat příliš druhů písma na jedné stránce
- řádka by neměla být delší než 40-60 znaků
- pro zvýraznění používat barvy, tučné písmo nebo kurzívu – zásadně nepoužívat podtrhnutí
- dodržovat typografická pravidla

14. Co by na žádné stránce nemělo chybět?

- každá stránka by měla obsahovat několik identifikačních údajů: kdo ji vytvořil včetně kontaktu, datum poslední modifikace, do kdy jsou uvedené informace platné
- z každého dokumentu by mělo být jasné, na kterém jsme serveru (někdo se ke stránce může dostat zvenčí pomocí odkazu)
- každý dokument by měl obsahovat výstižný název (element title)

15. Jaký je rozdíl mezi bitmapovou a vektorovou grafikou?

16. Jaký je zásadní rozdíl mezi formáty GIF, JPEG a PNG?

17. Jaké formáty obrázků lze vložit do HTML stránky?

18. Jaké formáty grafiky se na webu používají?

Vektorová - pro firemní loga, schémata apod. je úspornější

SVG

- vektorový formát využívající XML syntaxi
- obrázek se skládá ze základních grafických elementů jako úsečka, kružnice, text, ...
- může být uložen v samostatném souboru nebo přímo jako součást stránky v XHTML
- možnost podle potřeby zvětšovat/zmenšovat
- s obrázkem lze manipulovat pomocí rozhraní DOM a JavaScriptu
- podpora v prohlížečích

Komentář [JK8]: Dobrá nebo špatná? To jsem z toho jelen.

- vytváření obrázků v SVG – většinu vektorových editorů lze rozšířit o modul s exportem do SVG

Bitmapová

GIF

- nejpoužívanější formát
- umožňuje vytvářet obrázky maximálně ve 256 barvách
- není příliš vhodný pro zařazování naskenovaných fotografií na stránky
- animované obrázky
- jedna transparentní barva
- prokládané obrázky (rychlejší náhled na pomalé lince)
- špatná licence

JPEG

- pouze barevný režim TrueColor
- ztrátová komprimace dat (vhodná především pro fotografie)
- progresivní verze formátu (rychlejší zobrazení náhledu)

PNG

- náhrada formátu GIF (není omezen patentem)
- umožňuje vytvářet obrázky v mnoha barevných hloubkách včetně TrueColor (dokonce i 48bitový TrueColor), v odstínech šedi apod.
- alfa-kanál – pro každý bod obrázku lze určit jeho průhlednost
- neumožňuje tvorbu animovaných obrázků
- prokládání obrázků je oproti GIFu ještě vylepšeno
- podporují jen novější verze prohlížečů

19. proč je dobré použít atributy width a height při vkládání obrázků?

atributy umožňují nastavení šířky a výšky obrázku, takže zobrazený obrázek bude velký přesně podle těchto hodnot; dobré je to k tomu, aby se nezobrazil příliš velký, nebo naopak příliš malý, nerozhodil tak celé stránky

Komentář [JK9]: No tak to je s odpuštěným blbost. Tyhle atributy se používají k tomu, aby prohlížeč věděl kolik místa má pro obrázky vyhradit ještě předtím, než mu server pošle samotné obrázky, a nebylo tak nutné stránku několikrát překreslovat.

20. Co jsou synchronizovaná média a jaký je rozdíl mezi ostatními systémy?

- prezentace složená z nezávislých obrázků, zvuků, videosekvencí, textů apod.
- průběh prezentace se řídí buď podle času, nebo podle interakce s uživatelem
- u jednotlivých objektů můžeme určit jejich umístění na obrazovce a čas spuštění
- má menší nároky na přenosovou kapacitu než samostatné video
- lze snadno personalizovat – např. doplnění jména uživatele do prezentace
- formáty: SMIL a Flash

SMIL

- standard konsorcia W3C, založen na XML, na vytváření není potřeba žádný speciální software
- výhody: nezávislé na platformě, snadná integrace mnoha rozličných formátů
- nevýhody: teprve vznikající podpora v aplikacích

FLASH

- proprietární formát firmy Macromedia
- pro vytváření je nutné použít program Flash nebo obdobný SW
- IE a NN standardně obsahují přehrávače pro starší verze Flashe, zdarma lze získat přehrávač nejnovější verze

- nevýhody: vytváří se obvykle v komerčních aplikacích, které nejsou nejlevnější
 - výhody: Flash je program se snadnou obsluhou, oblíbený u designerů stránek, Flash přehrávač je standardní součástí prohlížečů
- fenomén dneška – celé stránky ve Flashi

Komentář [JK10]: A to je dobře nebo špatně?

