**Avaliação**

Parte superior do formulário

**1) Sobre o PHP, é correto afirmar:  
\* Marque todas as alternativas que respondem o enunciado da questão**

|  |  |
| --- | --- |
|  | Sua licença de uso é gratuito, mas não para fins comerciais. **(alternativa incorreta)** |
|  | É uma linguagem compilada. **(alternativa incorreta)** |
|  | O código-fonte fica disponível para o público que quiser visualizar, mas não executar. **(alternativa incorreta)** |
|  | Os arquivos PHP são enviados para o navegador do usuário, que por sua vez os interpreta. **(alternativa incorreta)** |
|  | Utiliza a memória RAM do servidor para realizar suas operações. **(alternativa correta)** |
|  | Pode ser utilizado com um servidor HTTP. **(alternativa correta)** |

http://www.softblue.com.br/public/images/sbv2_empty.png

|  |  |
| --- | --- |
| http://www.softblue.com.br/public/images/sbv2_TestCorrect.png | **Resposta correta!** O PHP é uma linguagem interpretada, e não compilada, além de ser de uso gratuito inclusive para fins comerciais. Seus códigos-fontes podem ser visualizados pelos administradores do servidor de hospedagem apenas, mas não pelo público do seu site. É possível acoplar o PHP a um servidor HTTP para utilizá-lo. Como o PHP é executado no servidor, e não no navegador do usuário, são os recursos do servidor que são utilizados, como memória RAM e processador. |

**2) Suponha que em seu projeto PHP exista um arquivo chamado counter.php que possui um código que toda vez que é executado incrementa o valor 1 em um arquivo de configuração específico, e que caso o arquivo de configuração não esteja disponível, o código seja ignorado. Qual o melhor comando para incluir este script em suas páginas PHP?**

|  |  |
| --- | --- |
|  | include **(alternativa correta)** |
|  | include\_once **(alternativa incorreta)** |
|  | require **(alternativa incorreta)** |
|  | require\_once **(alternativa incorreta)** |
|  | Qualquer uma dessas alternativas **(alternativa incorreta)** |

http://www.softblue.com.br/public/images/sbv2_empty.png

|  |  |
| --- | --- |
| http://www.softblue.com.br/public/images/sbv2_TestWrong.png | **Resposta incorreta!** Neste caso, como o arquivo pode ou não estar presente no servidor, o comando require não deve ser utilizado, pois não houveram requisitos informando que a aplicação devesse ser interrompida na falta do arquivo, mas sim que é um código que pode ser ignorado caso falhe. Como o código em questão executa operações, e não é uma função, mas sim um script PHP, o código em questão deve ser invocado por meio do comando include. |

**3) Qual será o resultado impresso pela execução do código-fonte apresentado a seguir?**

|  |  |  |
| --- | --- | --- |
| 1 | for($x=0; $x<5; $x++) { | |
| 2 | if($x == 3) { |

|  |  |  |
| --- | --- | --- |
| 3 | echo "a"; | |
| 4 | } |

|  |  |
| --- | --- |
| 5 | else { |
| 6 | echo "b"; | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | | } | | |
| 8 | | } | |
|  | | bbbab **(alternativa correta)** | | |
|  | | babbb **(alternativa incorreta)** | | |
|  | | bbabb **(alternativa incorreta)** | | |
|  | | Nenhuma das alternativas **(alternativa incorreta)** | | |
|  | | bbbba **(alternativa incorreta)** | | |

http://www.softblue.com.br/public/images/sbv2_empty.png

|  |  |
| --- | --- |
| http://www.softblue.com.br/public/images/sbv2_TestCorrect.png | **Resposta correta!** Para os valores de 0, 1 e 2 de $x, o valor b é impresso na tela, seguido de uma impressão de a e uma impressão de b antes do for finalizar. |

