# Jak vytvořit jednoduché webové stránky.

Od „vynálezu“ protokolu HTTP pro přenos dokumentů a jazyka HTML pro jejich popis již uběhlo více než 17 let. Za tu dobu se vyvíjel jak protokol, tak i jazyk. Od původního HTML verze 1 přes HTML verze 4 se přešlo až k XHTML verze 1.1 a CSS, do češtiny překládaným jako kaskádní styly.

Výraznou změnou XHTML vůči HTML je to, že se snaží oddělit od sebe obsahovou stránku textu od prezentační stránky. Důvod je prostý. Dnes existuje mnoho typů zobrazovacích jednotek, vyžadujících prezentaci stránek opravdu odlišným způsobem. Nejčastěji je to zobrazení textu na obrazovce a tisk téhož. Nebo převod textu do podoby vhodné pro zobrazení na mobilním telefonu, speciálních výstupních zařízeních (převod na zvukovou informaci) atd. Nepřehlédnutelným důvodem je možnost jednoduše změnit vzhled stránek po jejich vytvoření.

## Značky

K popisu úpravy textu slouží jazyk XHTML. Ten udává která část textu je nadpis, odstavec, tabulka, kam se má zobrazit obrázek, kterou část textu chceme zvýraznit, definice odkazů na jiné stránky apod. K určení jak bude část textu opravdu vypadat slouží popis realizovaný pomocí jiného jazyka, CSS (kaskádních stylů). Zde je možné určit vlastnosti prostoru kam text umisťujeme (počet a rozmístění sloupců, barva pozadí, obrázek pozadí, …), textu (řádkování, mezery mezi znaky a slovy), vlastnosti písma (barvu, velikost, typ písma, font, … ), vlastnosti odstavců (rámečky, odsazení, … ), vzhled tabulek atd.

K označení jednotlivých prvků textu se v jazyku XHTML používá značek. Značka klíčové slovo, případně klíčové slovo s parametry, uzavřené do špičatých závorek „< >“. Zapisuje se malými písmeny. Může obsahovat i číslice.

Např. <body> se používá pro označení začátku textové části dokumentu.

Značky mohou být párové, nebo nepárové. Párové vytyčují začátek a konec nějaké oblasti. Počáteční značka vystupuje ve významu otevírací závorky, koncová pak ve významu uzavírací závorky. Koncová značka se od počáteční liší přidáním znaku „/“ před klíčové slovo.

Např. <title>Dokonalé webové stránky</title> může být titulek Vaší webové stránky, čili text, zobrazující se v záhlaví prohlížeče.

Nepárové značky se používají pro uvození nějakého příkazu, který potřebujete vložit do textu. Je to např. příkaz pro vložení obrázku, vložení parametru do záhlaví dokumentu, vložení nové řádky do odstavce. Např.

<img src=”logo.gif“ />

<meta name="generator" content=" HTML Validator (http://www.htmlvalidator.com/)" />

<br />

Původně neobsahovaly nepárové značky koncové lomítko. Doporučení XHTML zavádí i pro nepárové značky jakési zakončení, reprezentované trojicí znaků „mezera“ „/“ a „>“ (viz. <br />).

## První stránka

„Prázdná“ XHTML stránka může vypadat např. takto:

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<title>Zde bude výstižný název stránky </title>

<meta name="generator" content="CSE HTML Validator Professional (http://www.htmlvalidator.com/)" />

<meta http-equiv="Content-Language" content="cs">

<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">

</head>

<body>

<p>

<img src="a.gif" alt="" />

</p>

</body>

</html>

Pro vysvětlení obsahu dokumentu jej uměle rozdělíme na tři části.

První část začíná řetězcem <?xml version="1.0" encoding="UTF-8"?>. Jsou zde uvedeny informace podle kterého standardu je stránka vytvořena. Tuto část je výhodné opsat nebo si ji nechat vygenerovat některým z WYSIWYG editorů (WYSIWYG je akronym anglické věty „What you see is what you get“, česky „co vidíš, to dostaneš“).

Druhá část začíná značkou <head> a končí značkou </head>. Z informací uvedenými mezi těmito závorkami, je nejdůležitější titulek stránky (<title> … </title>). Dále jsou zde tzv. “meta” značky, které slouží prohlížeči dokumentu k rozpoznání jazyka a použitého kódování. Pokud meta značky chybí, nemusí prohlížeč zobrazovat znaky s diakritickými znaménky správně.

Třetí část je tělo dokumentu. Je ohraničeno značkami <body> a </body>. Zde je uveden zobrazovaný obsah stránky.

