

1.1 Interaktivita web stránky

Prehliadač (okrem jeho pôvodnej funkcie prezerania hypertextového dokumentu) sa v súčasnosti stáva prostriedkom prístupu k informáciám v rôznych databázach a súčasne novou platformou pre aplikácie nezávislé na konkrétnom operačnom systéme. Mení sa aj úloha web stránky – z pôvodnej statickej stránky umiestnenej na www serveri a poskytujúcej konštantné informácie na stránku reagujúcu na požiadavky používateľa a poskytujúcu informácie, ktoré sa menia v čase. Zabezpečiť túto novú úlohu je možné tak, že každá požiadavka na stránku vyvolá spustenie programu (tzv. skriptu), ktorý zistí potrebné aktuálne informácie a na ich základe vytvorí novú výslednú podobu stránky v HTML. Ďalšie možnosti riešenia interaktivity stránky na strane klienta predstavuje technológia Java Script (1996 - obdoba Pascalu, vkladá sa do HTML stránky), technológia Java-
 applet (Java - obdoba C, po sieti sa šíri platformovo nezávislý byte-code, interpretovaný JVM (Java Virtual Machine) prehliadača) a technológia dynamické HTML (1997 MS IE4, vylepšený JavaScript, každý objekt web stránky je prístupný a možno meniť jeho obsah, zobrazenie, polohu, štýl - vizuálne efekty). Podľa miesta realizácie možno jednotlivé technológie rozdeliť v zmysle tabuľky, ale je možná aj ich vzájomná kombinácia.

○

KLIENT	SERVER
JavaScript	Server Side JavaScript
Java-aplety	cgi skripty
Dynamické HTML	Hypertextový preprocesor PHP
	Active Server Pages

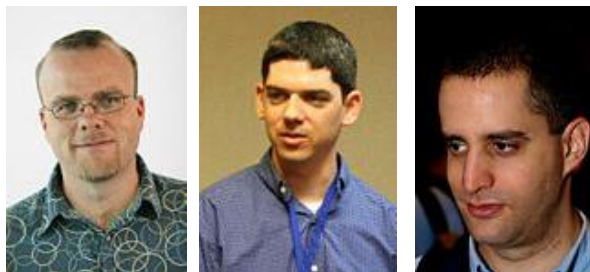
Skripty možno rozdeliť do dvoch skupín:

- serverom vkladané vsuvky - príkazy skriptu sa kombinujú s príkazmi HTML. Pred odoslaním sú tieto príkazy vyhodnotené a spracované www serverom tak, že prehliadač dostane už bežnú HTML stránku. Ich najstarším predstaviteľom je SSI (Server Side Includes), väčšie možnosti majú komerčné systémy LiveWire (Netscape) - dnes pod názvom SSJS (Server Side Java Script) a ASP (Active Server Pages -- Microsoft) (pracuje sa aj na produkte Chilli!ASP, ktorý má byť kompatibilný s ASP a má bežať aj pod Unixom) ako aj voľne šíriteľný systém PHP.
- cgi (Common Gateway Interface) skripty - program zostavený v programovacom jazyku, uložený v spustiteľnej forme (napr. exe súbor). Pri príchode požiadavky obsahujúcej odkaz na tento program sa tento spustí a svoj výstup (vo formáte HTML) odošle prehliadaču ako odpoveď. Na zostavenie cgi sa používajú príkazové shelly (dávkové súbory), hlavným predstaviteľom je jazyk Perl, ale aj bežné jazyky ako C, Pascal alebo Java.

Tvorba skriptov je na rozdiel od HTML o niečo náročnejšia činnosť, ktorá vyžaduje znalosti základov programovania a príslušného prostriedku na tvorbu skriptu. Pre prácu so skriptami musí byť k dispozícii tzv. www server s podporou daného skriptu (nie každý server podporuje všetky druhy skriptov). Medzi najrozšírenejšie patrí web server Apache, teraz aj vo verzii pre Windows.

1.1.1 PHP

Typickým predstaviteľom a jedným z najrozšírenejších je skriptovací jazyk PHP (Professional Home Pages), je voľne šíriteľný a platformovo nezávislý (beží na Unixe - najlepšie na Apache i pod Windows).



