



## CONCURSO PÚBLICO 2015

### CARGO

### ANALISTA DE TECNOLOGIA DA INFORMAÇÃO DESENVOLVIMENTO DE SISTEMAS

#### INSTRUÇÕES AOS CANDIDATOS

- Verifique se este caderno contém 25 questões. Caso não contenha, solicite imediatamente ao fiscal de sala outro caderno.
- Para se dirigir ao fiscal, erga o braço.
- Deixe à vista seu documento de identidade.
- Deixe cadernos, livros, bolsas e outros objetos ao lado ou abaixo da cadeira. Quem estiver com telefone celular deverá retirar a bateria, ou nesta impossibilidade, desligá-lo. Não é permitido o uso de qualquer tipo de aparelho eletrônico dentro do prédio de provas.
- Durante a realização da prova não será permitido qualquer tipo de comunicação entre os candidatos e não poderão fazer uso de livros, manuais, impressos, anotações, máquinas calculadoras, aparelho auricular, óculos com lentes escuras.
- É proibido fumar no interior do prédio de provas.
- Poderá utilizar a grade ao final do caderno para marcar previamente as respostas.
- Para cada questão existe apenas uma resposta certa.
- O cartão resposta, se danificado pelo candidato não será substituído.
- Em hipótese alguma deve ser colocado nome, assinatura ou outro tipo de identificação fora dos espaços reservados para tal no cartão resposta.
- Transcreva as respostas para o cartão resposta, preenchendo totalmente o círculo com caneta esferográfica com tinta preta ou azul escuro, não sendo permitido o uso de caneta porosa ou corretivo líquido.
- A entrega da prova e ida ao sanitário só poderá ocorrer depois de transcorrida uma hora do início da prova.
- Ao terminar a prova, deverá ser entregue, obrigatoriamente, ao fiscal de sala, seu cartão resposta devidamente assinado, podendo levar consigo o caderno de questões.
- Após a entrega da prova, o candidato deverá retirar-se imediatamente do prédio de aplicação da prova, não sendo permitido, nesse local, o uso dos sanitários e de qualquer aparelho eletrônico.
- A prova terá duração de até 2 horas.
- Será excluído do concurso o candidato que agir com incorreção ou descortesia com qualquer pessoa da equipe encarregada da aplicação das provas, comissão central ou candidato participante do processo.
- Os 2 (dois) últimos candidatos que permanecerem em sala de prova, só poderão retirar-se conjuntamente e após sua assinatura na ata de presença.
- O gabarito será divulgado em até 48 horas após o término da prova, no site [www.progep.furg.br](http://www.progep.furg.br).

Conforme consta no edital, as ferramentas e versões que devem ser consideradas para a solução das questões são: PHP 5.3, Javascript 1.8, HTML 5, CSS 3, Google Chrome 40 ou superior, SQL 2008, PostgreSQL 8.4.

As questões 1, 2, 3, 4, 5, 6, 7 e 8 devem levar em consideração um banco de dados constituído por tabelas cargo e servidor, implementado em PostgreSQL, com as respectivas definições:

```
CREATE TABLE cargo (
    cd_cargo integer NOT NULL,
    ds_cargo character varying(30),
    CONSTRAINT cargo_pkey PRIMARY KEY (cd_cargo)
);
CREATE TABLE servidor (
    nr_matricula integer NOT NULL,
    cd_cargo integer,
    nm_servidor character varying(40),
    cd_sexo character(1),
    CONSTRAINT servidor_pkey PRIMARY KEY (nr_matricula),
    CONSTRAINT servidor_cd_cargo_fkey FOREIGN KEY (cd_cargo)
    REFERENCES cargo (cd_cargo) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE RESTRICT
);
```

As tabelas contêm as seguintes informações:

Cargo	
cd_cargo	ds_cargo
15	Analista de TI
25	Técnico em TI
35	Administrador
45	Técnico Administrativo

Servidor			
nr matricula	cd cargo	nm servidor	cd sexo
401	15	Nilton	M
402	45	Maria	F
403	45	Roberto	M
404	35	João	M
405	15	Eduardo	M
406	null	Júlio	M
407	25	Thais	F
408	null	Marcelo	M

Implementadas pelas classes "Cargo" e "Servidor", nos arquivos "class.Cargo.php" e "class.Servidor.php", respectivamente.

