

Disciplina: Programação II
1º Semestre de 2016
Gabarito da AD1

Examine as afirmativas a seguir assinalando V para verdadeiro ou F para falso entre os respectivos parênteses. Apresente uma justificativa de sua resposta. Observações:

- não serão aceitas respostas sem justificativa;
- afirmações verdadeiras podem ser justificadas com exemplos que caracterizem o caso geral da afirmação enquanto que as falsas podem ser justificadas com um contra-exemplo ou pela correção da afirmativa.

Questão 1 (V)

É possível utilizar um nome para uma variável local que já tenha sido usado no mesmo script em uma variável global. Neste caso, as duas variáveis são associadas a duas memórias distintas.

Ex: Sim, há essa possibilidade. É o que ocorre com variáveis globais cujos nomes coincidem com variáveis locais em funções:

```
<?php
$i = 10;
function f() {
    $i = 9;
}
f();
echo $i;
?>
```

\$i continua a valer 10 no escopo global, independente de f(), pois f() tem seu próprio escopo.

Questão 2 (V)

A conversão entre tipos numéricos pode alterar o resultado esperado de uma operação aritmética.

Ex: Isso é esperado e fica claro, por exemplo pelo truncamento de um ponto flutuante convertido para inteiro:

```
<?php
$a = 2.3;
echo (int) ($a/2);
?>
```

Questão 3 (F)

O uso de break no interior de um laço de repetição pode ser substituído pelo comando die, obtendo-se o mesmo resultado.

R: O comando die() interrompe totalmente a execução do script enquanto o comando break; apenas interrompe o laço atual, permitindo que o script prossiga após o laço.

Questão 4 (V)

Utilizando o recurso conhecido como encapsulamento oferecido na orientação a objetos de PHP, é possível modelar uma classe sobre a qual não se deseja revelar como ela funciona internamente, apenas como seus objetos transmitem os seus atributos.

R: Sim, este é o principal propósito do encapsulamento. Por exemplo, podemos ter uma função que encapsula a informação de endereço com o método `getEnderecoCompleto()`, sem saber exatamente como a informação é gerenciada internamente na classe:

```
<?php
class Endereco {

    private $_rua;
    private $_numero;
    private $_bairro;
    private $_cidade;
    private $_pais;

    function __construct($r = 'r. Dr. Azevedo', $n = 10,
                        $b = 'Vila Autódromo', $k = 'Rio de
Janeiro',
                        $p = 'Brasil') {

        $this->_rua = $r;
        $this->_numero = $n;
        $this->_bairro = $b;
        $this->_cidade = $k;
        $this->_pais = $p;
    }

    function getEnderecoCompleto() {
        return $this->_rua . ',' . $this->_numero
            . ', ' . $this->_bairro
            . '/' . $this->_cidade . ' - ' . $this->_pais;
    }

}

$endereco = new Endereco();
echo $endereco->getEnderecoCompleto();
?>
```

Questão 5 (F)

O uso do comando break em desvios condicionais aninhados faz com que as condições seguintes ao break não sejam avaliadas. Assim, no código a seguir por exemplo é preciso utilizar o comando break para que ao encontrar uma nota maior ou igual a 6, o script não teste as demais condições.

```
if ($nota >= 6.0) echo  
    "passou" break;  
elseif ($nota >= 4.0) echo  
    "AP3" break;  
else  
    echo "reprovado";
```

R: Não é preciso utilizar o break no código acima, pois a condição if – else if – else é mutuamente exclusiva. Avalia-se apenas o código dentro da primeira condição verdadeira, ou em último caso, avalia-se o código no else. O comando break ao final de cada condicional é completamente dispensável.

Questão 6 (F)

Formulários são inseridos em páginas web com o objetivo principal de formatar o layout (apresentação) da página.

R: Não, formulários servem como envelopes de dados, que podem ser preenchidos e submetidos ao servidor. O principal uso de formulários nada tem a ver com formatação mas sim com comunicação cliente-servidor.

Questão 7 (V)

Toda função que recebe os parâmetros \$a e \$b usando passagem por cópia, pode ser reescrita utilizando passagem por referência dos mesmos parâmetros, sem alterar sua funcionalidade original. Enquanto que a afirmação contrária não é verdadeira.

R: Verdade. Uma função que já receba uma cópia já não estará contando com a referência dos parâmetros para suas operações. A mudança da passagem por cópia para passagem por referência não alterará em nada seu funcionamento. Já uma função que receba parâmetros por referência pode porventura requerer que a referência seja modificada no escopo que chamou a função, e portanto, uma cópia da informação na variável apenas não será suficiente para que mantenha seu funcionamento.

Questão 8 (F)

Todas as variações de índices utilizados no código a seguir são válidos em PHP para acessar o vetor criado como \$cardapio = array ("arroz", "feijao", "bife", "salada"):

```
$cardapio[0]  
$cardapio["Bife"]  
$cardapio[sizeof($cardapio)]
```

R: A segunda linha de código retornará índice não definido pois o array foi definido apenas como array simples e não como um mapeamento chave => valor. No terceiro caso o erro é ainda pior pois a posição em sizeof de um array sempre será uma posição inválida, pois o tamanho de um array será uma posição posterior à última (índices iniciam no zero).

Questão 9 (F)

Segundo o modelo conceitual de aplicações web, é correto afirmar que os navegadores classificados como clientes leves são os que possuem capacidade de funcionamento em hardwares de dispositivos móveis tais como celulares e tablets.

R: O cliente magro ou leve, ou terminal burro, é um conceito de computador em uma rede cliente-servidor que possui o mínimo possível de recursos e aplicativos instalados localmente, deixando o processamento de informação todo a cargo do servidor. Normalmente possuem apenas uma tela de terminal, necessária para abrir um navegador que serve de janela para uma máquina virtual no servidor, ou se utilizam de conexões do tipo área de trabalho remoto.

Questão 10 (V)

É possível criar um laço de repetição infinito usando o comando continue em seu interior.

R: Criar um laço de repetição infinito não tem relação com o uso do comando continue. Continue apenas passa o laço para a próxima iteração, em que o critério de parada será testado, permitindo que ele seja interrompido ou continue. O que define se o loop é infinito é a ausência de um critério de parada atingível. Mas é possível sim formular um exemplo em que o critério de parada não pode ser atingido em função do uso da cláusula continue. Ex:

```
<?php
$i = 0;
while($i < 10) {
    continue;
    $i++;
}
?>
```

Que resultará no cenário em que \$i++ nunca será chamado, tornando o critério de parada inatingível. Por isso é de suma importância que o critério de parada e o incremento do laço estejam sempre que possível na cláusula where/for. Uma forma de reescrever o código acima evitando este cenário seria:

```
<?php
$i = 0;
while($i++ < 10) {
    continue;
}
?>
```

Questão 11 (F)

É possível desenvolver uma página web utilizando apenas HTML que simule uma calculadora simples que recebe como entrada dois valores e realiza uma das quatro operações aritméticas básicas (soma, subtração, multiplicação ou divisão) e em seguida apresente o resultado na tela.

R: Pelo menos seria necessário também o uso de Javascript. HTML é linguagem de marcação, definindo apenas a estrutura da página.

Questão 12 (F)

Devemos usar aspas simples em strings (cadeias de caracteres) que precisam ser avaliadas dinamicamente por nosso script PHP. Por outro lado, apenas as aspas duplas podem ser utilizadas em cadeias de caracteres a serem exibidas exatamente como foram escritas no código, ou seja, sem serem avaliadas.

R: O funcionamento é exatamente o inverso.