

Desenvolvimento Web II

PHP Básico

Funções

- Conjunto de instruções utilizadas para uma tarefa específica;
- Podem receber ou não parâmetros;
- Podem gerar ou não uma saída;
- Pode ser utilizado para segregar o código (importante para o paradigma estrutural da linguagem)

Funções

- Em PHP:
 - Como melhores práticas, usar camelCase;
 - Não é necessário declarar o retorno;

```
<?php  
function soma($a,$b)  
{  
    return $a+$b;  
}  
?>
```

Funções

- Em PHP:
 - Como melhores práticas, usar camelCase;
 - Não é necessário declarar o retorno;
 - Pode estar em qualquer lugar do código (inclusive no final), desde que o programa passe por aquele trecho durante a execução.

Funciona

```
<?php
echo soma(4,2);
function soma($a,$b)
{
    return ($a+$b);
};
?>
```

Não Funciona

```
<?php
echo soma(4,2);
if (false){
    function soma($a,$b)
    {
        return ($a+$b);
    };
}
?>
```

Funções

- Em PHP:
 - Não é case-sensitive; Ex.: echo, Echo
 - Variáveis passadas como parâmetros para a função são copiadas para outro espaço da memória.
 - Estrutura de uma função:

```
<?php
Function nomeDaFuncao($arg1,$arg2, $arg3, $argn){
    return($arg1+$arg2+$arg3+$argn);
}
?>
```

Funções

- Em PHP:
 - Não permite sobrecarga de métodos, Porém permite parâmetros opcionais.

Exemplo do uso de parâmetros opcionais
(funciona no PHP)

```
<?php
function soma($a, $b, $c = null){
    if($c == null)
        return $a+$b;
    else
        return $a+$b+$c;
}
echo soma(1,3)."\n".soma(1,3,3);
?>
```

Exemplo de Sobrecarga de métodos
(Não funciona no PHP)

```
#include <iostream>
using namespace std;
int soma(int a, int b)
{
    return a + b;
}
int soma(int a, int b, int c)
{
    return a + b + c;
}

int main()
{
    //cout << 21 + 12;
    //return 0;
    cout << soma(1,4)<<"\n";
    cout << soma(1,4,4);
}
```

Include

- Função utilizada para incluir outros arquivos (php, html) no programa atual;
- Permite a segregação do código em arquivos diferentes (organizar o código).
- Somente a partir do include no programa o programa chamado passa a ser reconhecido pelo programa atual.

```
<?php

$color = 'green';
$fruit = 'apple';

function soma($arg1, $arg2)
{
    return $arg1+$arg2;
}
?>
```

Arquivo exemplo.php

```
<?php
echo "A $color $fruit"; // A
echo soma(1,1); //erro
include 'exemplo.php';
echo soma(1,1);
echo "A $color $fruit"; // A green apple
?>
```

Include x Require

- Require possui a mesma função do include, porém:
- Na ausência do arquivo incluído, a aplicação emite um aviso (E_WARNING) que permite que o script continue.
- Na ausência do arquivo incluído, a falha produzirá um erro fatal de nível E_COMPILE_ERROR que irá interromper o script.

Include_once require_once

- As expressões `require_once` e `include_once` são idênticas as expressões `require` e `include` respectivamente, exceto que o PHP verificará se o arquivo já foi incluído e, em caso afirmativo, não o incluirá (requererá) novamente;
- Ideal para boas práticas de programação.