#### class.Cargo.php

```
<?php
class Cargo {
    private $dados = array();
    private $db = null;
    private $cd_cargo = null;
    public function defineDB($db){ $this->db = $db; }
    public function define($cd){
        $r = @pg_query ($this->db,'SELECT * FROM cargo WHERE cd_cargo='.$cd.'');
        if (!pg_num_rows($r)) throw new Exception('Erro Define Cargo');
        $this->dados = pg_fetch_assoc($r);
        $this->cd_cargo = $this->dados['cd_cargo'];
    }
    private function obrigatorios() {
        if (empty($this->dados['cd_cargo']) or empty($this->dados['ds_cargo']))
            throw new Exception('Erro Obrigatório Cargo');
    }
    public function inserir() {
        if (empty($this->db)) throw new Exception('Erro DB Cargo');
        $this->obrigatorios();
        $r = @pg_query ($this->db,"insert into cargo (cd_cargo,ds_cargo)
            values ({ $this->dados['cd_cargo']}, '{ $this->dados['ds_cargo'] }')
            returning *;");
        if ($r) $this->dados = pg_fetch_assoc($r);
        else throw new Exception('Erro Inserir Cargo');
    }
}
```

```

public function alterar(){
    if (empty($this->db)) throw new Exception('Erro DB Cargo');
    $this->obrigatorios();
    $r = @pg_query ($this->db,"update cargo set
        cd_cargo={$this->dados['cd_cargo']},
        ds_cargo='{$this->dados['ds_cargo']}'
        where cd_cargo='{$this->cd_cargo}' returning *;");
    if ($r) $this->dados = pg_fetch_assoc($r);
    else throw new Exception('Erro Alterar Cargo');
}
public function remover(){
    if (empty($this->db)) throw new Exception('Erro DB Cargo');
    $r = @pg_query ($this->db,"delete from cargo
        where cd_cargo='{$this->cd_cargo}';");
    if (!$r) throw new Exception('Erro Remover Cargo');
}
public function obterDados() { return $this->dados; }
public function defineCdCargo ($v) { $this->dados['cd_cargo'] = $v; }
public function defineDsCargo ($v) { $this->dados['ds_cargo'] = $v; }
}
?>

```

#### class.Servidor.php

```

<?php
class Servidor {
    private $dados = array();
    private $db = null;
    private $nr_matricula = null;
    public function defineDB($db){ $this->db = $db; }
    public function define($nr){
        $r = @pg_query ($this->db,"SELECT * FROM servidor WHERE
        nr_matricula='{$nr}';");
        if (!pg_num_rows($r)) throw new Exception('Erro Define Servidor');
        $this->dados = pg_fetch_assoc($r);
        $this->nr_matricula = $this->dados['nr_matricula'];
    }
    private function obrigatorios () {
        if (empty($this->dados['nr_matricula']) or
            empty($this->dados['cd_cargo']) or
            empty($this->dados['nm_servidor']) or
            empty($this->dados['cd_sexo']) )
            throw new Exception('Erro Obrigatório Servidor');
    }
    public function inserir(){
        if (empty($this->db)) throw new Exception('Erro DB Servidor');
        $this->obrigatorios();
        $r = @pg_query ($this->db,"insert into servidor
        (nr_matricula,cd_cargo,nm_servidor,cd_sexo)
        values ({ $this->dados['nr_matricula']},
        '{ $this->dados['cd_cargo']}',
        '{ $this->dados['nm_servidor']}',
        '{ $this->dados['cd_sexo']}' ) returning *;");
        if ($r) $this->dados = pg_fetch_assoc($r);
        else throw new Exception('Erro Inserir Servidor');
    }
    public function alterar(){
        if (empty($this->db)) throw new Exception('Erro DB Servidor');
        $this->obrigatorios();
        $r = @pg_query ($this->db,"update servidor set
        nr_matricula={$this->dados['nr_matricula']},
        cd_cargo={$this->dados['cd_cargo']},
        nm_servidor='{$this->dados['nm_servidor']}',
        cd_sexo='{$this->dados['cd_sexo']}'
        where nr_matricula='{$this->nr_matricula}' returning *;");
        if ($r) $this->dados = pg_fetch_assoc($r);
    }
}