**Rasmus Lerdorf (1968), Andi Gutmans (1975),
Zeev Suraski (1968)**

Vytvoril ho Rasmus [Lerdorf](#) v r. 1994 v Perle na evidovanie prístupu k jeho osobným stránkam. Postupne bol doplnený a rozšírený o začleňovanie SQL dotazov do stránok, formulárov a zobrazovanie výsledkov dotazov – Form Interpreter. Celosvetové rozšírenie získala verzia PHP/FI 2.0. Aktuálna verzia je PHP 3.0 od polovice r. 1998, bola zrýchlená a doplnená o ďalšie funkcie a pracuje aj pod Windows. V súčasnosti sa na označenie doporučuje termín hypertextový preprocesor PHP. Vývoj PHP stále pokračuje, v polovici r. 1999 bola uvoľnená verzia nového jadra PHP 4.0 s označením Zend (beta 2), ktorý je podstatne rýchlejší a pracuje s využitím niekoľkých

DLL. V tejto verzii je i ISAPI (Internet Server Application Programming Interface) modul php4isapi.dll rezidentný v pamäti a funkčne analogický s modulom pre Apache. Má vysoký výkon a znižuje zaťaženie systémových zdrojov celého servera.

PHP originally stood for Personal Home Page.^[8] It began in 1994 as a set of [Common Gateway Interface binaries](#) written in the [C programming language](#) by the [Danish/Greenlandic](#) programmer [Rasmus Lerdorf](#). Lerdorf initially created these Personal Home Page Tools to replace a small set of [Perl](#) scripts he had been using to maintain his [personal homepage](#). The tools were used to perform tasks such as displaying his résumé and recording how much [traffic](#) his page was receiving.^[3] He combined these binaries with his Form Interpreter to create PHP/FI, which had more functionality. PHP/FI included a larger implementation for the C programming language and could communicate with [databases](#), enabling the building of simple, dynamic [web applications](#). Lerdorf released PHP publicly on June 8, 1995 to accelerate [bug](#) location and improve the code.^[9] This release was named PHP version 2 and already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax was similar to Perl but was more limited, simpler, and less consistent.^[3]

[Zeev Suraski](#) and [Andi Gutmans](#), two [Israeli](#) developers at the [Technion IIT](#), rewrote the [parser](#) in 1997 and formed the base of PHP 3, changing the language's name to the [recursive initialism](#) *PHP: Hypertext Preprocessor*.^[3] The development team officially released PHP/FI 2 in November 1997 after months of [beta](#) testing. Afterwards, public testing of PHP 3 began, and the official launch came in June 1998. Suraski and Gutmans then started a new [rewrite](#) of PHP's core, producing the [Zend Engine](#) in 1999.^[10] They also founded [Zend Technologies](#) in [Ramat Gan](#), Israel.^[3]

On May 22, 2000, PHP 4, powered by the [Zend Engine](#) 1.0, was released.^[3] As of August 2008 this branch is up to version 4.4.9. PHP 4 is no longer under development nor will any security updates be released.^{[8][9]}

On July 13, 2004, PHP 5 was released, powered by the new [Zend Engine II](#).^[3] PHP 5 included new features such as improved support for [object-oriented programming](#), the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements.^[10] In 2008 PHP 5 became the only stable version under development. [Late static binding](#) had been missing from PHP and was added in version 5.3.^{[11][12]}

A new major version has been under development alongside PHP 5 for several years. This version was originally planned to be released as PHP 6 as a result of its significant changes, which included plans for full Unicode support. However, Unicode support took developers much longer to implement than originally thought, and the decision was made in March 2010^[13] to move the project to a branch, with features still under development moved to a trunk.

Changes in the new code include the removal of `register_globals`,^[14] [magic quotes](#), and [safe mode](#).^{[8][15]} The reason for the removals was that `register_globals` had given way to security holes, and the use of magic quotes had an unpredictable nature, and was best avoided. Instead, to escape characters, magic quotes may be replaced with the `addslashes()` function, or more appropriately an escape mechanism specific to the database vendor itself like `mysql_real_escape_string()` for [MySQL](#). Functions that will be removed in future versions and have been deprecated in PHP 5.3 will produce a warning if used.^[16]

Many high-profile open-source projects ceased to support PHP 4 in new code as of February 5, 2008, because of the GoPHP5 initiative,^[17] provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5.^{[18][19]}

Since version 5.4, PHP has native support for Unicode or multibyte strings, allowing strings as well as class-, method-, and function-names to contain non-ASCII characters.

