# **7. Syntaxe PHP**

# **Webové stránky**

S rozšířením internetu nejprve vzniklo obrovské množství statických webových stránek. Určitě jste si nějaké webové stránky někdy zkusili vytvořit, jedná se o textový soubor, kde pomocí značek označíte určitou část jako nadpis, odstavec, obrázek a podobně. Výsledná stránka je statická. Nemůže se tedy měnit, je to pouze elektronický dokument, který můžeme jen číst.

HTML stránky jsou jednoduše uložené na serveru. Jakmile klient (uživatel s webovým prohlížečem) pošle požadavek na server, server mu jednoduše vrátí přesně tu stránku, co má uloženou. Této architektuře se říká klient-server.

# **PHP**

PHP PHP: Hypertext Preprocessor, [Hypertextový](https://cs.wikipedia.org/wiki/Hypertext) [preprocesor](https://cs.wikipedia.org/wiki/Preprocesor)“, původně Personal Home Page) je [skriptovací](https://cs.wikipedia.org/wiki/Skriptovac%C3%AD_jazyk) [programovací jazyk](https://cs.wikipedia.org/wiki/Programovac%C3%AD_jazyk). Je určený především pro programování dynamických [internetových stránek](https://cs.wikipedia.org/wiki/Webov%C3%A1_str%C3%A1nka) a webových aplikací například ve formátu [HTML](https://cs.wikipedia.org/wiki/HyperText_Markup_Language), [XHTML](https://cs.wikipedia.org/wiki/Extensible_HyperText_Markup_Language).

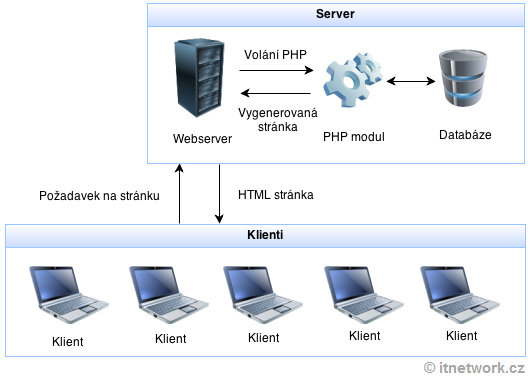
Při použití PHP pro dynamické stránky jsou skripty prováděny na straně [serveru](https://cs.wikipedia.org/wiki/Server) – k uživateli je přenášen až výsledek jejich činnosti. Interpret PHP skriptu je možné volat pomocí dotazovacích metod [HTTP](https://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol) nebo pomocí [webových služeb](https://cs.wikipedia.org/wiki/Webov%C3%A1_slu%C5%BEba).

[Syntaxe](https://cs.wikipedia.org/wiki/Syntax) jazyka je inspirována několika programovacími jazyky ([Perl](https://cs.wikipedia.org/wiki/Perl), [C](https://cs.wikipedia.org/wiki/C_(programovac%C3%AD_jazyk)), [Pascal](https://cs.wikipedia.org/wiki/Pascal_(programovac%C3%AD_jazyk)) a [Java](https://cs.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk))). PHP je nezávislý na platformě, rozdíly v různých operačních systémech se omezují na několik systémově závislých funkcí a skripty lze většinou mezi operačními systémy přenášet bez jakýchkoli úprav.

# **Webové aplikace**

Po obrovském rozmachu internetu a webových stránek se hledaly způsoby, jak do stránek přidávat nějakou dynamickou funkčnost. Tyto pokusy došly postupem času tak daleko, že v dnešní době jsme schopni dosáhnout toho, aby se webová stránka chovala úplně stejně, jako desktopová aplikace (třeba MS Word, všichni určitě znáte např. Google Docs nebo Skydrive). Takové webové stránce se říká **webová aplikace**.

Webové aplikace fungují tak, že se klient zeptá serveru na určitý dokument. Na serveru ale běží tzv. CGI skript, což je program, který dokáže vygenerovat do stránky to, co uživatel požaduje. Stránka tedy na serveru již neleží, ale je dynamicky vytvářena podle toho, co uživatel chce. Právě PHP je nejpoužívanějším CGI skriptovacím jazykem, ve kterém se webové aplikace píší.



