

# Programovací jazyk PHP

**JAZYK PHP** "HYPERTEXTOVÝ PREPROCESOR" JE INTERPRETOVANÝ, SLABĚ TYPOVANÝ PROGRAMOVACÍ JAZYK Z RODINY JAZYKA C. POUŽÍVÁ SE PRO VÝVOJ WEBOVÝCH APLIKACÍ. NENÍ TO IDEÁLNÍ JAZYK PRO POCHOPENÍ PROGRAMOVÁNÍ A VZOROVÉ ŘEŠENÍ PROGRAMÁTORSKÝCH PROBLÉMŮ. UMOŽŇUJE PROGRAMOVAT JAK PROCEDURÁLNĚ TAK I OBJEKTOVĚ. BĚŽÍ NA STRANĚ SERVERU. PHP JE OPEN SOURCE. BĚŽÍ NA PLATFORMÁCH WINDOWS, LINUX, MAC OS X, ... JE KOMPATIBILNÍ WEB SERVEREM APACHE I IIS. PODPORUJE SPOLUPRÁCI S ŘADOU DATABÁZÍ. PRVNÍ VYDÁNÍ JAZYKA V ROCE 1995. PŮVODNĚ NAPSÁN V PERLU A POZDĚJI PŘEPŠÁN V JAZYCE C, KVŮLI ZVÝŠENÍ VÝKONU. VÝVOJ JAZYKA PHP NEUSTÁLE POKRAČUJE. PHP5 PŘINESLO MOŽNOST OBJEKTOVÉHO PROGRAMOVÁNÍ. V LISTOPADU 2017 POUŽÍVALO PRO VÝVOJ WEBOVÝCH APLIKACÍ Z 83% PHP, Z 14,2% ASP.NET, Z 2,5% JAVA A DÁLE OSTATNÍ.

PHP JE POUŽIT NA WEBECH: GOOGLE.COM, FACEBOOK.COM, YAHOO.COM, WIKIPEDIA.ORG, WORDPRESS.COM, TWITTER.COM

PRO VÝVOJ A POUŽÍVÁNÍ PHP MŮŽEME NAINSTALOVAT XAMPP NEBO WAMP – PRO WINDOWS – ZPROVOZŇNÍ WEB SERVER APPACHE, PHP A MYSQL. VÝVOJOVÁ PROSTŘEDÍ PRO PHP JSOU NAPŘÍKLAD: PLACENÉ: PHPStorm, ZARMA: Aptana Studio, NetBeans, Eclipse. LZE ALE VYVÍJET I POMOCÍ CHYTRÝCH EDITORŮ: Atom, Notepad++, PSPad, ....