21. Vymenujte co nejvíce programovacích jazyků a vývojových prostředí na straně serveru

HTML, XHTML, CSS, JavaScript + DOM, VBScript, Perl, Tcl, Python, PHP, ASP, JSP, DHTML, XML, AJAX, SSI, CGI, SAPI

Komentář [JK11]: No v té otázce určitě není „co nejvíce“. Stačí mi tak 3-4.

Komentář [JK12]: Na straně serveru? Tak to tyhle asi nebudou.

Komentář [JK13]: DHTML na straně serveru, a jéje, to bude nula bodů za otázku.

Komentář [JK14]: Od kdy je XML programovací jazyk?

Komentář [JK15]: To není ani programovací jazyk ani vývojové prostředí.

Komentář [JK16]: Vite, co to je?

22. Popište DHTML

- marketingový pojem, který přinesly 4.x verze NN a IE
- prohlížeče měly oproti předchozím verzím mnohem bohatší objektový model, umožňovaly tvorbu velice interaktivních efektů
- dynamické HTML obvykle označuje skloubení HTML, JavaScriptu a kaskádových stylů
- další vývoj – behaviour
- kód obsluhy událostí je zcela oddělen od HTML stránky
- vazba mezi událostmi a elementy na stránce je definována pomocí CSS

23. K čemu slouží rozhraní CGI?

24. Rozdíl mezi metodami GET a POST

= Common Gateway Interface

- rozhraní definuje způsob komunikace web-serveru s aplikací
- CGI skript je program, který používá rozhraní CGI
- CGI skripty lze psát v téměř libovolném jazyce, stačí dodržet konvence rozhraní CGI
- shell, Perl, C/C++, Pascal, Python, ...
- podpora CGI nebývá implicitní, musí se ve web-serveru zapnout (bezpečnost)

Komentář [JK17]: Tyto texty neodpovídají na otázku, co to je rozhraní CGI.

Předávání parametrů přes rozhraní CGI

- existují dvě metody – GET a POST
 - způsob je určen přímo v HTML formuláři
- ```
<form ... method="post">
```
- ```
<form ... method="get">
```
- standardní je metoda GET
 - před odesláním prohlížeč všechna data z formuláře zakóduje do jednoho dlouhého řetězce
 - hodnoty polí jsou upraveny tak, aby je šlo zapsat jako součást URL
 - mezera → + speciální znaky, znaky s diakritikou apod. → %xx, kde xx je kód znaku v šestnáctkové soustavě
 - při metodě GET jsou zakódovaná data přidána za URL požadavku (za znak ?), rozhraní CGI předá skriptu data v proměnné prostředí *QUERY_STRING*
 - při metodě POST jsou data předávána v těle HTTP požadavku, CGI skript je dostane na svůj standardní vstup
 - kromě samotných dat z formuláře, předá web-server i další užitečné údaje pomocí proměnných prostředí

25. K čemu se používají skriptovací jazyky na straně serveru?

Základní principy generování stránek na serveru

- na serveru je dynamicky generováno HTML na základě požadavku uživatele

- do prohlížeče je odesílán již jen čistý HTML kód
- není potřeba žádný speciální prohlížeč, lze použít libovolný se základní podporou HTML
- v případě potřeby lze na serverem generovaných stránkách použít i klientské technologie (např. JavaScript)

Interakce s webovou aplikací na serveru

- lze použít vše, co vyvolá HTTP požadavek na webový server
- HTML formuláře
- odkazy
- automatické otevření nové stránky pomocí JavaScriptu
- obsah stránky se nemusí řídit jen požadavky uživatele, ale může záviset i na externích vstupech (čas) – např. graf vývoje burzovního indexu

26. Co je PHP?

Hypertextový preprocesor PHP

- přímo do HTML kódu se zapisují jednoduché příkazy
- jednoduchá syntaxe založená na C, Perlu a Javě
- speciálně navržený jazyk pro tvorbu webových aplikací
- velmi rozsáhlá knihovna funkcí
- nezávislost na platformě – může spolupracovat s v podstatě libovolným serverem na libovolné platformě
- **OSS** – dostupný zdarma včetně zdrojových kódů

Komentář [JK18]: Víte co to je? Nemá tam být místo „O“ třeba „A“?

