



1. (FGV - 2015 – CM/Caruaru - Analista Legislativo - Informática) Analise o código Python a seguir.

```
L1=[10,20,30]
L2=[40,50]
L1.append(L2)
print L1
```

Assinale a opção que descreve corretamente o que acontece quando esse programa é executado no Python 2.7.

- a) Produz uma mensagem de erro, porque tenta executar uma operação inválida.
- b) Exibe "[10, 20, 30, [40, 50]]".
- c) Exibe "[10, 20, 30, 40, 50]".
- d) Exibe "[10, 20, 30], [40, 50]".
- e) Exibe "[]".

#### Comentários:

O comando append inclui o valor da variável L2 na posição final do vetor L1. Como o conteúdo da variável L2 é um vetor de tamanho 2 [40,50], ele que será incluído na 4ª posição de L1. Ué, professor, e porque a resposta não é letra C?? Muito bem observado, padawan! O método append inclui a lista L2 como se fosse só um elemento! Ou seja, a lista final tem 4 elementos ao invés de 5, pois L2 é tratado como se fosse uma coisa só! O gabarito, portanto, é a letra B.

Gabarito: B

2. (IESES - 2015 - IFC-SC - Informática - Programação Web e Dispositivos Móveis) Analise o seguinte trecho de código em Python 3.3.2 e assinale a alternativa correta:

```
class Cachorro:
    def tipo(self):
        return "10"

class Basset(Cachorro):
    def tipo(self):
        return "100"

cachorro1 = Cachorro()
cachorro2 = Basset()

print(cachorro1.tipo())
print(cachorro2.tipo())
```

- a) Ocorrerá um erro de execução na linha "def tipo(self):" logo abaixo de "class Basset(Cachorro):".
- b) A execução de "print(cachorro2.tipo())" mostrará o valor "10" na saída padrão.
- c) A execução de "print(cachorro2.tipo())" mostrará o valor "100" na saída padrão.
- d) Ocorrerá um erro de execução na linha "print(cachorro2.tipo())".

Comentários:

Questão sobre herança, **cachorro2** herda da classe Cachorro, porém o atributo 'tipo' é sobrescrito. Passando a retornar "100".

Gabarito: C

- 
3. (IESES - 2015 - IFC-SC - Programação Web E Dispositivos Móveis) Analise as seguintes afirmativas sobre operadores do Python 3.3.2 e assinale a alternativa correta:

I.  $b \% = a$  é equivalente a  $b = b \% a$ .

II.  $<>$  é o operador de comparação que significa "diferente de" em Python. O operador  $!=$  também é, mas este é um operador obsoleto e não se recomenda mais sua utilização.

III. O operador  $\sim$  é responsável por fazer o "XOR binário", ou seja, realiza um "OU exclusivo" bit a bit.

- a) Somente a afirmativa I é verdadeira.
- b) As afirmativas I, II e III são verdadeiras.
- c) Somente as afirmativas I e III são verdadeiras.
- d) Somente as afirmativas II e III são verdadeiras.

### Comentários:

(I)  $b \% a$  é a forma reduzida de representar  $b = b \% a$ . Assim como  $b += a$  equivale  $b = b + a$ ; (II) O operador de "diferente de" do Python é o `!=`; (III) O `~` é o NOT binário. O XOR binário é o `^`.

Gabarito: A

---

#### 4. (IESES - 2015 - IFC-SC - Programação Web e Dispositivos Móveis) Sobre listas em Python 3.1.5:

- a) `list.remove(a)` remove o primeiro item da lista cujo valor é `a`.
- b) `list.pop(a)` adiciona um item de valor `a` ao início da lista.
- c) `list.append(a)` adiciona um item à lista cujo índice será `a`.
- d) `list.index(a)` retorna o valor do item cujo índice é `a`.