Často je scénář následující:

1. Uživatel vyťuká do prohlížeče adresu (třeba eshop.cz/tiskarna-epson-123) a tím pošle požadavek serveru
2. Server zavolá PHP modul
3. PHP modul se podívá, co uživatel chce (tady chce vypsat informace o tiskárně). Připojí se k databázi a načte data, která chce klient. Na základě dat vygeneruje webovou HTML stránku.
4. Hotová stránka je zaslána klientovi. Ten vidí již jen statickou webovou stránku, která však byla dynamicky vytvořená podle jeho požadavku.

# **Instalace PHP**

Webové aplikace často používají linuxový operační systém, webserver Apache, databázi MySQL a programovací jazyk PHP. Tato čtveřice je často označována zkratkou LAMP. Všechny zmíněné technologie jsou zadarmo. Pokud následující technologie provozujeme pod Windows, používá se někdy zkratka WAMP.

# **IDE**

Prostředí pro PHP máme tedy připravené. Zbývá ještě stáhnout aplikaci, ve které budeme psát zdrojový kód. A ne, nebude to Notepad ani PSPad. Jelikož je PHP dynamický jazyk, lze v aplikacích udělat poměrně velké množství chyb. Potřebujeme opravdu něco chytrého, co nám bude zdrojový kód kontrolovat a aktivně napovídat.

Co se týče těch opravdu kvalitních prostředí, doporučuji IDE PhpStorm, které obsahuje mnoho funkcí šetřících čas programátora. Z těch IDE, které jsou zadarmo, je nejlepší asi NetBeans, i když není tolik chytrý a je poměrně pomalý.

# **Syntax PHP**

Základem, při psaní PHP scriptu, je HTML základ. PHP Tag se vkládá do Body.

.<!DOCTYPE html>

.<html>

. <head>

. <meta http-equiv="Content-Type" content="text/html; charset=UTF-  
. 8">

. <title></title>

. </head>

. **<body>**

. **<?php**

.

. **?>**

. **</body>**

.</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Příkaz** **echo** slouží k vypsání textu nebo proměnné:

**.<?php**

**echo**("Tento text do stránky vložilo PHP");

**.. ?>**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Proměnné** - Je to místo v paměti, kam si můžeme ukládat data a potom s nimi pracovat. Proměnné pojmenováváme libovolným názvem bez mezer a diakritiky. Před jejich název píšeme vždy dolar ($).

**.<?php**

. **$pozdrav** = "Ahoj";

. **echo**(**$pozdrav**);

.**?>**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Datové typy** - Každá proměnná je určitého typu, těmto typům se říká datové.

**.<?php**

. **$pozdrav** = "Ahoj";

. **$vek** = 15;

. **$pi** = 3.14;

.**?>**

**Pozdrav je typu string**, textový řetězec. **$vek je typu int**, což je celé číslo. **$pi je potom typu double** (nebo i **float**), tedy desetinné číslo.

PHP je tzv. dynamicky typovaný jazyk. To znamená, že datové typy nemusíme u proměnných zadávat (jako třeba v jazyce C), ale PHP si typ podle obsahu proměnné nastaví samo. Mezi typy také PHP samo převádí.

**.$a** = 10;

**.$b** = "20";

**.$c** = **$a** + **$b**;

**.echo**(**$c**); **Výstupem bude: 30**

**Sčítání a spojování Datových typů** – Sčítání jsme si již ukázali, používáme operátor (+). U spojování textových řetězců využijeme operátor (.).

**.$a** = 10;

**.$b** = 20;

**.$veta** = "Ahoj, je mi";

**.**

**.$soucet** = **$a** + **$b**;

**.$spojeni** = **$a** . **$b**;

**.echo**("Zde je součet čísel A a B: **$soucet**");

**.echo**('<br />');

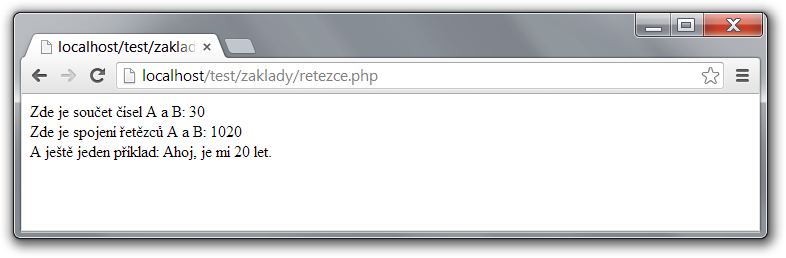
**.echo**("Zde je spojení řetězců A a B: **$spojeni**");

**.echo**('<br />');

**.echo**('A ještě jeden příklad: ');

**.echo**(**$veta** . " " . **$b** . " let.");

Výstup:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Apostrofy** - Pokud napíšeme textový řetězec do apostrofů (jednoduchých uvozovek), vloží se do něj přesně ten text.

