

# Parte 1

## 1) Responda com Verdadeiro ou Falso

### (F) PHP não tem variáveis com tipos

As variáveis em php não tem tipo definido. Elas assumem o tipo dos valores que a elas forem atribuídos.

### (F) PHP é uma linguagem embarcada

PHP é uma linguagem de script Open Source, podendo ser “misturada” ao fonte html. Pode-se dizer que php é uma linguagem embutida.

### (F) O caracter ";" é opcional em PHP, como em Javascript

Não é opcional, é obrigatório.

### (V) O tempo de vida de uma variável e definições de função ultrapassam o escopo de um arquivo

### (V) echo não é uma função, é uma construção da linguagem

"echo()" não é uma função atualmente (construtor da linguagem). Portanto, não é obrigatório usar parênteses. De fato, se você quer passar mais do que um parâmetro para echo, você não deve fechar os parênteses dentro de parênteses.

### (V) loops que usam *for* fazem pré-teste de condição, enquanto loops que usam *do/while* fazem pós-teste

---

## 2) Arquivos PHP têm normalmente que extensão?

"(Nome\_do\_arquivo).PHP"

---

## 3) Como devemos terminar um arquivo PHP para instruir o servidor a produzir código fonte?

"?>"

---

## 4) PHP é uma linguagem:

- ☐ interpretada
  - ☐ compilada
  - ☒ ambos
- 

## 5) Responda:

**Qual a função do operador %?**

Operador que retorna módulo, ou "resto". Ex.: \$a % \$b (Resto de \$a dividido por \$b).

**Qual a função do operador =?**

Operador de atribuição. Ex.: o valor de "\$a = 3" é 3.

**Qual a função do operador ==?**

Operador de igualdade. Verdadeiro (TRUE) se \$a é igual a \$b.

**Qual a função do operador ===?**

Verdadeiro (TRUE) se \$a é igual a \$b, e eles são do mesmo tipo (somente para PHP4).

**Qual a função do operador !==?**

Verdadeiro se \$a não é igual a \$b, ou eles não são do mesmo tipo (somente para o PHP4).

**Para que serve o \$?**

As variáveis no PHP são representadas por um cifrão (\$) seguido pelo nome da variável. Os nomes de variável no PHP fazem distinção entre maiúsculas e minúsculas.

**O que significa uma linguagem que é dita "fracamente tipada"?**

Como as variáveis não precisam ser declaradas, qualquer tipo pode ser atribuído a uma dada variável.

**Qual é a diferença entre "\$d" and '\$d'?**

"\$d" retorna o valor da variável \$d, enquanto '\$d' é a string \$d.

---

**6) Responda com Verdadeiro ou Falso**

**(F) \$123 é um identificador válido**

Não pode começar com número

**(V) \$\_umavariavel é um identificador válido**

**(V) \$abcdefghijklmnop é um identificador válido**

---

**7) Escreva um loop que gere a seguinte série de nmeros: 1 3 5 7 9 11**

```
for ($i = 0; $i <= 5; $i++) {  
    $j = 2*$i + 1;  
    echo $j." ";  
}
```

1 3 5 7 9 11

---

**8) Escreva um loop que gere a seguinte sequência de nmeros: 80 75 70 65 60 55 50 45 40**

```
for ($i = 85; $i >= 45 ; $i) {
    $i = ($i - 5);
    echo $i." ";
}
```

80 75 70 65 60 55 50 45 40

---

### 9) Escreva um loop que gere a seguinte série de asteriscos:

```
*
**
***
****
*****
```

```
$a = '*';
for ($i = 1; $i <= 5; $i++) {
    echo $a."<br>";
    $a = $a.'*';
}
```

```
*
**
***
****
*****
```

---

### 10) Escreva um loop que gere a seguinte série de asteriscos:

```
***
****
*****
*****
*****
*****
***
```

```
$a = "*****<br>";
$b = $a;
for ($i = 1; $i <= 3; $i++) {
    $a = substr($a, 1);
    $b = $a.$b.$a;
}
echo $b;
```

```
***
****
*****
*****
```