PHP interpreters are available on both [32-bit](#) and [64-bit](#) operating systems, but on [Microsoft Windows](#) the only official distribution is a 32-bit implementation, requiring Windows 32-bit compatibility mode while using [Internet Information Services](#) (IIS) on a 64-bit Windows platform. As of PHP 5.3.0, experimental 64-bit versions are available for MS Windows.^[22]

[Release history](#) – prehľad vývoja jednotlivých verzii PHP.

PHP skript sa začleňuje priamo do HTML stránky, je porovnateľný s ASP, ale je kratší a zrozumiteľnejší. Je možné ho získať z [www.php.net](#) v zdrojovom tvare pre Unix (Linux) aj v exe tvare pre Windows. Skript je možné napr. odladiť vo Windows a potom ho sprístupniť na výkonnom unixovom serveri. Na prípravu skriptu je vhodný akýkoľvek HTML editor – od Poznámkového bloku cez Programmer's File Editor až po HomeSte, prípadne MultiEdit (pod Unix – vi, joe, pico, emacs). Wysiwyg editory (Netscape Composer, Front Page) sú pre tento účel nevhodné. Napísaný skript - uložiť na www server s podporou PHP, prezeranie - akýkoľvek prehliadač, pri vývoji - vhodné vyskúšať najbežnejšie prehliadače.

PHP skript predstavuje obyčajný HTML kód doplnený o príkazy PHP (vychádza z jazykov C, Java a Perl). Na rozlíšenie od bežnej HTML stránky sa tieto skripty ukladajú do súboru s príponou php, php3, phtml podľa konfigurácia servera. Súbor sa ukladajú do toho istého adresára ako súbory HTML, v prípade Windowsu názov adresára závisí od použitého www servera (napr. web_pages pre Xitami). Keď www server dostane požiadavku na súbor s touto príponou, odovzdá ju systému PHP (preprocesoru), ktorý vykoná interpretáciu PHP príkazov, stránku vráti serveru a ten ju odošle klientovi.

Použitie PHP je jednoduché a intuitívne. Príkazy PHP sa od HTML kódu oddeľujú vyhradenými znakmi '<?php' a '?>'. Tým sú všetky PHP skripty kompatibilné s nástrojmi na automatické spracovanie SGML a XML dokumentov. PHP interpretuje iba príkazy umiestnené medzi vyhradenými znakmi. Príkazy sa oddeľujú bodkočiarkou. Názov premennej sa povinne začína znakom \$ (dolar), priradenie – operátor = (ako v C), porovnanie – operátor ==. Veľké a malé písmená sa rozlišujú v identifikátoroch, nie v príkazoch. Komentár // na jeden riadok, /* a */ ako v C. Pozri <http://phpwebdev.net> alebo [PHP in a Nutshell](#).

1.1.2 Vybrané príkazy PHP a príklady ich použitia

Výpis informácií o celom systéme – operačnom, web serveri, php, nastavenia a mnoho ďalšieho - [pr1.php](#):

```
<?php
    PHPInfo();
?>
```

Tento skript neobsahuje žiadne značky HTML a iba jediný (aj to špeciálny) príkaz PHP.

Výpis hodnôt výrazov - príkaz echo – výpis reťazca ("...") alebo premennej - [pr2.php](#):

```
<HTML>
<HEAD>
<TITLE>Vypisy retazca a premennych - pr2.php</TITLE>
</HEAD>
<BODY>
<H2>Vypis retazca</H2>
<?php echo "Hello World!";?>
<br><?php $hello = "Este raz Hello World!"; echo $hello; ?>
<br>
<H2>Vypis premennych</H2>
<?php $a=1; $b=3; $c=$a+$b; echo "$a + $b = $c\n"; ?>
</BODY>
</HTML>
```

Príkaz vetvenia if – else – endif (za if a else sa píše dvojbodka) alebo jednoduchá if – else (prípadne if – elseif else) konštrukcia podľa jazyka C. Aktuálny čas a dátum vracia funkcia Date(). Ak má parameter 'A', vracia doobedu reťazec 'AM' a poobede 'PM'. S parametrom "H:i" vracia hodiny a minúty oddelené bodkou - [pr3.php](#):