```

```

        else throw new Exception('Erro Alterar Servidor');
    }
    public function remover(){
        if (empty($this->db)) throw new Exception('Erro DB Cargo');
        $r = @pg_query ($this->db,"delete from servidor
        where nr_matricula='{ $this->nr_matricula}';");
        if (!$r) throw new Exception('Erro Remover Servidor');
    }
    public function obterDados(){ return $this->dados; }
    public function obterNmServidor() { return $this->dados['nm_servidor']; }
    public function defineNrMatricula($v) { $this->dados['nr_matricula']=$v; }
    public function defineCdCargo($v) { $this->dados['cd_cargo']=$v; }
    public function defineNmServidor($v) { $this->dados['nm_servidor']=$v; }
    public function defineCdSexo($v) { $this->dados['cd_sexo']=$v; }
    }
    ?>

```

- 1) Assinale a alternativa que apresenta apenas a(s) afirmativa(s) correta(s), de acordo com trecho de código PHP abaixo:

```

<?php
require ('class.Cargo.php');
require ('class.Servidor.php');
$db = pg_connect("host=localhost port=5432 dbname=furg");
try {
    $s = new Servidor();
    $s->defineDB($db);
    $s->define(402);
    $s->defineNrMatricula(485);
    $s->defineNmServidor('Adalberto');
    $d = $s->obterDados();
}
catch (Exception $e) { echo $e->getMessage(); }
?>

```

- I - O campo nm\_servidor do registro de chave primária 402 da tabela servidor terá como o valor a *string* Adalberto.  
 II - A variável \$d é um *array* associativo cujo índice nm\_servidor tem como valor a *string* Adalberto.  
 III - A classe Servidor não permite que a chave primária da tabela servidor seja modificada, assim será levantada uma exceção e nenhuma modificação será realizada.

- a) Apenas I  
 b) Apenas II  
 c) Apenas III  
 d) Apenas I e II  
 e) Apenas II e III

- 2) Analise o trecho de código PHP abaixo e assinale a alternativa que apresenta o número de registros da tabela servidor que serão modificados:

```

<?php
require ('class.Cargo.php');
require ('class.Servidor.php');
$db1 = pg_connect ("host=localhost port=5432 dbname=furg");
$db2 = pg_connect ("host=localhost port=5432 dbname=furg");
try {
    $s = new Servidor();
    $s->defineDB($db2);
    pg_query ($db2,'begin');
    $sql ="select s.nr_matricula from servidor s
    left join cargo c on c.cd_cargo = s.cd_cargo
    order by s.nr_matricula";
    $r = pg_query ($db1, $sql);
    while ($serv = pg_fetch_assoc($r)) {

```

```

        $s->define($serv['nr_matricula']);
        $d = $s->obtemDados();
        $s->defineNmServidor($d['nm_servidor'].' da Silva');
        $s->alterar();
    }
    pg_query ($db2,'commit');
}
catch (Exception $e) { pg_query ($db2,'rollback'); }
?>

```

- a) 0
- b) 2
- c) 4
- d) 6
- e) 8

3) Analise o trecho de código PHP abaixo e assinale a alternativa que apresenta o número de registros da tabela servidor que serão modificados:

```

<?php
require ('class.Cargo.php');
require ('class.Servidor.php');
$db1 = pg_connect ("host=localhost port=5432 dbname=furg");
$db2 = pg_connect ("host=localhost port=5432 dbname=furg");
try {
    $s = new Servidor();
    $s->defineDB($db2);
    $sql = "select nr_matricula from servidor
    where cd_cargo in (
        select s.cd_cargo from servidor s
        inner join cargo c on c.cd_cargo=s.cd_cargo
        group by s.cd_cargo having count(*)>1
    )
    order by nr_matricula desc ";
    $r = pg_query ($db1,$sql);
    while ($serv = pg_fetch_assoc($r)) {
        $s->define($serv['nr_matricula']);
        $d = $s->obtemDados();
        $nr = $d['nr_matricula'] + 4 ;
        $s->defineNrMatricula($nr);
        $s->alterar();
    }
}
catch (Exception $e) {}
?>

```

- a) 0
- b) 1
- c) 2
- d) 4
- e) 6

4) Analise o trecho de código PHP abaixo e assinale a alternativa que apresenta o texto que será impresso na tela como saída do programa:

```

<?php
require ('class.Cargo.php');
require ('class.Servidor.php');
$db1 = pg_connect ("host=localhost port=5432 dbname=furg");
$db2 = pg_connect ("host=localhost port=5432 dbname=furg");
try {
    $sql="select cd_cargo from cargo order by cd_cargo";
    $r = pg_query($db1,$sql);