### Comentários:

(a) correto. Caso não exista o valor `"a"`, retorna erro; (b) O `pop` na verdade remove o item na posição `"a"` da lista, ou caso seja chamado como `pop()` - sem argumento - remove o último item da lista. O que adiciona um item ao topo da lista é o `list.insert(0,a)`, onde `0` pode ser substituído por qualquer posição da lista; (c) O `append` adiciona o valor `"a"` na última posição da lista; (d) O `index` retorna a primeira posição que o valor `"a"` é encontrado na lista, caso não seja encontrado, retorna erro.

Aproveitando que estamos falando de lista, seguem os demais comandos para você:

- `list.extend(L)` - é o mesmo que o `append`, mas ao invés de receber apenas um valor, recebe uma nova lista, que será adicionada ao final de `"list"`;
- `list.count(a)` - retorna quantas vezes o valor `"a"` aparece na lista;
- `list.sort()` - ordena os valores da lista;
- `list.reverse()` - reverte os elementos da lista (o último vira o primeiro, e assim sucessivamente).

Gabarito: A

5. (IESES - 2015 - IFC-SC - Programação Web e Dispositivos Móveis) O conjunto correto de palavras reservadas para a construção de uma estrutura de controle em Python 3.4.3 é:

- a) "if", "elif" e "else".
- b) "if", "else if" e "else".
- c) Somente "if"; o restante da estrutura de controle ("senão se" e "senão") é realizado simplesmente com indentação.
- d) "if", "elif" e "else".

Comentários:

Além do if e do else, o Python também possui o "elif" que é o "senão se".

Gabarito: D

6. (UERJ - UERJ - Analista De Sistemas - Grid) Considere o trecho do programa Python abaixo:

```
def dobra (y):  
    x = y + y  
    return x  
  
x = 5  
dobra(x)  
dobra(x)  
print x
```

O valor impresso ao executarmos o programa é:

- a) 5
- b) 10
- c) 15
- d) 25

Comentários:

Dentro do método "dobra(y)" a variável x é de outro escopo. Portanto, a variável x que recebe o valor 5, não é alterada em nenhum momento no corpo principal do programa. Seu valor continua sendo 5.

Gabarito: A

7. (UERJ - UERJ - Analista De Sistemas) A linguagem Python possui a seguinte característica:

- a) é uma linguagem compilada
- b) exige declaração de código
- c) a tupla é um tipo mutável
- d) é orientada a objetos

Comentários:

(a) É uma linguagem interpretada; (b) Entendo esse item como declaração de tipo, então não exige, python é uma linguagem dinâmica; (c) A tupla é equivalente a uma lista, porém é um tipo imutável, seu valor não pode ser modificado; (d) Correto!

Gabarito: D

8. (FCC – TRT/MG - Técnico Judiciário - Tecnologia Da Informação) Considere o código fonte Python abaixo.

```
def calcular(n):  
    resultado = []  
    a, b = 0, 1  
    while a < n:  
        I  
        .....  
    return resultado  
  
res = calcular(100)  
print res
```

Para que seja exibido [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] a lacuna "I" precisa ser preenchida corretamente com:

- a) resultado.add(a)  
a, b = b, a+b
- b) resultado.insert(a)  
a, b = b, a+b
- c) resultado.append(a)  
a, b = b, a+b
- d) resultado.add(a)

a, b = a, a+b  
e) resultado.append(a)  
a, b = a+b, b

#### Comentários:

Primeiro temos que saber que o correto a ser utilizado é o append, pois queremos inserir valores a cada iteração sem perder os anteriores. Então ficamos entre a letra C e a E. Veja que esse algoritmo utiliza a atribuição de tupla, ou seja, "a, b = 0, 1" equivale a a = 0 e b = 1. Perceba que o algoritmo continua como a sequência de fibonacci, a recebe sempre a soma do item atual e do próximo.

