

# PHP

(v190902)

Skriptovací jazyk na strane servera - všetko sa odosiela na server, ktorý požiadavky (skripty) spracováva a výsledky predáva späť klientovi (www-prehliadaču). Výstupom php skriptu býva najčastejšie časť (prípadne celá) webovej stránky.

- **PHP skript vkladáme medzi značky <?php ?>**
- **príkazy sa oddeľujú bodkočiarkou ;**
- priradenie (hodnoty) je =
- porovnanie (premenných) je ==
- názov každej premennej začína znakom \$
- je CASE sensitive, tzn. premenné \$pocet, \$Pocet a \$POCET sú tri rôzne premenné

## Vloženie PHP skriptu do stránky:

Zdrojový kód	Ukážka výsledného HTML kódu
<pre>&lt;body&gt; &lt;h1&gt;Super stránka s dátumom&lt;/h1&gt; Dnes je &lt;b&gt;&lt;?php echo Date("w. m. Y");?&gt;&lt;/b&gt;&lt;br&gt; Budúcnosť je už teraz &lt;/body&gt;</pre>	<pre>&lt;body&gt; &lt;h1&gt;Super stránka s dátumom&lt;/h1&gt; Dnes je 4. 10. 2032 Budúcnosť je už teraz &lt;/body&gt;</pre>

## Výpis do stránky – príkaz echo:

- `echo "Hello world!<br>This is PHP!\n";`
- `echo "Vysledok je <b>$cislo centimetrov</b>.";`
- `echo "Vysledok je <b>". ($cislo-5*$pocet). "centimetrov</b>.";`
- `echo "<img src=\"obrazok.png\" alt=\"$premenna\">";`
- `echo '';`
- `echo Date("w. m. Y");`

**Textové reťazce** – môžu byť v apostrofoch ('retazec v apostr.') aj úvodzovkách ("retazec v uvodz."). Rozdiel je v tom, že v reťazci v úvodzovkách sa premenná vyhodnotí, ale v apostrofoch nie.

```
$str1 = "Ahoj" . " Fero"; //zreťazenie stringov (pomocou bodky)
$ Cislo = 123;
$str2 = "<b>Výsledok je $cislo</b>";
```

## Komentáre:

- `// komentáre do konca riadka`
- `/* viacriadkové komentáre */`

## Premenné:

- musí začínať znakom \$, napr. \$pocet
- nedefinujeme typy, php si ich definuje samo (automaticky), existujúce typy: integer (\$cislo = 123), double (\$realne = 12.3, string (\$str1 = "Ahoj", \$str2 = 'IKT 5')
- `isset(premenna)` – vráti TRUE, ak bola premennej priradená nejaká hodnota
- `empty(premenna)` – opak `isset()`
- `unset(premenna)` – odstránenie premennej a uvoľnenie z pamäti

- GetType(premenna), SetType(premenna, typ) – zistenie, resp. nastavenie typu premennej
- Is\_Integer(premenna), Is\_Double(premenna), Is\_String(premenna), Is\_Array(premenna), Is\_Object(premenna) – zistenie, či premenná je daného typu

### Operátory:

```
$sucet = 5 + 4; $rozdiel = 9 - 3; $nasob = 12 * 3; $del = 13 / 3;
$zvysok = 14 % 5; //modulo
$suc = $suc + 1; $suc++;
$suc = $suc + 5; $suc += 5;
$x == .,<,>,<=,>= $y //porovnanie
$x && $y //a súčasne
$x || $y //alebo
$x != $y //nerovná sa
```

### Vetvenie:

1. spôsob zápisu príkazu if	2. spôsob zápisu príkazu if	príkaz switch (nahradenia viacerých príkazov if)
<pre>if (podmienka) { zoznam_prikazov1; } else { zoznam_prikazov2; }</pre>	<pre>if (podmienka) : zoznam_prikazov1; elseif (podmienka2) : zoznam_prikazov2; else : zoznam_prikazov3; endif;</pre>	<pre>switch(výraz): case "hodnota1": príkazy 1; break; case "hodnota2": príkazy 2; break; default: príkazy 3; endswitch;</pre>

**Ternárny operátor:** podmienka ? hodnota-ano : hodnota-nie

Príklad: `echo ($c1>$c2) ? $c1 : $c2;`

Príklad 2: `echo "Cislo ".(($c1>$c2) ? $c1 : $c2)." Je vacsie";`

### Rôzne typy cyklov:

zdrojový kód	ukážka
<pre>while(podmienka) { zoznam_prikazov }</pre>	<pre>\$i = 0; while (\$i &lt; 10) { echo "\$i&lt;br /&gt;"; \$i++; }</pre>
<pre>do { zoznam_prikazov } while (podmienka);</pre>	
<pre>for (prikaz1; prikaz2; prikaz3) { zoznam_prikazov }</pre> <p><b>prikaz1</b> – vykonáva sa iba na začiatku cyklu (inicializačná hodnota/podmienka – napr. priradenie nuly – začínaj pre počítadlo od nuly)  <b>prikaz2</b> – ak je TRUE, zoznam príkazov sa vykoná (podmienka ukončenia)  <b>prikaz3</b> – príkazy sa vykonajú na konci každého cyklu (zvyčajne zvyšovanie počítadla)</p>	<pre>for(\$i = 0; \$i &lt; 10; \$i++) { echo "\$i&lt;br&gt;"; }</pre>