

21. Algoritmizace - podprogramy - procedury a funkce (s parametry a bez)

Procedury a funkce

Procedury a funkce tvoří posloupnosti instrukcí, které potřebujeme v programu provádět na různých souborech dat nebo na různých místech programu. Procedura nebo funkce může být po deklaraci použita kdekoli v následujícím textu bloku programu. Rozdíl mezi procedurou a funkcí je v tom, že funkce vrací hodnotu a může se použít přímo ve výrazech. Procedura se vyvolá příkazem volání procedury ke splnění jedné nebo více operací.

Procedury a funkce umožňují vnořovat přídavné bloky do hlavního programového bloku. Každá deklarace procedury nebo funkce má hlavičku, po které následuje blok příkazů. Procedura se aktivuje příkazem volání procedury, funkce se aktivuje vyhodnocením výrazu, který obsahuje její volání. Za identifikátorem procedury nebo funkce může následovat seznam parametrů, uzavřený v kulatých závorkách. Každý parametr v seznamu má své místo, kterému odpovídá příslušný typ.

Použití funkcí přispívá nejenom k rychlejšímu vývoji programů, ale především celý programový kód výrazně zpřehledňuje. A v neposlední řadě můžeme už jednou naprogramované funkce znovu využít v budoucích programech a podstatně tak zkrátit dobu programování.

PHP - funkce

Funkce je jakousi zkratkou pro větší počet procesů, pokud je nechceme vypisovat celé.

Deklarace funkce

Pokud umíte pracovat s funkcemi v Javascriptu, budou pro vás funkce v PHP snadné. Kromě předem zavedených a vyhrazených funkcí PHP, si můžete deklarovat své vlastní:

```
function napis(){  
echo("<p>Ahoj!</p>");  
}
```

function deklaruje funkci, `napis` je název funkce, (názvy pište bez diakritiky). U názvu funkce je třeba si dát pozor, aby se její název nekryl s názvem již zavedené funkce PHP (nemůžete použít např. `echo`, `return`...). V závorkách jsou uvedené argumenty funkce. /Funkce může být i bez argumentu. Složené závorky { } vymezují procesy funkce (složené závorky zapíšete pomocí SHIFT + [ú] a SHIFT + []) nebo pomocí PRAVÝ ALT + [F] a PRAVÝ ALT + [G]).

Kdykoli zavoláme tuto funkci skript vypíše odstavec ahoj. Taková funkce se nám může hodit, pokud třeba chceme opakovaně vypisovat nějaký delší text, nebo část zdrojového kódu.

Vyvolání funkce

Nyní jsme deklarovali funkci, ale teď ji chceme vyvolat:

```
napis();
```

Zapíšeme pouze název funkce a závorky. Skript vypíše 'ahoj'.

Funkce bez argumentu (procedury)

Je již zmíněná funkce napis(), která nemá uvnitř závorek žádný argument(proměnnou).

Funkce s argumentem (bez návratové hodnoty, procedury)

Kdybychom chtěli někdy napsat 'ahoj', jindy 'dobrý den', nebo něco úplně jiného, použijeme funkci s argumentem.

```
function napis($jmeno){
echo "<p>Tvé jméno je $jmeno</p>";
}
napis("dobrý večer"); //napíše dobrý večer
napis("ahoj"); //napíše ahoj
napis("dobré ráno"); //napíše dobré ráno
```

Deklarace je úplně stejná až na argument \$jmeno. Ten zavádí proměnnou \$jmeno. Když jsme vyvolali funkci a jako její argument jsme uvedli "dobrý večer" a "ahoj", byly tyto hodnoty přiřazeny k proměnné \$jmeno, následně je funkce zapíše.

Funkce vracející hodnotu, příkaz return (funkce v užším slova smyslu)

Můžete také zavést funkci, která bude vrací hodnotu:

```
function vrat($cislo){
return $cislo*2;
}
echo(vrat(20));
echo(vrat(100)/vrat(10));
```

Funkce vrat(); příkazem return vrátí \$cislo*2 (tedy vrací dvojnásobek zadané hodnoty). S danou funkcí lze potom nakládat jako s proměnnou, počítat s ní, vypisovat ji atd.