## ZÁKLADNÍ STRUKTURA PROGRAMU:

- PROGRAM PHP SE UKLÁDÁ DO SOUBORŮ S PŘÍPONOU PHP. NAPŘÍKLAD: test.php. PROGRAM PSANÝ V PHP ZAČÍNÁ <?php A KONČÍ ?>. TAKTO SE MŮŽE VKLÁDAT ČÁST SKRIPTŮ PŘÍMO DO HTML KÓDU. VE VELKÝCH PROJEKTECH JE ALE IDEÁLNÍ ODDĚLOVAT PHP KÓD A HTML.
- ZA KAŽDÝM PŘÍKAZEM NÁSLEDUJE STŘEDNÍK.
- PRO VÝPIS TEXTU SE POUŽÍVÁ JAZYKOVÁ KONSTRUKCE echo. POUŽÍVÁ SE NAPŘÍKLAD TAKTO: echo „toto je vypsaný text“; TAKTO MŮŽE VYPSAT I HTML TAGY.
- TEXT MŮŽE BÝT UZAVŘEN DO JEDNODUCHÝCH NEBO DVOJITÝCH UVOZOVEK. SPOJUJE SE POMOCÍ TEČKOVANÉ KONVENCE: \$a = \$b . " textik navíc"; DO DVOJITÝCH UVOZOVEK MŮŽEME VLOŽIT K TEXTU PROMĚNNOU A TA SE VYPIŠE. STEJNĚ I ZOBRAZÍ NOVÝ ŘÁDEK ATD. - \n, \t
- KOMENTÁŘE V PHP MOHOU BÝT JEDNOŘÁDKOVÉ POMOCÍ DVOU LOMÍTEK // . NEBO VÍCEŘÁDKOVÉ, KTERÉ ZAČÍNÁJÍ ZNAKY /\* A KONČÍ \*/. COKOLIV MEZI NIMI BUDE BRÁNO JAKO KOMENTÁŘ. COKOLIV CO BUDE OZNAČENO JAKO KOMENTÁŘ NEBUDE PROGRAMEM PROVEDENO.
- PROMĚNNÉ V JAZYCE PHP ZAČÍNÁJÍ DOLAREM \$, ZA KTERÝM NÁSLEDUJE NÁZEV PROMĚNNÉ SKLÁDAJÍCÍ SE Z PÍSMEN, ČÍSLIC A PODTRŽÍTKA\_. PROMĚNNÁ NEMŮŽE ZAČÍNAT ČÍSLEM. U PROMĚNNÝCH ZÁLEŽÍ NA VELIKOSTI PÍSMEN. PŘÍKLAD PROMĚNNÉ: \$pocet. PROTOŽE JE PHP SLABĚ TYPOVANÝ JAZYK, NEMUSÍME UVÁDĚT TYP PROMĚNNÉ. JEN JI ZMÍNÍME VE ZDROJOVÉM KÓDU A PŘÍŘADÍME JÍ HODNOTU. PROMĚNNÁ BUDE MÍT TAKOVÝ TYP, KTERÝ DO NÍ PŘÍŘADÍME.
- ZÁKLADNÍ DATOVÉ TYPY V PHP JSOU: STRING – TEXTOVÝ ŘETĚZEC, INTEGER – CELÉ ČÍSLO, FLOAT – DESETINNÉ ČÍSLO, BOOLEAN – LOGICKÉ HODNOTY (TRUE - PRAVDA/FALSE - NEPRAVDA), ARRAY – POLE HODNOT, ... POKUD SE PROVÁDÍ OPERACE MEZI DVĚMA PROMĚNNÝMI S ODLIŠNÝM TYPEM, PHP SE POKUSÍ AUTOMATICKY OBĚ PROMĚNNÉ PŘEVÉST NA STEJNÝ TYP, KTERÝ ODPOVÍDÁ OPERACI.
- ARITMETICKÉ OPERÁTORY: PLUS +, MÍNUS -, KRÁT \*, DĚLENO /, ZBYTEK PO DĚLENÍ %
- OPERÁTORY PŘÍRAZENÍ: = PŘÍŘADÍ HODNOTU NA PRÁVĚ STRANĚ DO PROMĚNNÉ NA LEVÉ STRANĚ. \$x = 5; PŘÍČTE, ODEČTE, ZNÁSOBÍ, VYDĚLÍ, VYTVOŘÍ MODULO ČÍSLA NĚČIM +=, -=, \*=, /=, PŘÍČTENÍ JEDNÉ \$y++, ODEČTENÍ JEDNÉ \$x--.
- OPERÁTORY POROVNÁNÍ: == JE ROVNO, === JE IDENTICKÉ, !=, <> NENÍ ROVNO, !== NENÍ IDENTICKÉ. IDENTICKÉ ZNAMENÁ, ŽE MÁ DOKONCE I STEJNÝ TYP. OPERÁTORY VĚTŠÍ >, MENŠÍ <, VĚTŠÍ NEBO ROVNO >=, MENŠÍ NEBO ROVNO <=.
- LOGICKÉ OPERÁTORY: AND, && - LOGICKÝ SOUČIN – A ZÁROVEŇ; OR, || - LOGICKÝ SOUČET – NEBO; ! NEGACE.
- POLE HODNOT MŮŽE OBSAHOVAT V JEDNÉ PROMĚNNÉ VÍCE HODNOT. INDEXOVANÉ NEBO NUMERICKÉ POLE MÁ ZA INDEX ČÍSLO, ASOCIATIVNÍ TEXT. DEFINICE POLE: \$studenti = array("David", "Roman", "Jan");, \$studenti[1] VRÁTÍ HODNOTU "Roman". INDEX ZAČÍNÁ OD 0 – TO JE PRVNÍ ELEMENT POLE. \$studenti[0] VRÁTÍ HODNOTU "David". ASOCIATIVNÍ POLE \$slide = array("David"=>"27", "Amy"=>"21", "Jan"=>"47"); VOLÁNÍ \$slide['Jan'] VRÁTÍ "47". POLE MOHOU BÝT I VÍCEDIMENZIONÁLNÍ. \$pozdrav['italsky'][2].
- PRO VKLÁDÁNÍ SOUBORŮ DO SOUBORŮ SE POUŽÍVAJÍ PŘÍKAZY include A require. NAPŘÍKLAD: include 'header.php'; KDYŽ SE POUŽÍJE require, A SOUBOR SE NENAJDE, PAK SE PROGRAM ZASTAVÍ A VYVOLÁ SE CHYBA. V PŘÍPADĚ POUŽITÍ PŘÍKAZU include, PROGRAM POBĚŽÍ VŽDY DÁL.

## ŘÍDÍCÍ STRUKTURY: JSOU STRUKTURY, KTERÉ ŘÍDÍ BĚH PROGRAMU:

**STRUKTURY PRO VĚTVENÍ PROGRAMU:** if (a > 5) { b = b+5; OSTATNÍ PŘÍKAZY; } elseif (...) { PŘÍKAZY... } else { PŘÍKAZY KTERÉ SE PROVEDOU, KDYŽ NEJSOU SPLNĚNY ŽÁDNÉ PODMÍNKY }, POVINNÝ JE JEN IF. POKUD JE PROVEDEN JEN JEDEN PŘÍKAZ, NEMUSÍ SE UVÁDĚT SLOŽENÉ ZÁVORKY.

switch (vyraz)