```

```

$nr = pg_num_rows($r);
while ($carga = pg_fetch_assoc($r)) {
    $nr++;
    $c = new Cargo();
    $c->defineDB($db2);
    $c->define($carga['cd_carga']);
    $c->remover();
}
}
catch (Exception $e) {}
echo $nr;
?>

```

- a) 3
- b) 4
- c) 5
- d) 6
- e) 8

- 5) Analise o trecho de código PHP abaixo e assinale a alternativa que apresenta o texto que será impresso na tela como saída do programa:

```

<?php
require ('class.Cargo.php');
require ('class.Servidor.php');
$db1 = pg_connect ("host=localhost port=5432 dbname=furg");
$db2 = pg_connect ("host=localhost port=5432 dbname=furg");
try {
    $str = '';
    $s = new Servidor();
    $s->defineDB($db2);
    $sql ="select s.nr_matricula from servidor s
left join cargo c on c.cd_carga=s.cd_carga
where s.cd_carga is null order by s.nr_matricula";
    $r = pg_query ($db1,$sql);
    while ($serv = pg_fetch_assoc($r)) {
        $s->define($serv['nr_matricula']);
        $s->remover();
        $d = $s->obtemDados();
        if (empty($d['nm_servidor'])) throw new Exception('ERRO!');
        $str.=(!empty($str)?', ':'').$d['nm_servidor'];
    }
    echo $str;
}
catch (Exception $e) { echo $e->getMessage(); }
?>

```

- a) ERRO!
- b) Nilton, Maria, Roberto, João, Eduardo, Júlio, Thais, Marcelo
- c) Maria, Roberto, Thais
- d) Nilton, João, Eduardo
- e) Júlio, Marcelo

- 6) Uma vez executados os seguintes comandos SQL,

```

insert into cargo (cd_carga,ds_carga) values (65,'SEM CARGO');
update servidor set cd_carga=65 where nr_matricula in (
select s.nr_matricula from servidor s
inner join cargo c on c.cd_carga=s.cd_carga );

```

assinale a alternativa que indica quantos registros da tabela servidor serão modificados:

- a) 0
- b) 2
- c) 3
- d) 6
- e) 8

- 7) A *query* abaixo objetiva buscar o quantitativo de servidores de cada sexo vinculados a cada cargo, mostrando algumas das informações. Identifique o problema de execução:

```
select c.ds_cargo, servidor.cd_sexo, c.cd_cargo, count(cd_sexo)
from cargo c
inner join public.servidor on c.cd_cargo = servidor.cd_cargo
group by c.ds_cargo, servidor.cd_sexo;
```

- a) A coluna `cd_sexo` não tem a tabela especificada.
- b) *Aliases* das tabelas devem ser declarados com o comando `as`.
- c) É necessário utilizar a cláusula `left join` ao invés do `inner join`.
- d) A coluna `c.cd_cargo` deve aparecer na cláusula `group by`.
- e) A tabela `cargos` deve ter um *schema* especificado.

- 8) Qual a saída de texto exibida em tela, pelos comandos `echo`, após a execução do código PHP abaixo?

```
<?php
$s1 = new Servidor();
$s1->defineNmServidor("Servidor01");
if(!empty($s1->obtemNmServidor())) echo "Nome: " . $s1->obtemNmServidor();
$s2 = new Servidor();
$s2->defineNmServidor("");
if(!empty($s2->obtemNmServidor())) echo "Nome: " . $s2->obtemNmServidor();
?>
```

- a) Nome: Servidor01Nome: Servidor01
- b) Nome: Servidor01Nome:
- c) Nome: Servidor01
- d) Nome: Nome:
- e) Nenhuma, pois o código contém um erro fatal.