27. Vysvětlí pojem Java Server Pages

- do HTML kódu se zapisují příkazy Javy
- k dispozici jsou podobně jako v ASP speciální objekty pro čtení dat z formulářů apod.
- pro lepší oddělení designu a logiky lze definovat „tag libraries“ – uživatelsky definované tagy, které volají předem připravené komponenty
- o spuštění JSP se stará servlet, který JSP automaticky převede do Javy, zkompiluje do bytecode a spustí

28. Vysvětlete, co je ASP.NET

- nová platforma Microsoftu (nebo také „opajcovaná Java“)
- aplikace se zdrojových kódů překládá do CIL (Common Intermediate Language) – obdoba javového bytecode
- o spuštění CIL se stará CLR (Common Language Runtime)
- před spuštěním je vždy CIL převeden do nativního kódu (obdoba JIT kompilace v Javě)
- CLR zatím existuje pouze pro Windows
- existuje několik projektů, jejichž cílem je vytvoření CLR pro další platformy (např. Mono pro Linux)
- všechny jazyky, které lze kompilovat do CIL (VB.NET, Managed C++, C#, ...) používají stejné knihovny (velká změna oproti předchozím verzím jazyků)
- výborná podpora XML
- hlavní tři knihovny – webové služby, Web Forms (tvorba webových aplikací), Windows
- s klasickými ASP nemá skoro nic společného
- vyvíjí se jako klasická klientská aplikace – prvky uživatelského rozhraní a obsluha událostí
- ASP.NET si webový server přeloží do nativního kódu, který se stará o postupné zaslání HTML kódu a obsluhu formulářových dat
- vygenerovaný kód detekuje použitý prohlížeč a tomu přizpůsobí generovaný HTML a JavaScriptový kód
- VisualStudio.NET umožňuje aplikace vyvinout pouhým „naklikáním“

Komentář [JK19]: Otázka je, „co to je ASP.NET“, ne „co to je .NET“.

29. Jaké jsou největší nedostatky současných technologií používaných pro vývoj webových aplikací?

Rychlost vývoje aplikací

- kompilované jazyky – pomalá
- po provedení každé změny je potřeba program rekompilovat (pracné a pomalé)
- interpretované
- rychlé změny – stačí opravit zdrojový kód a dát v prohlížeči reload
- rychlý běh aplikací a rychlý vývoj zároveň □ JSP, ASP.NET, ...
- programátor pracuje pouze se zdrojovým kódem skriptu
- o kompilaci se automaticky stará webový server nebo jeho modul

Komentář [JK20]: A jak to souvisí s otázkou? Takhle by to za moc bodů nebylo.

30. Jaké jsou základní stavební prvky HTML?

Elementy a tagy

Jednotlivé části HTML stránky se označují pomocí elementů. Každý element se skládá z počátečního tagu, obsahu elementu a ukončovacího tagu. Malý příklad:

```
<p>Pokusný odstavec a zároveň element.</p>
```

Ukázka obsahuje jeden element, počáteční tag <p> a ukončovací tag </p>.

Některé elementy (například
 a <hr>) nemusí mít ukončovací tag

Atributy

Elementy se obvykle používají pro vyznačení struktury dokumentu. Pokud potřebujeme nějak podrobněji vymežit význam elementu, použijeme

k tomu atribut. Atribut se zapisuje za počáteční tag. Odkazy vás rázem přenesou úplně

```
<a href="nekam.html">jinam</a>
```

Atribut má vždy nějakou hodnotu, která se mu přiřazuje. Pokud hodnota obsahuje pouze

písmena, číslice, pomlčku a tečku, nemusí se hodnota atributu uzavírat do uvozovek. V

opačném případě musíme hodnotu atributu uzavřít do uvozovek nebo do apostrofů. U jednoho

tagu můžeme samozřejmě použít více atributů. Atributy se mohou používat pouze u

počátečních tagů, nikoliv u ukončovacích.