Gabarito: C

---

9. (FGV - 2015 – TJ/BA - Analista Judiciário) Analise o trecho de programa, escrito em Python na versão 2.7, mostrado a seguir.

```
X=[]  
for i in range(10,1,-1):  
    X.append(i)  
print X[3:]
```

O trecho exibe:

- a) [7, 6, 5]
- b) [7]
- c) []
- d) [7, 6, 5, 4, 3, 2]
- e) [7, 6, 5, 4, 3, 2, 1]

#### Comentários:

A função range retorna uma lista que inicia no valor 10 e vai até que i seja >1. Com o passo sendo de -1, ou seja, X = [10, 9, 8, 7, 6, 5, 4, 3, 2]. A notação X[3:] significa que é para retornar os elementos da lista começando da posição 3 indo até o final. Lembre-se que o primeiro índice é "0", então o elemento é "7". Logo será exibido "[7, 6, 5, 4, 3, 2]."

Gabarito: D

---

10. (FCC - 2015 - CNMP - Desenvolvimento De Sistemas) Considere os fragmentos de programas Python a seguir:

Fragmento 1:

```
for n in range(2, 10):  
    for x in range(2, n):  
        if n % x == 0:  
            print n, '=', x, '*', n/x  
            break  
    else:  
        print n, 'é um número primo'
```

Fragmento 2:

```
a = ['Casa', 'Mala', 'Prova']  
for x in a:  
    print x, len(x)
```

É correto afirmar que:

- a) o Fragmento 1 está incorreto, pois laços não podem ter uma cláusula else.
- b) no Fragmento 2, a instrução for está incorreta, pois ela não pode iterar sobre a.
- c) o Fragmento 1 está incorreto, pois não é possível iterar sobre sequências numéricas utilizando a função range.
- d) no Fragmento 1 é verificado se o quociente da divisão de n por x corresponde a 0.
- e) os dois fragmentos de código estão corretos.

Comentários:

Python possui diversos recursos para trabalhar com listas, o fragmento 1 utiliza o range que já vimos em outras questões. Esse fragmento verifica se o resto (não o quociente) da divisão é igual a zero. Já o fragmento 2 utiliza o operador "in" o comportamento dele é parecido com o for each, ou seja ele itera sobre todos os objetos do vetor.

11. (FGV - 2015 – TJ/BA - Analista Judiciário) Analise o trecho de programa Python, na versão 2.7, apresentado a seguir.

```
L=[1,2,3,4,5,6,7,8]  
print L[::-1]
```

Ao ser executado, o resultado exibido é:

- a) [1, 2, 3, 4, 5, 6, 7, 8]
- b) [8]
- c) [ ]
- d) [8, 7, 6, 5, 4, 3, 2, 1]
- e) [1]

Comentários:

Para exibir a lista podemos definir 3 coisas: a posição inicial, a posição final e se será na ordem reversa ou não. Nesse caso, as posições inicial e final foram deixadas em branco, ou seja, será o valor padrão, que é a lista em toda a extensão. O "-1" indica que a lista será exibida em ordem reversa, ou seja "[8,7,6,5,4,3,2,1]".

Gabarito: D

12. (CETAP – 2010 – AL/RR – Analista de Sistemas) Sobre a linguagem de programação PYTHON, marque a alternativa INCORRETA.

- a) Python suporta a maioria das técnicas da programação orientada a objetos.
- b) Python suporta e faz uso constante de tratamento de exceções como uma forma de testar condições de erro e outros eventos inesperados no programa.
- c) As funções são definidas em Python utilizando a palavra chave def.
- d) A separação de blocos de código em Python é feita utilizando a indentação de código.
- e) O operador lógico de conjunção ("e", como em a e b) é &&.



## Comentários:

*É uma linguagem que suporta a grande maioria das técnicas de programação orientada a objetos, além de realizar tratamento de exceções. Tudo na linguagem Python é um objeto e sua sintaxe não é baseada diretamente em qualquer linguagem usada comumente. Em vez de vetores, ela inclui três tipos de estrutura de dados: listas, listas imutáveis e hashes, que são chamados de dicionários.*

(a) Conforme vimos em aula, está perfeito.