{ case JEDNA Z MOŽNÝCH HODNOT VÝRAZU (CELOČÍSELNÁ, ...):

PŘÍKAZY, KTERÉ SE PROVEDOU POKAŽDĚ, KDYŽ MÁ VÝRAZ HODNOTU Z NEJBLIŽŠÍHO case PŘED;  
break; // PŘESKOČÍ ZBYTEK ŘÍDÍCÍ KONSTRUKCE A UKONČÍ JI.

case JEDNA Z MOŽNÝCH HODNOT VÝRAZU (CELOČÍSELNÁ, ...):

PŘÍKAZY, KTERÉ SE PROVEDOU POKAŽDĚ, KDYŽ MÁ VÝRAZ HODNOTU Z NEJBLIŽŠÍHO case PŘED;  
break;

default:

PŘÍKAZY, KTERÉ SE PROVEDOU POKAŽDĚ, KDYŽ NIC JINÉHO NEPLATÍ;  
break;

}

continue; PRO VYNUCENÍ DALŠÍHO OPAKOVÁNÍ CYKLU A PŘESKOČENÍ ZBYTKU

break; PRO OPUŠTĚNÍ ŘÍDÍCÍ BLOKU ŘÍDÍCÍ STRUKTURY (IF, SWITCH, WHILE, FOR, ...)

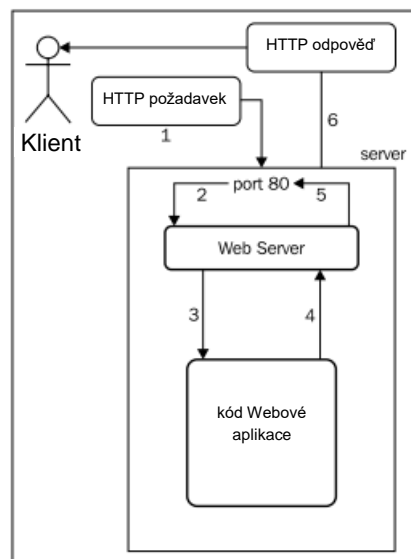
- SUPERGLOBÁLNÍ PROMĚNNÉ \$\_GET A \$\_POST JSOU POLE, KTERÁ OBSAHUJÍ ODESLANÉ PROMĚNNÉ. POST – PRO BEZPEČNÝ PŘENOS A PRO VELKÁ DATA, GET MENŠÍ DATA A PRO MOŽNOST ODKAZU DO HISTORIE. FORM – ACTION – METHOD A NAME U INPUTŮ S HODNOTAMI. PŘÍKLAD \$\_POST['jmeno'] – VRACÍ HODNOTU Z FORMULÁŘE PRO INPUT SE JMÉNEM 'jmeno'.

```
<form action="test.php" method="post"><input type="text" name="jmeno" value=""><input type="submit" name="odeslat" value="odeslat"></form>
if (isset($_POST['jmeno'])) { $jmeno = $_POST['jmeno']; echo $jmeno; } PRO GET VYPADÁ URL: www.test.cz/program.php?jmeno=napady
```

**TERMÍNY:** INTERPRET, SLABĚ TYPOVANÝ JAZYK, PROMĚNNÉ, OPERÁTORY, ŘÍDÍCÍ STRUKTURY, DATOVÉ TYPY, KOMENTÁŘE, OPERÁTORY, POLE, VÍCEDIMENZIONÁLNÍ POLE, ŘETĚZEC, CYKLUS, VĚTVENÍ PROGRAMU, ...

**PRAKTICKÁ DOVEDNOST:** VYTVOŘIT JEDNODUCHÝ PROGRAM V JAZYCE PHP S OZNAČENÍM ZAČÁTKU A KONCE, VYPSÁNÍM TEXTU. INICIALIZACE PROMĚNNÉ. ODESLÁNÍ GET NEBO POST FORMULÁŘE A ZPRACOVÁNÍ ODESLANÝCH DAT. POUŽITÍ FUNKCE isset(). POUŽÍVÁNÍ CYKLŮ FOR, WHILE, DO WHILE, POUŽÍVÁNÍ VĚTVENÍ PROGRAMU, SEŠTAVENÍ LOGICKÉHO VÝRAZU.

**DOPORUČENÉ ZDROJE:** [www.php.net](http://www.php.net), [php.vrana.cz](http://php.vrana.cz), [www.itnetwork.cz/php/](http://www.itnetwork.cz/php/), [php.baraja.cz/dokumentace](http://php.baraja.cz/dokumentace)



## ŘÍDÍCÍ STRUKTURA PRO CYKLY:

for (inicializace; podmínka; výraz - přičtení)

{ tělo cyklu (DANÝ POČET OPAKOVÁNÍ) }

while ( PODMÍNKA )

{ tělo cyklu } pokud bude podmínka vždy true, bude se cyklus opakovat do nekonečna.

CYKLUS S PODMÍNKOU NA ZAČÁTKU

do

{ tělo cyklu }

while ( PODMÍNKA );

CYKLUS S PODMÍNKOU NA KONCI