Znakové entity

V HTML mají znaky '<' a '>' speciální význam. Pokud je potřebujeme zapsat do stránky,

musíme použít tzv. znakovou entitu. Znaková entita má tvar &název_entity

Komentář [JK21]: A na konci je ještě středník

31. K čemu se využívají hlavičky v HTML odesílaných ze strany uživatele (z prohlížeče)?

32. . K čemu se využívají hlavičky v HTML zasílané klientovi jako součást odpovědi?

Komentář [JK22]: Neměl by to být spíš protokol HTTP?

Komentář [JK23]: Neměl by to být spíš protokol HTTP?

hlavičky obecně:

- některé hlavičky lze použít v požadavku i v odpovědi
- některé jsou specifické pro požadavek, resp. odpověď
- ne všechny hlavičky jsou povinné, většina je volitelná

používají se k identifikaci údajů

- např. Date datum a čas požadavku/odpovědi, Content-Type druh zasílaných dat

Host doménová adresa serveru – umožňuje správnou funkci více virtuálních serverů na jedné společné adrese

Location přesměrování na jinou stránku

- Domlouvání obsahu

Accept seznam typů dat podporovaných klientem

Accept-Charset seznam kódování, které podporuje klient

Accept-Language seznam podporovaných jazyků

Allow seznam metod, kterými je dostupný určitý objekt

Praktické využití HTTP hlaviček

Zákaz kešování stránek:

- informace na stránce se mění v čase
- burzovní indexy
- on-line přístup do IS
- reklamní bannery

33. Jak lze nahradit stavovost např. v e-shopech?

- přenášení údajů v URL a skrytých polí formuláře
- cookies
- session proměnné

a) Předávání stavových proměnných v URL a skrytých polích formulářů

- nebezpečné – všechny stavové informace jsou v každém požadavku/odpovědi
- zbytečně zvyšuje přenosovou kapacitu
- velmi pracné na implementaci – za každý odkaz a do každého formuláře se musí přidat všechny stavové proměnné

b) Cookies

- krátká informace, kterou si server uloží v prohlížeči
- při následujících přístupech k témuž serveru je cookie zaslána zpět, cookie je vázána na server a případně i na adresář – informace se nedostanou k tomu, komu nepatří
- časová platnost cookie
- session cookie – platí do té doby, než se vypne prohlížeč, nastavena na konkrétní délku

Předávání stavových informací pomocí cookies

- nebezpečné – všechny stavové informace jsou v každém požadavku/odpovědi
- implementace je velice snadná
- některé starší prohlížeče cookies nepodporují, novější umožňují cookies vypnout
- na cookies bohužel nelze spoléhat

c) Session proměnné

- každému novému uživateli se přiřadí unikátní identifikátor (tzv. session-id)
- předává se s každým požadavkem pomocí cookie nebo parametrů v URL, resp. skrytých polí ve formuláři

• session-id je konstruováno tak, aby bylo těžko odhadnutelné (většinou náhodné číslo + hashovací funkce MD5 nebo SHA)

- pro každé session-id má webový server vyhrazen prostor pro ukládání dat (proměnných)

Předávání stavových informací pomocí session proměnných

- poměrně bezpečné – s každým požadavkem se přenáší jen malá část dat a session-id
- šetří kapacitu sítě – data jsou ukládána přímo na web-serveru
- velice snadná implementace – většina prostředí pracuje se session proměnnými téměř stejně jako s běžnými proměnnými

Komentář [JK24]: Vůbec neodpovídáte na otázku, k čemu se ty hlavičky dají využít. To povídání okolo je zbytečné.

Komentář [JK25]: Stavovost? Spíše asi bezstavovost, ne?

Komentář [JK26]: Kryptografické dnes doporučují alespoň SHA-256 nebo SHA-512. MD5 a SHA-1 radši nepoužívejte.