Pro napsání jednoduchého textu vystačíme s výše uvedenu předlohou, do které mezi značky <title> … </title> vložíme název této stránky a mezi značky <body> … </body> odpovídající text. Soubor nazveme index.htm nebo index.html a můžeme si jej zobrazit v prohlížeči (např. tak, že pravým tlačítkem myši vybereme soubor a v otevřeném menu vybereme některý z HTML prohlížečů).

## Styly a jejich umístění

Pokud chceme změnit standardní zobrazení, musíme vytvořit odpovídající styly a spojit je se značkami textu. To je základní myšlenka. Nejjednodušší způsob je přidat ke značce parametr, popisující její vlastnost. Např. doplnění značky pro odstavec (paragraf) o vlastnost určující barvu písma způsobí, že text odstavce bude modrý a pozadí stříbrné.

<p style=“color:black;background-color:silver;“> … </p>

Z příkladu je vidět, že jsme nadefinovali dvě vlastnosti: barvu písma a barvu pozadí. Každá vlastnost má název (color, background-color) a hodnotu (black, silver). Název a hodnota jsou odděleny znakem ‘:’ a vlastnosti jsou zakončeny znakem ‘;’. Každá vlastnost má pochopitelně přiřazenu množinu hodnot, definovaných ve specifikaci CSS.

Pokud bychom do sebe styly vnořovali, platí vlastnosti toho posledně vnořeného.

Tento způsob je však vhodný pouze pro pokusy, protože pokud byste chtěli styl změnit, museli byste tak učinit všude.

Pokud bychom si vystačili s jednoduchým spojením značek a stylu jejich zobrazení, kdy je daná značka v celém dokumentu zobrazována podle jednoho stylu, můžeme udělat následující. Do záhlaví dokumentu přidáme blok definicí stylů, např.:

<head> …

<style type=“text/css“>

p { color:black; background-color:silver; }

h1, h2, h3 { color:red; background-color:white; }

</style>

</head>

kde p je označován jako selektor za kterým následuje deklarace stylu uzavřená do složených závorek. Můžeme též zkrátit zápis vyjmenováním více selektorů se stejnými vlastnostmi. Jména selektorů oddělíme čárkami. Nyní budou v textu dokumentu všechny paragrafy psány černě na stříbrném pozadí a hlavičky h1 až h3 červeně na bílém pozadí.

Nevýhodou je, že takto nadefinované styly jsou vztaženy ke stránce, do které jsme je zapsali. Chcete-li je použít i v jiném souboru, musely by se překopírovat. Jednodušší je umístit styly do samostatného souboru a do stránky je nakopírovat automaticky. Do záhlaví můžeme napsat konstrukci

<head>

<style type=“text/css“> @import url(styl.css) </style>

</head>

kde styl.css je soubor, obsahující definici stylů. Ty se vloží do dokumentu a my je můžeme obdobně vkládat i do ostatních stránek.

Jinou možností je sdělit prohlížeči jméno souboru se styly. Opět do záhlaví zapíšeme následující příkazy:

<head>

<link rel=“stylesheet type=“text/css“ href=“styl.css“ >

</head>

V tomto případě si prohlížeč zadaný soubor stáhne samostatně (jednou) a využije jej ve všech stránkách, kde je na něj odkaz.

## Třídy a identifikátory

V předchozích případech jsme viděli, že styl zobrazení jednotlivých prvků dokumentu můžeme zapisovat přímo se značkami nebo do odděleného místa. Vysvětlili jsme si, že je lepší oddělit text od jeho prezentace. V příkladu uvedeném výše a zkopírovaném níže vidíme, že lze nadefinovat vlastnosti jednotlivých značek odděleně od jejich výskytu.

p { color:black; background-color:silver; }

h1, h2, h3 { color:red; background-color:white; }

V tomto případě budou ale všechny typy prvků textu psány jedním stylem. To ale neřeší požadavek, kdy bychom potřebovali stejně označené části textu psát různými styly. Např. různě formátované odstavce. Tento problém řeší zavedení vlastních selektorů ve dvou formách – selektor třída a selektor identifikátor. V obou případech je selektor nějaké vymyšlené jméno, které nesmí začínat číslicí. Jméno by ale mělo být voleno tak, aby souviselo s nadefinovanými vlastnostmi. Selektor třídy je jméno předcházené tečkou, selektor identifikátoru jméno předcházené znakem ‘#’. Např. selektor třídy