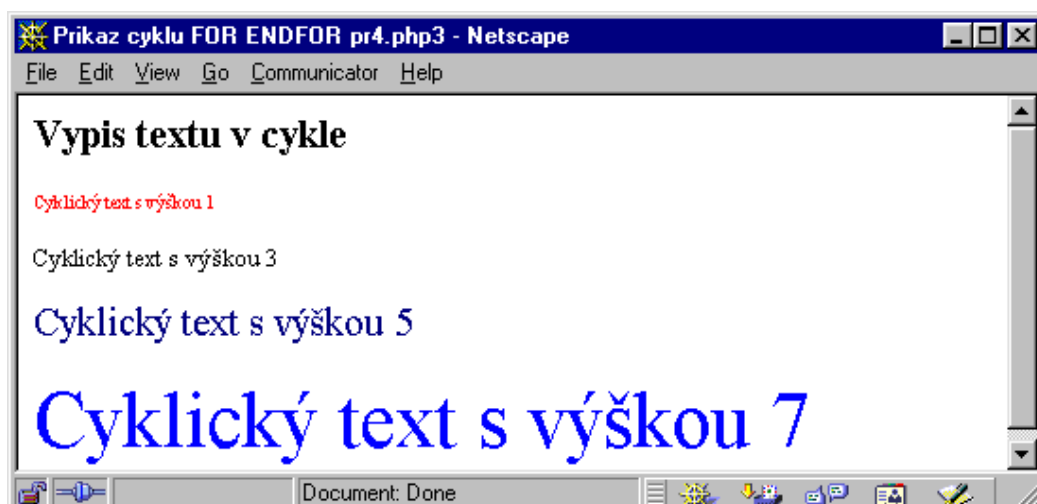
```
<HTML>
<HEAD><TITLE>Podmieneny prikaz - pr3.php</TITLE>
</HEAD>
<BODY>
<H2>Vypis AM - PM typu HTML</H2>
<?php if (date("A")=="AM"): ?>
Dobré ráno!
<?php else: ?>
Dobré popoludnie!
<?php endif ?>
Presný čas : <?php echo Date("H:i"); ?>
<H2>Vypis AM - PM typu PHP </H2>
<?php if (date("A")=="AM") {
    echo "Dobré ráno!";
} else {
    echo "Dobré popoludnie!";
} ?>
Pred chvíľou bolo : <?php echo Date("H:i"); ?>
</BODY>
</HTML>
```

Príkaz cyklu for – endfor - výpis textu s rôznymi veľkosťami a farbami písma v poli - [pr4.php](#):

```
<HTML>
<HEAD>
<TITLE>Príkaz cyklu FOR ENDFOR - pr4.php</TITLE>
</HEAD>
<BODY>
<H2>Vypis nadpisov</H2>
<?php $farby = array(1=>"red", "green", "black", "murcon", "navy", "teal", "blue") ?>
<?php for ($i=1; $i<=7; $i+=2): ?>
<P><FONT SIZE="<?php echo $i?>" COLOR="<?php echo $farby[$i]?>">
Cyklický text s výškou <?php echo $i?></FONT>
<?php endfor ?>
</BODY>
</HTML>
```

Prvý riadok skriptu tvorí pole \$farby, inicializované menami farieb. Pole je v PHP indexované od nuly, zápis '1=>' umožňuje uloženie hodnôt do poľa až od indexu 1 (index v poli farieb zodpovedá veľkosti písma). Položky poľa sú prístupné pomocou \$farby[index]. Prehliadač dostane tento výsledný HTML kód:

<P>	Cyklický text s výškou 1
<P>	Cyklický text s výškou 3
<P>	Cyklický text s výškou 5
<P>	Cyklický text s výškou 7



Zhrnutie riadiacich štruktúr PHP – príkazy vetvenia (if), cyklov (while, do – while, for), prepínač (switch) a vloženie súboru (require, include – s vyhodnotením). Pre príkazy if, while a for je možné použiť buď bežný tvar (ako v programovacom jazyku C) alebo oddeľovač : a príslušné ukončenie príkazu (endif, endwhile, endfor). To je vhodné najmä pri kombinácii PHP a HTML kódu. Príkazy break a continue prerušujú vykonávanie celého cyklu, resp. aktuálnu iteráciu s prechodom na začiatok ďalšej iterácie cyklu.