\*\*\*\*\*

\*\*\*\*\*

\*\*\*

---

**11) Escreva um loop que gere a seguinte saída:**

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

```
$b = "";
for ($i = 1; $i <= 6; $i++) {
    $a = $i;
    $b = $b." ".$a;
    echo $b."<br>";
}
```

(ou)

```
$a = '1';
for ($i = 1; $i <= 6; $i++) {
    echo $a." ";
    $b = $a + $i;
    $a = $a." ".$b;
}
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

---

**12) Escreva um loop que gere a seguinte saída:**

```
(1 2)(2 3)(3 4)(4 5)(5 6)
(1 2)(2 3)(3 4)(4 5)
(1 2)(2 3)(3 4)
(1 2)(2 3)
(1 2)
```

```
for ($j = 5; $j >= 0; $j--) {
    for ($i = 1; $i <= $j; $i++) {
        $a = $i;
        $b = $i + 1;
        echo "(".$a." ".$b.") ";
    }
}
```

```
    echo "<br>";
}
```

(1 2) (2 3) (3 4) (4 5) (5 6)

(1 2) (2 3) (3 4) (4 5)

(1 2) (2 3) (3 4)

(1 2) (2 3)

(1 2)

---

### 13) Escreva um trecho de programa que identifique se um determinado número é um telefone válido no estado do Rio de Janeiro:

```
echo "Escreva o numero do telefone:";
echo"
<form action=\"parte_1.php\" method=\"post\"><br>
<input name=\"pre\" size=\"2\" maxlength=\"2\" type=\"text\"><br>
<input name=\"tel\" size=\"9\" maxlength=\"9\" type=\"text\"><br>
<input type=\"submit\"><br>
</form>
";

<?php
$pre = $_POST['pre'];
$tel = $_POST['tel'];
$confirma_tel = ereg("^([0-9]{4})(-[0-9]{4})?$", "$tel");

if ($confirma_tel == true && $pre == "21"){
    echo "<b>".$pre." ".$tel."<br>";
    echo "Telefone válido";
}
else {
    echo "<b>".$pre." ".$tel."</b><br>";
    echo "Telefone inválido";
}
?>
```

## Parte 2

### 1) Numa arquitetura de três camadas:

- a) Que programa é executado na camada do cliente?
- b) Cite uma vantagem de clientes leves em comparação com clientes "pesados".
- c) Que camada é responsável pela maior parte da lógica de uma aplicação web?
- d) Que camada é responsável pelo armazenamento da maior parte dos dados?

- a) O Navegador. Ex. Mozilla Firefox, IE, Opera.
  - b) Um cliente "leve" é um cliente pouco especializado. Roda aplicações como HTML com formulários. Aplicações pesadas requerem aplicações específicas. Uma vantagem dos clientes leves em relação ao cliente pesado é a independência de plataforma de acesso. Qualquer navegador em qualquer sistema operacional roda um cliente leve.
  - c) Camada de aplicação ou camada intermediária.
  - d) Camada do Banco de dados.
- 

### 2) Cite algumas vantagens de armazenar dados usando um SGBD ao invés de arquivos comuns.

- Independência do programa que acessa os dados;
  - Possibilidade de descrever os dados desejados sem especificar os algoritmos para acessá-los;
  - Gerenciar relacionamentos complexos;
  - Acesso concorrente a dados compartilhados: transações;
  - Escalabilidade: Grandes massas de dados são armazenadas e acessadas de forma eficiente;
  - Segurança;
  - Tolerância a falhas;
  - Interfaces de administração;
  - Sistema integrado pronto para o uso;
- 

### 3) A linguagem SQL é constituída de duas partes. Dê o nome dessas partes e descreva sucintamente suas características.

- *Linguagem de Definição de Dados, que permite:*
    - Criar e remover bancos de dados;
    - Criar e remover tabelas (alterar o esquema de um BD);
    - Criar e remover índices;
  - *Linguagem de Manipulação de Dados:*
    - Permite realizar buscas, consultar e alterar tabelas;
    - É o conjunto de comandos normalmente empregado por uma aplicação no uso normal de um banco de dados;
-

4) No banco de dados de clientes e contas usado em aula, que comandos SQL poderiam ser usados para:

- a) Inserir um cliente com nome "Flávio de Sousa", residente à "Rua Fonte Nova, 19", e com uma conta cujo número é 9 e cujo saldo é 1000 reais.
- b) Acrescentar uma coluna chamada telefone com capacidade para 12 dígitos na tabela cliente.
- c) Alterar o telefone de "Carlos Pereira" para 2123456789.

a) INSERT into cliente (nome, endereco) values ('Flávio de Sousa', 'Rua Fonte Nova, 19');

INSERT into conta (numero, saldo) values ('9', '1.000');

b) ALTER TABLE `cliente` ADD `telefone` VARCHAR( 12 ) NOT NULL;

c) UPDATE `cliente` SET `telefone` = '2123456789' WHERE `nome` = 'Carlos Pereira' LIMIT 1 ;

---

As questões a seguir consideram as relações pessoa e parente criadas num banco de dados com os comandos abaixo:

```
create table pessoa (  
id int not null auto_increment,  
nome varchar(80) not null,  
sexo char(1) not null,  
primary key (id) );
```

```
create table parente (  
pai int not null,  
filho int not null,  
primary key (pai,filho) );
```

Essas tabelas modelam pessoas e relações de parentesco entre elas. Assim, por exemplo, se existe uma pessoa chamada "Pedro da Silva" cujos pais são "Joao da Silva" e "Maria da Silva", então, eles são representados pelas seguintes linhas nas tabela pessoa e parente:

pessoa		
id	nome	sexo
1	Pedro da Silva	M
2	Joao da Silva	M
3	Maria da Silva	F

parente		
pai	filho	
2	1	
3	1	

Escreva consultas em SQL para obter as seguintes informações:

5. Número de filhos de "Joao da Silva".

```
SELECT count( * )  
FROM parente pa, pessoa pe  
WHERE pa.pai = pe.id  
AND pe.nome = 'Joao da Silva';
```

**6. Número de filhos de "Joao da Silva" com "Maria da Silva".**

```
SELECT count(*)
FROM parente pa1, parente pa2, pessoa pe1, pessoa pe2,
WHERE pa1.filho = pa2.filho
and pa2.pai = pe2.id
and pe2.nome= 'Joao da Silva'
and pa1.pai = pe1.id
and pe1.nome = 'Maria da Silva';
```

**7. Nomes dos pais de "Pedro da Silva".**

```
SELECT pe2.nome
FROM pessoa pe1, pessoa pe2, parente pa
WHERE pe1.nome = 'Pedro da Silva' AND pe1.id = pa.filho AND pe2.id = pa.pai;
```

**8. Nome do pai de "Pedro da Silva".**

```
SELECT pe2.nome
FROM pessoa pe1, pessoa pe2, parente pa
WHERE pe1.nome = 'Pedro da Silva'
AND pe1.id = pa.filho AND pe2.sexo = 'M' AND pe2.id = pa.pai;
```

**9. Nome da mãe de "Pedro da Silva".**

```
SELECT pe2.nome
FROM pessoa pe1, pessoa pe2, parente pa
WHERE pe1.nome = 'Pedro da Silva' AND pe1.id = pa.filho AND pe2.sexo = 'F' AND
pe2.id = pa.pai;
```

**10. Nome dos avós de "Pedro da Silva".**

```
SELECT pe2.nome
FROM pessoa pe1, pessoa pe2, parente pa1, parente pa2
WHERE pe1.nome = 'Pedro da Silva' AND pe1.id = pa1.filho AND pa2.filho = pa1.pai
AND pe2.id = pa2.pai;
```

**11. Nomes das filhas que "Maria da Silva" teve com outra pessoa que não "Joao da Silva".**

```
SELECT pe2.nome
FROM parente pa1, parente pa2, pessoa pe1, pessoa pe2, pessoa pe3
WHERE pe1.nome = 'Maria da Silva'
AND pe1.id = pa1.pai
AND pe2.id = pa1.filho
AND pe2.sexo = 'F'
AND pe3.id = pa2.pai
AND pa2.filho = pa1.filho
AND pe3.sexo = 'M'
AND pe3.nome != 'Joao da Silva';
```