**4) Qual será o resultado impresso pela execução do código-fonte apresentado a seguir?**

|  |  |
| --- | --- |
| 1 | $x = 3; |
| 2 | if($x++ == 6 || $x++ == 5 || $x++ == 4) { | |

|  |  |  |
| --- | --- | --- |
| 3 | $x = $x; | |
| 4 | } |

|  |  |
| --- | --- |
| 5 | else { |
| 6 | $x = $x + $x++; | |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | | } | |
| 8 | | echo $x; | | |
|  | | 6 **(alternativa incorreta)** | | |
|  | | 7 **(alternativa incorreta)** | | |
|  | | 3 **(alternativa incorreta)** | | |
|  | | 5 **(alternativa incorreta)** | | |
|  | | 4 **(alternativa incorreta)** | | |
|  | | Nenhuma das alternativas **(alternativa correta)** | | |

http://www.softblue.com.br/public/images/sbv2_empty.png

|  |  |
| --- | --- |
| http://www.softblue.com.br/public/images/sbv2_TestCorrect.png | **Resposta correta!** Nenhuma das alternativas está correta pois é o valor 13 que é impresso na tela. Quando o operador ++ está ao lado direito da variável, a mesma só e atualizada após a expressão em questão ser avaliada. Portanto, é comparado se 3 == 6, 4 == 5 e por último 5 == 4, fazendo com que o código acesse o bloco else e incremente $x (6) com $x (totalizando em 12), e por último realizando um incremento ++ em 12, resultando em 13. |

**5) Qual será o resultado impresso pela execução do código-fonte apresentado a seguir?**

|  |  |
| --- | --- |
| 1 | $x = 3; |
| 2 | if($x++ > 0 && $x++ < 5) { | |

|  |  |  |
| --- | --- | --- |
| 3 | $x = $x + 1; | |
| 4 | } |

|  |  |
| --- | --- |
| 5 | else { |
| 6 | $x = $x - 1; | |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | | } | |
| 8 | | echo $x; | | |
|  | | 3 **(alternativa incorreta)** | | |
|  | | Nenhuma das alternativas **(alternativa incorreta)** | | |
|  | | 4 **(alternativa incorreta)** | | |
|  | | 7 **(alternativa incorreta)** | | |
|  | | 5 **(alternativa incorreta)** | | |
|  | | 6 **(alternativa correta)** | | |

http://www.softblue.com.br/public/images/sbv2_empty.png

|  |  |
| --- | --- |
| http://www.softblue.com.br/public/images/sbv2_TestCorrect.png | **Resposta correta!** Neste comando if o operador && (otimizado) foi utilizado, mas para uma expressão do tipo E, é necessário confirmar todas as sentenças, enquanto nenhuma sentença falsa apareça para interromper a avaliação das demais expressões lógicas. Portanto, $x é atualizado duas vezes na avaliação da condição do comando if, e mais uma no bloco de aceite, atingindo o valor 6. Observe que os operadores ++ atualizam $x somente após suas comparações serem realizadas. |

**6) Sobre lógica de programação utilizada pelo PHP, é correto afirmar:  
\* Marque todas as alternativas que respondem o enunciado da questão**

|  |  |
| --- | --- |
|  | $y = $x++; e $y = ++$x; são duas formas diferentes de se obter o mesmo resultado. **(alternativa incorreta)** |
|  | Os sinais de = e == comparam se dois objetos são iguais, e se dois objetos são idênticos, respectivamente. **(alternativa incorreta)** |
|  | Os operadores & e && não possuem a mesma funcionalidade. **(alternativa correta)** |
|  | ($x++ == 1 && $y) e ($y && $x++ == 1) são duas formas diferentes de se obter o mesmo resultado. **(alternativa incorreta)** |
|  | $x > $y e $y < $x são duas formas de se escrever a mesma condição. **(alternativa correta)** |

http://www.softblue.com.br/public/images/sbv2_empty.png

|  |  |
| --- | --- |
| http://www.softblue.com.br/public/images/sbv2_TestWrong.png | **Resposta incorreta!** Os comandos $x++ e ++$x executam a mesma função de incrementar o valor de $x, mas quando utilizados em uma expressão matemática podem oferecer resultados diferentes, uma vez que $x++ utiliza o valor atual de $x na expressão, e depois incrementa-o, enquanto ++$x primeiro incrementa $x e depois utiliza o valor atualizado na expressão, resultando em diferentes valores para $y, apesar de $x no final do código apresentar o mesmo valor. <BR>Os sinais de = e == são utilizados para atribuir valor em uma variável e comparar valores, respectivamente. <BR> O operador && é otimizado para não avaliar a condição inteira caso ele encontre um parâmetro falso, enquanto que & avalia a condição inteira, mesmo que já conclua que não será verdadeira baseado em algum parâmetro falso encontrado. Portanto, além de terem funcionalidades diferentes, podem afetar o resultado se na expressão condicional houver alguma operação em algum dos lados, como é o caso da expressão ($y & $x++ == 1). |