1.1.3 Spracovanie údajov formulára

Spolupráca skriptu PHP s formulárom HTML je založená na použití rovnakých identifikátorov v oboch súboroch. V HTML sú definované atribútmi NAME v jednotlivých vstupných prvkoch. Uvažujme už uvedený príklad s formulárom - [Pr5.htm](#):

```
<HTML>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
<head><title> Formulár príklad pr5.htm</title></head>
<body>
<p align=center>Prieskum PC</p>
<form method="GET" action="pr5.php">
<p>Tvoje priezvisko :
<input type="text" size=25 maxlength=30 name="meno"
value="Tak sa volam"></p><hr>
<p>Mas pocitac
<input type="checkbox" name=m1 value="ano">Ano</p>
Typ pocitaca
<input type="radio" name=m2 value="386">386
<input type="radio" name=m2 value="486">486
<input type="radio" checked name=m2 value="586">Pentium
<input type="radio" name=m2 value="686">Pentium Pro
<hr>
<p align=center>
<input type="submit" name="Odosli" value="Odoslat">
<input type="reset" name="null" value="Znova"></p>
</form>
</body>
</HTML>
```

V súbore pr5.php3 na obsluhu tohoto formulára (ACTION = PR5.PHP) sa budú používať premenné \$meno, \$m1 a \$m2 s hodnotami, ktoré zapísal používateľ do polí formulára (Tvoje priezvisko, Mas pocitac a Typ pocitaca). Veľmi jednoduchá verzia obsluhy formulára môže vyzeráť takto: ([pr5.php](#))