**.$jmeno** = 'Karel';

**.$text** = 'Jmenuji se $jmeno \n S tímto "jménem" jsem spokojen.';

**.echo**(**$text**);

Výstup:

Jmenuji se $jmeno \n S tímto "jménem" jsem spokojen.

Vidíme, že si PHP vůbec nevšímá znaku dolaru, dvojitých uvozovek ani zpětného lomítka.

Správný zápis pomocí apostrofů a s pomocí teček(spojování řetězců):

**.$jmeno** = 'Karel';

**.$text** = 'Jmenuji se ' . **$jmeno** . ' \n S tímto "jménem" jsem spokojen.';

**.echo**(**$text**);

Výhodu možnosti vložení dvojitých uvozovek do apostrofů pocítíme ve chvíli, kdy vypisujeme nějaké HTML:

**.**echo('PHP se učím z tutoriálů na <a href="http://www.devbook.cz">devbooku</a>.');

**Uvozovky** - vozovky jsou v PHP chytřejší apostrofy a do řetězce zapsaném pomocí uvozovek můžeme jednoduše vkládat proměnné.

**.**$jmeno = 'Karel';

**.**$text = "Jmenuji se $jmeno \n S tímto jménem jsem spokojen.";

**.**echo($text);

Výstup:  
 Jmenuji se Karel  
 S tímto jménem jsem spokojen.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pole** - Představte si, že ve svém programu potřebujete uložit více proměnných stejného typu. Např. česká jména měsíců. Asi tušíte, že bude nějaká lepší cesta, než začít zběsile bušit proměnné mesic1, mesic2, mesic3...

Pro uložení více proměnných stejného typu nám PHP nabízí datovou strukturu, které se říká pole.

**.** /\*Příklad Pole se známkami \*/ **.**$znamky = array(1, 2, 3, 4, 2, 2, 1, 3, 2, 5);  
**.**  
**.**/\*Výpis z pole \*/  
**.** echo('Třetí známka v poli je ' . $znamky[2]);

Výstup:   
 Třetí známka v poli je 3

Průměr ze známek bychom mohli udělat následovně:

**.**$prumer = array\_sum($znamky) / count($znamky);  
**.** echo($prumer);

**Pole České datum:**

Jak již bylo řečeno, do pole si můžeme uložit prakticky cokoli, třeba názvy měsíců. Můžeme potom vypsat uživateli aktuální datum a to dokonce česky:

**.**$mesice = array('leden', 'únor', 'březen', 'duben', 'květen', 'červen',

**.** 'červenec', 'srpen', 'září', 'říjen', 'listopad', 'prosinec');

**.**$den = date('j');

**.**$mesic = date('m');

**.**$mesicSlovy = $mesice[$mesic - 1];

**.**$rok = date('Y');

**.**echo("Ahoj, dnes je $den. $mesicSlovy $rok");

V programu vytvoříme pole názvů měsíců. Dále si pomocí funkce date() zjistíme číslo dne, měsíce a roku. Do proměnné $mesicSlovy si budeme chtít uložit aktuální měsíc z pole. Když je říjen (měsíc 10), máme v proměnné $mesic hodnotu 10. Prvek v poli s hodnotou "říjen" má však index 9, jelikož jsou číslované od nuly. Proto sáhneme na přihrádku s indexem $mesic - 1. Nakonec jen jednoduše vypíšeme obsah proměnných. Kompletní práci s datem a časem se také naučíme později.

Výstup:

Ahoj, dnes je 2. říjen 2013

## Napojení na DB + Vytvoření DB

**<?php***//Připojení k DB*$servername = **"localhost"**;  
$username = **"root"**;  
$password = **"root"**;  
 $sql = **new** mysqli($servername, $username, $password);  
  
  
*// Vytvoření DB*$sqli = **"CREATE DATABASE looooooooola"**;  
**if** ($sql->query($sqli) === **TRUE**) {  
 **echo "Database created successfully"**;  
}  
**?>**

### Vytvořit tabulku

$sqlt = **"CREATE TABLE Zakaznik )  
(  
id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,   
jmeno VARCHAR(30) NOT NULL,  
prijmeni VARCHAR(30) NOT NULL,  
email VARCHAR(50),  
reg\_date TIMESTAMP  
)"**;