**Manual do iniciante 7.0- PHP**

**Comandos do banco de dados**

**Comandos citados**

**Nomenclatura criada para teste**

**INTRODUÇÃO PHP**

Todo comando PHP é escrito entre a tag <**?PHP ?** > tudo que você deseja visualizar na tela pode ser feito utilizando o comando **ECHO** e a mensagem dentro de aspas duplas finalizando com ponto e vírgula.

**<? PHP**

**ECHO “VALID PHP”;**

**?>**

**VARIÁVEIS**

Todas variáveis escrita em **PHP** recebe cifrão “**$**” ao ser escrita não deve utilizar caracteres especial o tipo de variável é definido pelo tipo de dados que ela recebe caso precise definir o tipo utiliza o **TYPECAST** que pode ser **INT**, **FLOAT**, **DOUBLE**, **STRING**.

**$IDADE=3;**

**$IDADE= (INT)3;**

A concatenação no PHP pode ser feita através de pontos onde a STRING são colocadas entre aspas e as variáveis solta antes do ponto e vírgula.

**ECHO “IDADE: ”. $IDADE;**

Também pode ser feito através de aspas duplas colocando dentro das aspas duplas sem diferenciando apenas o que é variável pelo cifrão.

**ECHO “IDADE: $IDADE”;**

**OPERADORES**

**Operadores de atribuição**

O operador de atribuição sinal de **IGUALDADE** serve para atribuir valores a variável.

**$valor = $valor\_compra**

O operador de atribuição de **MAIS** seguido do sinal de **IGUAL** serve para atribuir um valor a uma variável somando com o valor já existente na variável.

**$valor += $valor\_compra**

O operador de atribuição de **SUBTRAÇÃO** seguido do sinal de **IGUAL** serve para subtrair um valor a uma variável do valor total existente na variável.

**$valor += $valor\_compra**

O operador de atribuição de **MAIS** **MAIS** ou **MENOS** **MENOS** serve para somar o mais um ou subtrair um da variável.

**$C ++** ou **$C --**

**Operadores aritméticos**

Adição **+**, subtração **-**, multiplicação **\***, divisão **/**, modulo **%**

**OPERADORES LÓGICOS**

Maior quer

**$MAIOR > $QUER**

Menor quer

**$MENOR < $QUER**

Maior ou igual a quer

**$MAIOR OU IGUAL >= $QUER**

Menor ou igual a quer

**$MENOR OU IGUAL <= $QUER**

Diferente de

**$DIFERENTE <> $DE** ou **$DIFERENTE != $DE**

Igual a – mesmo valor independentemente do tipo

**$IGUAL == $A**

Idêntico a - mesmo valor e mesmo tipo

**$IDÊNTICO === $A**

**SWITCH**

**SWITCH** serve para uso de múltipla escolha exemplo caso o valor da variável for **X** então execute ação **B** ou seleção de grupos de casos para executar determinada ação, para criar uma estrutura **SWITCH** utiliza uma variável e em seguida as **CASE** caso a variável seja esse valor faça isso e finaliza com **BREAK** cada opção ou grupo de opção, para a última **CASE** caso não tenha a opção **DEFAULT** não é necessário o uso do **BREAK**;

**<? PHP**

**$VALOR=$\_GET["VALOR"];**

**SWITCH ($VALOR)**

**{**

**CASE 1:**

**ECHO"OPÇÃO 1";**

**BREAK;**

**CASE 2:**

**ECHO"OPÇÃO 2";**

**BREAK;**

**CASE 3:**

**ECHO"OPÇÃO 3";**

**DEFAULT;**

**ECHO “RESULTADO ALTERNATIVO”;**

**};**

**?>**

**WHILE**

A função **WHILE** serve para cria um **LOOP** de execução até que seja atendido a condição determina para ele para isso utiliza operadores de atribuição para que em cada volta do **LOOP** variável esteja com um valor diferente, exemplo de uso.

**<? PHP**

**$VALOR=$\_GET["VALOR"];**

**WHILE ($VALOR<10)**

**{$VALOR ++;**

**ECHO"VALOR: $VALOR<BR>";};**

**?>**

**FOR**

A função FOR serve para mesmo propósito do WHILE criar loop com diferença que é determinado nele a variável o valor que deseja como verdadeiro e ação caso não seja após isso vem o bloco de comando que o loop deve executar.

**<? PHP**

**$VALOR=$\_GET["VALOR"];**

**FOR ($VALOR; $VALOR<10; $VALOR ++)**

**{**

**ECHO"VALOR: $VALOR <BR>";**

**};**

**?>**

**FOREACH($VARIAVEL\_ARRAY AS $VARIAVEL\_QUE\_RECEBERA VLOR DA ARRY A CADA RODADA){}**

**FOREACH($VARIAVEL\_ARRAY AS $INIDICE => $VARIAVEL\_QUE\_RECEBERA VLOR DA ARRY A CADA RODADA){}**

**FUNÇÃO**

Função são usadas para analisar dados e devolvendo determinados valores de acordo com a função são elas funções de **STRING**, **NÚMERICA**, **BOOLEANAS**...

Exemplos de funções booleanas:

A função booleana testa os valores disponibilizado devolve valores **1** para verdadeiros e **0** para falsos.

**IS NUMBER (1)** // nessa função ele testa se o valor inserido é numérico e devolve no caso **1** como verdadeiro.

Exemplos de funções string:

A função de string trabalha com dados de caracteres.

**STRLEG (“ABCDE”)** // nessa função ele conta a quantidade de caracteres que existe nos dados fornecido nesse caso o retorno de **5.**

Exemplos de funções string:

As funções numéricas tratas o **INTEGER** ou os **FLOATS**.