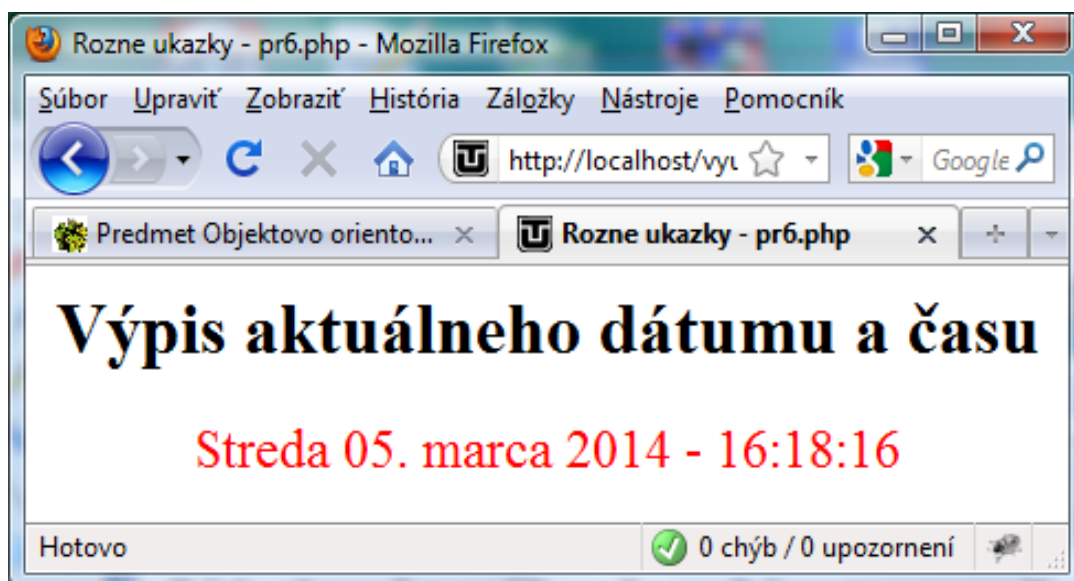
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
<TITLE>Obsluha formulára - odpoved na HTML - pr5.php</TITLE>
</HEAD>
<BODY>
<H1>Obsluha formulára Prieskum PC </H1>
<?php $meno=$_REQUEST["meno"]; $m1=$_REQUEST["m1"]; $m2=$_REQUEST["m2"];
echo "TVOJE MENO JE $meno A "
if ($m1=="ano"): echo "SETRI NA NOVY"; echo " A MAS PROCESOR $m2";
else: echo "KUP SI POCITAC ! ";
endif; ?>
</BODY>
</HTML>
```

1.1.4 Výpis aktuálneho dátumu a času

Funkcia Date() má celý rad riadiacich parametrov s týmto významom:

parameter	význam
d	číslo dňa v mesiaci (01 – 31)
D	anglická skratka názvu dňa v týždni
j	číslo dňa v mesiaci (1 – 31)
l	anglický názov dňa v týždni
N	číslo dňa v týždni ISO (1 pondelok, 7 nedeľa)
S	angl. prípona poradového čísla dňa v mesiaci, 2 znaky
w	číslo dňa v týždni (0 nedeľa, 6 sobota)
z	číslo dňa v roku
W	číslo týždňa v roku ISO, týždeň začína v pondelok
F	anglický názov mesiaca
m	číslo mesiaca (01 – 12)
n	číslo mesiaca (1 – 12)
M	anglická skratka mesiaca (3 znaky)
t	počet dní daného mesiaca 28-31
L	1 – prestupný rok, 0 – nie
o	ISO rok, má rovnakú hodnotu ako Y, okrem toho, že ak číslo týždňa ISO (W) patrí k minulému alebo budúcemu roku, použije sa tento rok
Y	rok na štyri miesta
y	rok na posledné dve miesta
a	indikátor doobedu/poobede (am/pm)
A	indikátor doobedu/poobede (AM/PM)
B	Swatch Internet time – 000 až 999
g	hodina (1 – 12)
G	hodina (0 – 23)
h	hodina (01 – 12)
H	hodina (00 – 23)
i	minúta (00 – 59)
s	sekunda (00 – 59)
u	mikrosekundy
e	časová zóna GMT, UTC
I	1 – dátum v letnom čase, 0 – inak
O	rozdiel ku GMT v hodinách, napr. +0200
P	rozdiel ku GMT s dvojbodkou medzi H:M +02:00
T	skratka časovej zóny EST, MDT
Z	posun časového pásma v sekundách
c	ISO dátum 2015-03-13T12:08:31+01:00
r	dátum v RFC Fri, 13 Mar 2015 12:08:31 +0100
U	počet sekúnd od začiatku Unixu (1.1.1970)

Parametre možno zadávať podľa požadovaného tvaru výpisu dátumu a času. Prispôsobenie výpisu na naše názvy dní možno dosiahnuť pomocou asociatívneho poľa dni (pole s indexom v tvare znakového reťazca), pre mesiace stačí obyčajné pole array, lebo Date(m) vracia priamo číslo mesiaca. Jednotlivé reťazce sa spájajú do výsledku pomocou operátora “.” (nie +). Znak vo formátovacom reťazci funkcie Date sa vracajú spolu s výsledkom, čo možno využiť na doplnenie medzier, dvojbodky a podobne.



[pr6.php](#)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
<TITLE> Actual date & time - pr6.php</TITLE> </HEAD>
<BODY>
<H1 ALIGN=CENTER> Výpis aktuálneho dátumu a času </H1>
<P ALIGN=CENTER> <FONT SIZE=5 COLOR=RED>
<?php
    $dni["Mon"] = "Pondelok";
    $dni["Tue"] = "Utorok";
    $dni["Wed"] = "Streda";
    $dni["Thu"] = "Štvrtok";
    $dni["Fri"] = "Piatok";
    $dni["Sat"] = "Sobota";
    $dni["Sun"] = "nedeľa";

    $mesiace = array(1=>"januára", "februára", "marca", "apríla", "mája", "júna",
        "júla", "augusta", "septembra", "októbra", "novembra", "decembra");

    $dnes = $dni[Date("D")] . Date(" d. ") . $mesiace[(int)Date("m")] . Date(" Y - H:i:s");
    echo $dnes;
?>
</FONT>
</BODY>
</HTML>
```

Zápis (int) znamená pretypovanie reťazcového výstupu Date() na celočíselný index poľa.

1.1.5 Obnovenie stránky

Príkaz

```
<META HTTP-EQUIV="Refresh" CONTENT="5;URL=css2.htm">
```

slúži na vykonanie odkazu na danú stránku URL po zadanom čase (5 sek) a musí byť umiestnený ako prvý v súbore php.

Ak je v príkaze pre URL uvedený názov vlastného súboru, prevedie sa odkaz na neho, dôjde teda k obnoveniu (refresh) stránky.