*É uma linguagem que suporta a grande maioria das técnicas de programação orientada a objetos, além de realizar tratamento de exceções. Tudo na linguagem Python é um objeto e sua sintaxe não é baseada diretamente em qualquer linguagem usada comumente. Em vez de vetores, ela inclui três tipos de estrutura de dados: listas, listas imutáveis e hashes, que são chamados de dicionários.*

(b) Conforme vimos em aula, está perfeito.

*Python permite a definição de funções por meio da palavra-chave **def**, seguida do nome da função e parênteses* (Ex: *def NomeFuncao(parametrosFuncao)*). A indentação também é muito importante, visto que – diferente de outras linguagens que possuem blocos limitados por chaves (`{ }`) ou palavras-chaves (*begin/end*) – em Python blocos são delimitados pela própria indentação/tabulação.

(c) Conforme vimos em aula, está perfeito (apesar de estar escrito Python).

*Python requer uma indentação padronizada. Em outras linguagens, como C/C++ ou JavaScript, a indentação não é necessária devido aos delimitadores de blocos, sendo utilizada somente para melhor visualização. As IDEs que suportam Python têm, em geral, a função de indentação automática, visto que é um pouco complicado ficar indentando tudo na mão.*

(d) Conforme vimos em aula, está perfeito (apesar de estar escrito Python).

Operadores Aritméticos		Operadores Lógicos	
+	Soma	>	Maior
-	Subtração	<	Menor
*	Multiplicação	>=	Maior ou igual
/	Divisão	<=	Menor ou igual
//	Divisão de Inteiros	==	Igual

**	Potenciação	!=	Diferente
%	Módulo	not	Negação
		and	Conjunção
		or	Disjunção

(e) Conforme vimos em aula, trata-se do `and` e, não, `&&`.

Gabarito: E

13. (PaqTcPB – 2012 – UEPB – Analista de Sistemas) No comando de atribuição em Python `valor = raw_input("Digite um valor: ")`, qual o tipo da variável `valor`?

- a) str
- b) bool
- c) int
- d) float
- e) file

Comentários:

Os tipos primitivos de Python podem ser simples ou compostos. Os Tipos Simples são: *int* (*boolean*), *long*, *float* e *complex*. Os Tipos Compostos são *list*, *tuple*, *string* e *dictionary*. Para leitura de dados, a linguagem pode utilizar a função embutida *raw\_input()*, que lê uma string do usuário e armazena em uma variável – seria similar a função *scanf()* da linguagem C.

Conforme vimos em aula, existem algumas funções embutidas! Uma delas é a `raw_input()`, que requisita uma string do usuário e armazena em `valor`.

Gabarito: A

14. (FCC – 2012 – MPE/PE – Analista de Sistemas) Em Python, os métodos de lista permitem utilizar listas como pilhas, onde o item adicionado por último é o primeiro a ser recuperado. Para adicionar um item ao topo da pilha, e para recuperar um item do topo da pilha utilizam-se, respectivamente os métodos:

- a) `append()` e `pop()`.
- b) `insert()` e `top()`.
- c) `addTop()` e `pop()`.
- d) `add()` e `get()`.

e) `addItem()` e `top()`.

Comentários:

Galera, `append()` é utilizado para adicionar um elemento no final de uma lista ou no topo de uma pilha; `pop()` é utilizado para recuperar um item de uma lista ou do topo de uma pilha.