**7) Sobre a linguagem PHP, é correto afirmar:  
\* Marque todas as alternativas que respondem o enunciado da questão**

|  |  |
| --- | --- |
|  | O comando switch analisa uma condição lógica, executando seu bloco de código se esta condição for verdadeira, ou ignorando-a se for falsa. **(alternativa incorreta)** |
|  | for($x=1; $x<=9; $x++) { /\*bloco\*/ } é o mesmo que $x = 0; while($x++ < 9) { /\*bloco\*/ } **(alternativa correta)** |
|  | Um comando if pode apresentar quantos blocos else for necessário, mas somente no máximo um bloco else if. **(alternativa incorreta)** |
|  | $x = 1; while(++$x < 10) { /\*bloco\*/ } é o mesmo que do { /\*bloco\*/ } while(++$x < 10); **(alternativa incorreta)** |
|  | Todo comando if deve possuir ao menos um bloco else ou else if. **(alternativa incorreta)** |
|  | if($x == 2) { /\*1\*/ } else if ($x == 3) { /\*2\*/ } else { /\*3\*/ } possui a mesma estrutura lógica que switch($x) { case 2: /\*1\*/ break; case 3: /\*2\*/ break; default: /\*3\*/ break; } **(alternativa correta)** |

http://www.softblue.com.br/public/images/sbv2_empty.png

|  |  |
| --- | --- |
| http://www.softblue.com.br/public/images/sbv2_TestWrong.png | **Resposta incorreta!** Um comando if é composto pelo bloco de código do if, zero ou mais blocos else-if, e zero ou um (no máximo) bloco else final. Um comando switch não analisa uma condição lógica, mas sim, o valor de uma variável, e baseado em seu valor, encaminha para o bloco de execução adequado. <BR>Dependendo das condições do comando if, é possível sim construir um comando switch que tenha o mesmo resultado do comando if, assim como um comando for pode ser escrito de outras formas por meio do comando while, por exemplo. |

**8) Sobre o código a seguir, é correto afirmar:  
\* Marque todas as alternativas que respondem o enunciado da questão**

|  |  |  |
| --- | --- | --- |
| 1 | 01: for($x=0; $x<10; $x++) { | |
| 2 | 02:     $y = $x; |

|  |  |  |
| --- | --- | --- |
| 3 | 03:     if($x == 9) { | |
| 4 | 04: |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | | 05:     } | | |
| 6 | | 06: } | |
|  | | Os comandos break e exit possuem a mesma funcionalidade de encerramento. **(alternativa incorreta)** | | |
|  | | Ao utilizar break ou continue na linha 4, o valor de $y será o mesmo após este script ser executado, e o código continuará executando após o bloco do comando for. **(alternativa correta)** | | |
|  | | É possível utilizar os comandos break e continue em um comando foreach. **(alternativa correta)** | | |
|  | | Ao utilizar break ou continue na linha 4, o valor de $x será o mesmo após este script ser executado, e o código continuará executando após o bloco do comando for. **(alternativa incorreta)** | | |
|  | | Ao utilizar exit na linha 4, o valor de uma das variáveis continuará o mesmo após este script ser executado , e o código continuará executando após o bloco do comando for. **(alternativa incorreta)** | | |

http://www.softblue.com.br/public/images/sbv2_empty.png

|  |  |
| --- | --- |
| http://www.softblue.com.br/public/images/sbv2_TestCorrect.png | **Resposta correta!** Este código apresenta o mesmo valor para $y se um comando break ou continue forem incluídos na linha 4. Contudo, o valor de $x varia dependendo do comando, pois o comando break interrompe toda a execução do comando for, enquanto o comando continue atualiza a variável $x uma última vez ainda ao tentar continuar com a próxima iteração do comando, mas não entra no bloco de código pois seus valores já não são mais satisfeitos. <BR>Ao utilizar o comando exit na linha 4, o código é interrompido totalmente, e não continua executando após o comando for, pois o comando exit interrompe a execução do arquivo PHP como um todo, e não apenas algum comando. |

Parte inferior do formulário