1.1.6 Počítanie prístupov na stránku

Je založené na inkrementovaní počítadla pri každom prístupe na stránku. Počet prístupov je uložený v súbore (databázovej tabuľke). Na adresár, kde bude uložený, musí mať PHP právo zápisu (chmod 753 meno). Premenná cesta ukazuje na tento adresár, napr.

```
$cesta = "/user2/ucitelia/FBERG/horovcak/data/";
```

Meno súboru je možné zvoliť alebo odvodiť od názvu stránky. Ten je v PHP dostupný v premennej \$SCRIPT_NAME, pričom náhradu lomítok a bodiek v názve znakom podčiarknik '_' urobí funkcia StrTr:

```
$nazov=StrTr($GLOBALS["SCRIPT_NAME"],"./\\","_");
```

Ak zvolíme pre súbor príponu .dat, úplná špecifikácia súboru bude mať tvar

```
$filename=$cesta.$nazov.".dat";
```

Ak súbor už existuje (File_Exists), treba ho otvoriť (Fopen r+), prečítať počet prístupov (FgetS), zväčšiť o 1, vrátiť na začiatok (Rewind), zapísať novú hodnotu (FputS) a zavrieť (Fclose). Prvýkrát súbor treba vytvoriť (w) a nastaviť počet na 1. Celý algoritmus je uložený v tvare funkcie v súbore counter.php, ktorý je začlenený do stránky príkazom require. Výsledný názov súboru bude counter.dat.

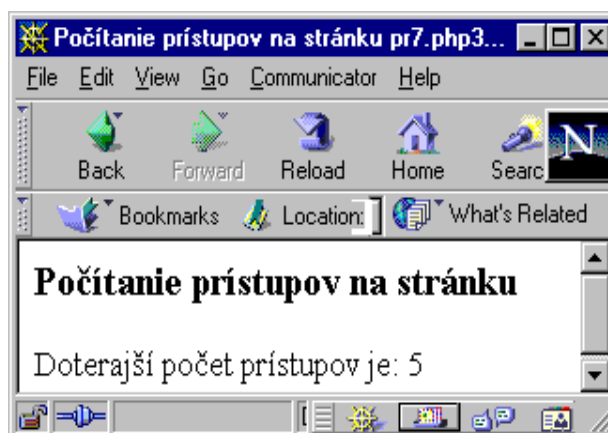
counter.php

```
<?php function GetCount() {
    $cesta=" ../ data/";           //adresar data
    // $filename=$cesta.$nazov.".dat";           //specifikacia suboru
    $filename="counter.dat";           //specifikacia suboru
    if(File_Exists($filename)):           //existuje tento subor
        $fp=Fopen($filename,"r+");           //otvorenie
        if(!$fp)return;           //neuspesne
        $count=FgetS($fp,10)+1;           //citaj pocet pristupov
        Rewind($fp);
    else:
        $fp=Fopen($filename,"w");//vytvorenie noveho
        if(!$fp)return;           //neuspesne
        $count=1;           //nastav pocet pristupov
    endif;

    FputS($fp,$count);           //zapis novy pocet
    Fclose($fp);           //zatvor subor
    echo $count;           //vypis pocu pristupov
}
GetCount();
?>
```

Umiestnenie počítadla do stránky [Pr7.php](#)

```
<HTML>
<HEAD>
<TITLE>Počítanie prístupov na stránku
pr7.php</TITLE>
</HEAD>
<BODY>
<H3>Počítanie prístupov na stránku</H3>
Doterajší počet prístupov je:
<?php require "counter.php" ?>
</BODY>
</HTML>
```



Ďalšie možnosti riešenia sú zostaviť počítadlo v tvare vloženého obrázku vo formáte GIF s využitím grafickej knižnice GD alebo komplexné databázové riešenie pre podporu štatistiky návštevnosti viacerých stránok v rôznych časových intervaloch [1].

Literatúra

- [1] Kosek J.: PHP Tvorba interaktivních internetových aplikací. Grada Praha 1999, ISBN 80-7169-373-1, 490 str.
- [2] Kosek J.: Aplikace na Webu, CW 9-46/98
- [3] Bakken S.S.: PHP Manual. <http://www.php.net>
- [4] Gabal' E.: PHP od A po Z. <http://www.zive.sk>, 2001
- [5] PHP in a (somewhat large) Nutshell [online] dostupné z <<http://phpwebdev.net/>>