- 9) Analise o trecho de código abaixo e assinale a alternativa correspondente ao resultado da consulta:

```
CREATE TABLE letras( letra CHAR(1) NOT NULL );
CREATE TABLE vogais( vogal CHAR(1) NOT NULL );
INSERT INTO letras(letra) values('A'),('A'),('A'),('B'),('C');
INSERT INTO vogais(vogal) values('A'),('A'),('B'),('B');
SELECT count(*)
FROM (
    SELECT * FROM letras
    EXCEPT ALL
    SELECT * FROM vogais
) as tabela
```

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

10) São exemplos de DDL, DML e DCL, respectivamente:

- a) INSERT, CREATE, TRUNCATE
- b) SELECT, INSERT, REVOKE
- c) CREATE, UPDATE, COMMIT
- d) TRUNCATE, GRANT, ROLLBACK
- e) ALTER, SELECT, REVOKE

11) O código PHP abaixo tem por objetivo transformar um vetor de porcentagens, com chaves ordenadas de zero a quatro, em um vetor com números reais. O que impede que este algoritmo funcione conforme o desejado?

```
<?php
function funcao(&$valor) {
    $valor = (float) $valor / 100;
    return $valor;
}
$vetor = array(false => 100, '1' => 80, 2 => 60, 40, 20);
array_walk($vetor, 'funcao');
?>
```

- a) A função não deveria ter um comando return, pois já altera os valores do vetor por referência.
- b) A variável \$valor, passada com referência na função, deveria ser a variável \$vetor.
- c) O array está mal construído, não tendo chaves ordenadas de zero a quatro como especificado.
- d) O segundo parâmetro da função array\_walk é passado como string, e não como ponteiro para função.
- e) Nada impede que o algoritmo funcione conforme desejado.

12) Qual a saída de texto quando um arquivo válido de nome relatorio.pdf for selecionado no input de name arquivo do formulário abaixo e o enviado através do botão Enviar?

```
<?php if ($_SERVER['REQUEST_METHOD'] == 'GET') { ?>
<form method="post" action="<?php echo $_SERVER['SCRIPT_NAME'] ?>"
enctype="multipart/form-data">
<input type="file" name="arquivo"/><input type="submit" value="Enviar"/>
</form>
<?php } else {
    error_reporting(0);
    if (isset($_FILES)) {
        $caminho = '/tmp/' . $_FILES['name'];
        if (move_uploaded_file($_FILES['tmp_name'], $caminho))
            print "Arquivo salvo em $caminho.";
        else print "Não foi possível salvar em $caminho.";
    } else { print "Arquivo não é válido."; }
}
?>
```

- a) Não foi possível salvar em /tmp/.
- b) Arquivo não é válido.
- c) Arquivo salvo em /tmp/relatorio.pdf.
- d) A execução do upload apresenta um erro fatal ao não referenciar a tag name do input: arquivo.
- e) Arquivo salvo em /tmp/.

13) Analise o trecho de código PHP abaixo e assinale a alternativa que apresenta o texto que será impresso na tela como saída do programa:

```
<?php
function funcao($v){
    static $x = null;
    if($x !== null) return $x;
    $x += $v;
    return $x;
}
```



```
$v = 3;
$v += funcao($v);
$v = funcao($v) + funcao($v) + funcao($v);
$v += funcao($v) + funcao($v);
echo $v;
?>
```

- a) 3
- b) 6
- c) 9
- d) 12
- e) 15

14) Qual a saída de texto exibida em tela, pelos comandos echo, após a execução do código PHP abaixo?

```
<?php
for($i = 0; $i < 10; $i++){
    if(!($i % 2)) escreve($i);
    else echo $i;
}
function escreve($i){
    $continua = true;
    while($continua){
        switch($i){
            case 2:
                echo $i;
            case 1:
                echo $i;
                $continua = false;
            case 0:
                echo $i;
            default:
                $continua = false;
        }
        $i++;
    }
}
?>
```

- a) 0123456789
- b) 012223579
- c) 0123579
- d) 13579
- e) 02468

15) Sobre os conceitos de Orientação a Objetos, quais das afirmativas abaixo são corretas?

I - Encapsulamento: mecanismo utilizado para impedir o acesso direto aos atributos de um objeto.  
II - Herança: mecanismo no qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos.  
III - Abstração: é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes.

- a) Apenas I
- b) Apenas I e II
- c) Apenas I e III
- d) Apenas II e III
- e) I, II e III

16) Quanto ao método ágil de desenvolvimento *Scrum*, quais afirmativas abaixo são corretas?

- I - Com o objetivo de fornecer transparência, quem solicita o trabalho define o conceito de "terminado", mesmo que não seja a mesma visão de quem realiza o trabalho.  
 II - A inspeção do processo de *Scrum* deve ser frequente, com o objetivo de detectar variações indesejadas ao objetivo do *Sprint*, mas desde não atrapalhe o andamento do trabalho.  
 III - *Sprints* possuem cinco fases: identificação de problemas, planejamento, desenvolvimento, revisão e retrospectiva.