.modry { color:blue; }

použijeme kdekoliv v textu a ten se zbarví na modro

<h1 class=”modry”> Nějaký modrý nadpis první úrovně </h1>

<p class=”modry”> První modrý odstavec. </p>

nebo můžeme definovat třídu h2.zelený a h3.zeleny, které jsou omezené na značku h2 a h3

h2.zeleny { color:green; }

h3.zeleny { color:yellow; }

ty pak použijeme v textu následovně

<h2 class=”zeleny”> Druhá úroveň bude zelená. </h2>

<h3 class=”zeleny”> Třetí úroveň bude žlutá. </h3>

Selektor indentifikátoru (id) nadefinujeme následovně:

#orange { color:navy; }

h2#orange {color:orange;}

Jejich použití v textu je stejné jako v případě třídy pouze s tím, že místo class použijeme id.

<h2 id=”orange”> Druhá úroveň bude tmavě modrá </h2>

<h3 id=”orange”> Třetí úroveň bude oranžová </h3>

Třídy a identifikátory lze kombinovat. Takže můžeme psát:

#orange { color:navy; }

.modry { color:blue; }

<p class=”modry” id=”orange” > Třída má přednost před id – bude to modré</p>

## Kontextové selektory a dědičnost

Styly se mohou nořit jeden do druhého. Styl, který je otevřen jako první, je uzavřen jako poslední. Je to totéž jako práce se závorkami v matematice.

Není-li uvedeno jinak, vlastnosti stylů se dědí, tj. vlastnosti nadefinované ve vnějším stylu přechází i na styl vnitřní. Jestliže je třeba nastavit jinou vlastnost, použijeme tzv. kontextové selektory. Jsou to selektory oddělené mezerou. Např.

p {color:green; }

div {color:blue; }

span {color:blue; }

div p {color:red; }

p span {color:green; }

Pak následující text se zobrazí s níže uvedenými barvami.

<div> modry <p> cerveny <span> zeleny </span> cerveny </p> modry </div>

Dokonce lze definovat

div p span{color:black; }

s výpisem

<div> modry <p> cerveny <span> černý </span> cerveny </p> modry </div>

## Vlastnosti písma

Není písmo jako písmo. Pomocí kaskádních stylů můžeme nastavit rodinu písma (font-family), řez písma (font-style), variantu písma (font-variant), tloušťku písma (font-weight) a velikost písma (font-size). Existují standardy, ve kterých jsou uvedeny všechny přípustné hodnoty. zde uvedu jen ty nejběžnější.

font-family: Arial, “Times New Roman“, “Courier New”, “MS Sans Serif”, “Verdana”

font-style: normal, italic, oblique

font-variant: normal, small-caps

font-weight: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900

font-size: xx-small, x-small, small, medium, large, x-large, larger, smaller, délka(em, px, pt), procenta

## Vlastnosti textu

I text lze formátovat. Pomocí vlastnoti textu můžeme nastavit ozdobení textu (text-decoration), zarovnání textu (text-align), vertikální zarovnání (vertical-align), velikost mezery mezi slovy (word-spacing), velikost mezery mezi písmeny (letter-spacing), velikost zarážky na začátku odstavce (text-indent), výšku řádku (line-height) a transformaci textu (text-transform). Opět je zde uveden neúplný výčet hdnot parametrů.

text-decoration: underline, overline, none, line-through

text-align: left, right, center, justify

vertical-align: sub, super, top, middle, bottom

word-spacing: normal, délka(em, px, pt)

letter-spacing: normal, délka(em, px, pt)

text-indent: délka(em, px, pt), procento

line-height:normal, procento, délka(em, px, pt), číslo

text-transform: none, capitalize, uppercase, lowercase

## Barvy

color: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow

background-color:

background-image: none, url(soubor)

background-repeat: no-repeat, repeat, repeat-x, repeat-y

background-attachment: scroll, fixed

background-position: top, bottom, left, right, center

## Seznamy

list-style-type: disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none

list-style-position: inside, outside

list-style-image: none, url()

## Boxy

margin:

* margin-top:
* margin-bottom:
* margin-left:
* margin-right:
* margin: top, bottom, left, right

padding:

* padding-top:
* padding-bottom:
* padding-left:
* padding-right:
* padding top, bottom, left, right

border:

* border-top:
* border-bottom:
* border-left:
* border-right:
* border:
* border-color:
* border-style: none, dotted, dashed, solid, double, groove, ridge, inset, outset
* border-width:

width: délka, procento

height: délka, procento

float: left, right, none

clear: left, right, both, none