34. Jaký je rozdíl mezi skriptováním na straně klienta a serveru?

klient – výpočet a překlad se provádí na straně klienta – např. java script...data se stáhnou ke klientovi, kde dojde k překladu
server - výpočet a překlad se provádí na straně serveru – např. PHP

35. Zásady správné tvorby hesel

- min 6 znaků
- žádný slovníkové slova + aby to nedávalo smysl
- čísla, písmena, ostatní znaky
- nikdy nikam nepsat, nikomu neříkat, pravidelně měnit

Komentář [JK27]: Tak takováhle otázka v testu není, ale mohl bych tam něco v tomto smyslu přidat, že?

36. Jaké funkce mělo nabízet prostředí pro vývoj webových aplikací?

mělo používat rozšiřující technologie – klientské skripty, dynamické html, Java-applety, ActiveX komponenty; díky těmto technologiím stránky mohou být více interaktivní a snazší na ovládání, mít rychlejší odezvu (nemusí se komunikovat se serverem), odlehčit zátěže na serveru

zvýraznění syntaxe kódu, možnost editace stránek rovnou na serveru, aby se mohl rovnou ten kód odladit (kontrola - něco jako kompilátor), náhled kódu do prohlížeče (tukneš a zobrazí ti to jak bude vypadat stránka), integrace s ostatními - sql na databáze, xml, ...

Komentář [JK28]: Každý na to může mít vlastní názor. Ale trend je dnes jasný – vývojové prostředí by vás mělo co nejvíce odstínit od takových trapných detailů jako jsou nějaké ActiveX a Java-applety.

Komentář [JK29]: Ne že by v PHP nešlo použít JDBC, určitě jej lze volat přes Java bridge, ale asi to nebude ten nejlepší postup. V PHP se nejčastěji používají abstraktní databázové třídy jako DB z PEAR, nebo PDO, které je standardní součástí PHP od verze 5.1. Jinak v testu tahle otázka není vázaná na PHP, ale obecně na jakýkoliv jazyk pro psaní webových aplikací.

37. Je možné programovat v php bez použití konkrétní databáze? Jak?

Ano - JDBC, ODBC

38. Proč není dobré ke zmenšování obrázků na stránkách používat atributy width a height?

zbytečně zabírají místo na serveru, lepší je obrázek ořezat v nějakém editoru, a tím zkrouhnout jeho velikost

při tomto způsobu se může stát, že obrázek bude takový vroubkovaný

Komentář [JK30]: A to snad někoho při současné ceně disků trápí?

Komentář [JK31]: Vroubkovaný nevím, to jsou snad jen kondomy.

39. Jaké znakové sady používá html a xhtml?

skládá se z:

sady instrukcí a pozice kódu

např. ASCII znaková sada, UCS – definováno standardy ISO10646

- UNICODE

Komentář [JK32]: Skutečný problém je v tom, že do prohlížeče se přenáší pro obrázek více dat, než je nutné pro zobrazení a plýtvá se tak přenosovou kapacitou a stránka se pomaleji vykreslí.

Komentář [JK33]: Tak tuhle odpověď jsem ani já nepochopil. Pravda je taková, že XHTML i HTML (od verze 4) používá pouze jedinou znakovou sadu ISO 10646 (alias Unicode).

40. 3 základní pilíře WWW

- SOAP, WSDL, UDDI

WSDL = Web Services Description Language

- přesný popis rozhraní webové služby dostupné přes SOAP
- z WSDL definice lze automaticky generovat klientský kód, který umí službu využívat, i skeleton serverového kódu

UDDI = Universal Description, Discovery and Integration

- univerzální adresář obsahující seznam a popis dostupných webových služeb
- umožňuje automatické nalezení požadované služby

Komentář [JK34]: To jsou tak možná pilíře webových služeb. Pilíře webu jsou HTML, HTTP a URI.

Komentář [JK35]: Závěrem vám přeji mnoho úspěchů v testu, doufám, že ho napíšete lépe, než kdybyste jen odevdali odpovědi z tohoto dokumentu.

Váš
..... Jirka Kosek