- a) Apenas I
- b) Apenas II
- c) Apenas I e II
- d) Apenas II e III
- e) I, II e III

- 17) Preencha corretamente as lacunas das afirmações abaixo, respectivamente, para que representem funções do *Scrum Master* no processo de *Scrum*:

- I - Remover obstáculos no progresso \_\_\_\_\_.  
 II - Planejar as implementações do *Scrum* \_\_\_\_\_.  
 III - Certificar-se que \_\_\_\_\_ saiba organizar o *Backlog* de Produto para maximizar valor.

- a) da organização - na equipe de desenvolvimento - o dono do produto
- b) da organização - para o dono do produto - a equipe de desenvolvimento
- c) da equipe de desenvolvimento - na organização - o dono do produto
- d) da equipe de desenvolvimento - para o dono do produto - a organização
- e) do dono do produto - na organização - a equipe de desenvolvimento

- 18) São artefatos do *Scrum*:

- a) Planejamento - Revisão - Retrospectiva
- b) Dono do Produto - *Scrum Master* - Equipe de Desenvolvimento
- c) Transparência - Inspeção - Adaptação
- d) *Backlog* de produto - *Backlog* de *Sprint* - Incremento
- e) *Feedback* - Comunicação - Simplicidade

- 19) Em relação ao XP (*eXtreme Programming*), assinale quais das afirmativas abaixo são corretas:

- I - Desenvolvimento guiado por testes, *refactoring*, programação em par e código coletivo são práticas do XP.  
 II - Equipes XP separam desenvolvedores dos detentores da lógica de negócio (clientes).  
 III - O XP não defende o código padronizado, priorizando a agilidade no desenvolvimento.

- a) Apenas I
- b) Apenas II
- c) Apenas I e II
- d) Apenas I e III
- e) Apenas II e III

- 20) Analise o trecho de código JavaScript abaixo e assinale a alternativa que corresponde ao valor apresentado pelo comando `alert`:

```
$x = "Concurso Analista de TI";
$x = $x.toLowerCase();
$x[2] = 'N';
$x.toUpperCase();
alert($x);
```

- a) concurso analista de ti
- b) CONCURSO ANALISTA DE TI
- c) Concurso Analista de TI
- d) coNcurso analista de ti
- e) CONCURSO analista de TI

- 21) Analise o trecho de código JavaScript abaixo e assinale a alternativa que corresponde ao valor de `$x` ao final da execução:

```
$vetor = [2, 3, 4, 5];  
$x = $vetor.reduce(function(a, b, c, d){  
    return a * a;  
});
```

- a) 4
- b) 54
- c) 120
- d) 256
- e) undefined

- 22) Analise o trecho de código JavaScript

```
var cargos = ["Analista", "Professor", "Motorista", "Policial"];  
_____  
alert(cargos.length);
```

e assinale a alternativa que substitui a lacuna de forma que o comando `alert` apresente o valor 3:

- a) `cargos.splice(2, 1);`
- b) `delete cargos[3];`
- c) `unset(cargos[2]);`
- d) `cargos.remove(2);`
- e) `cargos.delete("Motorista");`

- 23) Em relação ao CSS, qual o padrão correto que define o estilo em todos os elementos `h3` precedidos por `p`, como no exemplo abaixo?

- a) `p ^ h3{color:red}`
- b) `p ~ h3{color:red}`
- c) `p + h3{color:red}`
- d) `p, h3{color:red}`
- e) `p > h3{color:red}`

- 24) Em relação ao CSS, assinale quais das afirmativas abaixo são corretas:

- I - A propriedade `margin` aceita valores negativos.
- II - A propriedade `padding` aceita valores negativos.
- III - A propriedade `z-index` especifica a ordem de pilha de um elemento.

- a) Apenas III
- b) Apenas I e II
- c) Apenas I e III
- d) Apenas II e III
- e) I, II e III

- 25) Em relação ao HTML, qual das seguintes alternativas **NÃO** é um valor válido para o atributo `type` da tag `input`?

- a) `datetime-local`
- b) `email`
- c) `range`
- d) `phone`
- e) `week`