



Construção. web sites

Prof: José

# Funções

Uma função é um trecho de código encapsulado com objetivo específico que pode (ou não) receber parâmetros, utilizar qualquer parte do PHP (incluindo outras funções) para manipular os dados e retornar um valor de qualquer tipo.

O código dentro de uma função é encapsulado do código externo (e vice-versa), ou seja, a não ser que você queira, nada que estiver entre as chaves { e } (ações da função) é acessível ao código fora delas.

# Criando uma função em php

Para declarar uma função em PHP, utilizamos o operador "**function**" seguido do nome que desejamos, os argumentos (parâmetros) que a função poderá receber, e, por fim, as ações vêm entre chaves { e }.

```
// Função com dois parâmetros
function nome_da_funcao ( $parametro1, $parametro2 ) {
// Cria uma variável local $retorno
$retorno = $parametro1 + $parametro2;
// Retorna a soma do parâmetro 1 e 2
return $retorno;
}
```

Perceba que a palavra "return" (retorno) serve para retornar o valor que desejamos depois de executar as ações dentro da função. O interpretador do PHP para de ler o restante da função assim que encontrar a palavra "return", por este motivo, na grande maioria dos casos utilizamos essa palavra ao final da função para retornar o valor que desejamos.

função não executa nenhuma ação se ela não for chamada.

Para chamar uma função, utilizamos apenas seu nome, e os parâmetros entre parênteses e separados por vírgula.

Você também pode criar funções sem parâmetro, ou sem a palavra "return"

# Parâmetros dinâmicos

Caso seja necessário criar funções que tenham vários parâmetros criados dinamicamente, basta criar uma função sem parâmetros e enviar vários argumentos (valores) quando for chamar a mesma.

O PHP tem várias funções para extrair valores enviados para a função, como [func\\_get\\_args](#), que retorna um array contendo os argumentos enviados na chamada da função.

Veja um exemplo:

```
function exibe_parametros () {  
    $paramentos = func_get_args();  
    foreach ( $paramentos as $posicao => $valor ) {  
        echo 'Parâmetro ' . $posicao . ':' . $valor . '<br>';  
    }  
}  
exibe_parametros('Valor 1', 'Valor 2', 'Valor 3');
```

# Tabelas HTML

As tabelas são definidas com a tag <table>. Uma tabela é dividida em linhas (com a tag <tr>), e cada linha é dividida em células de dados (com a tag <td>). As letras td significam "table data," que é o conteúdo de uma célula de dados. Uma célula de dados pode conter texto, imagens, listas, parágrafos, formulários, filetes horizontais, tabelas, etc.

```
<table>
  <tr>
    <td>linha 1, célula 1</td>
    <td>linha 1, célula 2</td>
  </tr>
  <tr>
    <td>linha 2, célula 1</td>
    <td>linha 2, célula 2</td>
  </tr>
</table>
```

A tabela exposta até agora leva um padrão de informalidade. Algumas tabelas são definidas com os atributos “thead”, “tbody” e “tfoot, ficando aproximadamente

# Exercício

- Crie uma função que some todos os parâmetros passados para ela Ex: soma(1,2,3) resposta 6 soma(2,2) resposta 4
- Crie uma função que receba um array de parâmetro e imprima um select com todos os valores do array
- Crie uma função que retorne se o valor é par ou impar



# Exercício

- Crie uma pagina html que devera apresentar um calendário
- Crie uma função que calcule o fatorial de um numero
- Crie uma função q dado o ano de nascimento calcule a idade da pessoa