**ABS (-15)** // nessa função ele converte os números fornecidos casos seja negativo em números positivos nesse caso o retorno seria 15.

Para criar uma função é necessário declara **FUNCTION** seguido do **NOME** da função entre parênteses as **VARIÁVEIS** e entre chaves a **AÇÃO** executada nessas variáveis e o valor que será retornado por essa variável.

**FUNCTION SOMA ($A, $B) {$C=$A+$B; RETURN $C;};**

Para chamar a função criada utiliza o comando o nome da função e entre parêntese a parametrização necessária.

**SOMA (1,2);**

Exemplos de funções

**ABS (), POW () AO CUBO, SQRT (), ROUD (), CEIL (), FLOOR (), NUMBER\_FORM (,2)**

**IF-ELSE-ELSE IF**

Para utiliza o é necessário declara o **IF** e entre parênteses a condição necessária para que aquele **IF** seja executado dentro de chaves a execução.

**IF ($A>$B) {ECHO “A MAIOR QUE B”;}**

Caso não atenda aquela condição utiliza o **ELSE** para que seja determinada outra condição opcional e caso precise mais de uma utiliza o **ELSE** **IF** e por último sempre o **ELSE**.

**IF ($A>$B) {ECHO “A MAIOR QUE B”;}**

**ELSE IF ($A<$B) {ECHO “A MENOR QUE B”;}**

**ELSE {ECHO “ A IGUAL B”;};**

**DATA**

As datas são projetadas como uma variável ARRAY então sua projeção tem que ser feita através do PRINT\_R(); e sua projeção sem nenhuma modificação da data atual como exemplo

**$DATE=GETDATE ();**

**PRINT\_R ($DATE);**



Sua projeção fica muito extensa e dividida para que as data não venha nesse formato é usado comandos onde o formata a data vem apenas a parte da ARRAY desejada

**PRINT\_R ($DATE[YEAR]);**

**RETORNO 2019**

**ARRAY**

**(**

**[SECONDS] => 40**

**[MINUTES] => 58**

**[HOURS] => 21**

**[MDAY] => 17**

**[WDAY] => 2**

**[MON] => 6**

**[YEAR] => 2003**

**[YDAY] => 167**

**[WEEKDAY] => TUESDAY**

**[MONTH] => JUNE**

**[0] => 1055901520**

**)**

https://www.php.net/manual/pt\_BR/function.getdate.php

Para definir a data de sua região utiliza comando com a cidade onde está entre os parenteses

**DATE\_DEFAULT\_TIMEZONE\_SET ('AMERICA/FORTALEZA');**

Uma função também usada para formatar datas da maneira que deseja é a

**STRFTIME (COM A VARIÁVEL, E SEU PARAMTRO DE FORMATÇÃO)**

**https://www.php.net/manual/pt\_BR/function.strftime.php**

É para que a descrição fique em português utiliza o comando

**SETLOCALE (LC\_TIE, “PT-BR”)**

**Conexão PHP com MYSQL**

Para criar uma conexão PHP com MYSQL é necessário utiliza uma variável para receber essa conexão atribuindo uma conexão com parâmetros de servidor, usuário, senha e banco para sempre que necessário utilizar essa conexão.

**$CONEXAO = MYSQLI\_CONECT(“LOCALHOST”,”ROOT”,”1234”,”BANCO”);**

Para identificar quando ocorreu durante a conexão utilizado a função **MYSQLI\_CONNECT\_ERRNO ()**

**IF (MYSQLI\_CONNECT\_ERRNO ()) {**

**DIE ("CONEXÃO FALHOU”. MYSQLI\_CONNECT\_ERRNO ())};**

**SELECT COM PHP**

Para que seja feita uma consulta no banco com PHP é necessário utiliza uma variável para receber o comando que irá ser executado no banco de dados em seguida criar uma variável onde vai receber o resultado do comando utilizando o comando **MYSQLI\_QUERY**.

**$QUERY=”SELECT \* FORM TB\_SETOR; ”;**

**$CONEXAO = MYSQLI\_CONECT (“LOCALHOST”, ”ROOT”, ”1234”, ”BANCO”);**

**$RESULTADO=MYSQLI\_QUERY ($CONEXAO, $QUERY);**

Caso deseje vê se ocorreu algum erro utiliza o comando **IF** com o comando **DIE** que é utilizado para imprimir quando ocorre erro de conexão.

**IF (! RESULTADO) {DIE (“ERRO DE CONEXÃO”);}**

**INSERT COM PHP**

Para fazer um INSERT em uma tabela no banco primeiro é necessário ter uma conexão configurada após isso é preciso declara dentro de uma variável o que o comando de INSERT ou a PROCEDURE

**$CONSULTA=”CALL PR\_SETOR (‘$VALOR’); ”;**

Logo após determinar o valor para variável utiliza a variável de conexão com a variável de INSERT junto com a função MYSQLI\_QUERY().

**UTILIZANDO COMANDO DE CONEXÃO PHP ENTRE PÁGINAS**

Require\_once (página.php) serve para que possa ser solicitado uma página de conexão em PHP.

INCLUDE\_ONCE(PÁGINA PHP) para solicitar umá página PHP na tela.

$valor = Condição ? verdadeiro : falso

$= array(‘’,) []

Valor nulos ?: isset operador ternário

Conexão

$CONETAR=mysqli\_conect(“locahost”,”root”,”senha”,”bancodedados”)

Mysqli\_close($conec)

MYSQL\_FECTH\_ROW($VARIAVEL) ASSOC

Mysqli\_free\_result()

Require\_once(“”);

Header(location: p.php);