Gabarito: A

15. (FCC – 2012 – TRE/CE – Analista de Sistemas) É INCORRETO afirmar que em Python,

7

a) excetuando-se as classes, tudo é um objeto (tipos, valores, funções, métodos e instâncias), e todos possuem atributos e métodos associados.

b) um ponto fundamental da sintaxe para métodos é que o primeiro argumento é especial, e convencionou-se utilizar o nome `self` para ele.

c) o método construtor (p.ex.: `__init__()`) é um método opcional invocado quando a classe é instanciada.

d) instâncias são objetos criados a partir de uma classe definida pelo programador; o que Python chama de instância é frequentemente denominado objeto em outras linguagens.

e) é possível acessar os atributos da instância diretamente, sem a necessidade de usar um método.

Comentários:

*É uma linguagem que suporta a grande maioria das técnicas de programação orientada a objetos, além de realizar tratamento de exceções. Tudo na linguagem Python é um objeto e sua sintaxe não é baseada diretamente em qualquer linguagem usada comumente. Em vez de vetores, ela inclui três tipos de estrutura de dados: listas, listas imutáveis e hashes, que são chamados de dicionários.*

*Uma coisa interessante é que, quando se utiliza métodos em Python, convencionou-se nomear o primeiro argumento como `self` (Ex: `def metodoQualquer(self, xpto1, xpto2)`). Além disso, quando uma classe é instanciada como um objeto, o método construtor (`__init__`)*

é opcionalmente invocado. *Diferente de outras linguagens, é possível acessar atributos da instância diretamente, sem precisar de métodos.*

Conforme vimos em aula, não há exceções: tudo é objeto (Obs: Phyton →→)!

Gabarito: A

16. (FCC – 2012 – TRE/CE – Analista de Sistemas) Sobre Python é correto afirmar:

a) é uma linguagem compilada, ou seja, o código-fonte de um programa é lido pelo compilador, que cria um arquivo binário, executável diretamente pelo hardware.

8

b) É uma linguagem fortemente tipada, ou seja, é preciso declarar variáveis e seus tipos.

c) Suporta funcionalidades comuns na orientação a objetos: herança, herança múltipla, polimorfismo, reflexão e introspecção.

d) Uma lista em Python é um conjunto de valores acessados por um índice numérico, inteiro, começando em 1. Assim como em outras linguagens, a lista pode armazenar apenas valores de um mesmo tipo.

e) Uma String Python é uma sequência imutável, alocada estaticamente, com restrição de tamanho.

Comentários:

*E o que seria Python? É uma linguagem de programação de alto nível, interpretada, imperativa, multiparadigma, e de tipagem forte e dinâmica.* Vamos falar um pouco sobre essas características! *Por que é de alto nível?* Porque é uma linguagem com nível de abstração bastante elevado, bem longe do código de máquina e mais próximo da linguagem humana. Easy!

(a) Conforme vimos em aula, trata-se de uma linguagem interpretada e, não compilada.

*E o que seria Python? É uma linguagem de programação de alto nível, interpretada, imperativa, multiparadigma, e de tipagem forte e dinâmica.* Vamos falar um pouco sobre essas características! *Por que é de alto nível?* Porque é uma linguagem com

nível de abstração bastante elevado, bem longe do código de máquina e mais próximo da linguagem humana. Easy!

(b) Conforme vimos em aula, é fortemente e dinamicamente tipada, logo é necessário declarar, basta inicializar as variáveis.

*É uma linguagem que suporta a grande maioria das técnicas de programação orientada a objetos, além de realizar tratamento de exceções. Tudo na linguagem Python é um objeto e sua sintaxe não é baseada diretamente em qualquer linguagem usada comumente. Em vez de vetores, ela inclui três tipos de estrutura de dados: listas, listas imutáveis e hashes, que são chamados de dicionários.*

(c) Conforme vimos em aula, está perfe 3>!

*É uma linguagem que suporta a grande maioria das técnicas de programação orientada a objetos, além de realizar tratamento de exceções. Tudo na linguagem Python é um objeto e sua sintaxe não é baseada diretamente em qualquer linguagem usada comumente. Em vez de vetores, ela inclui três tipos de estrutura de dados: listas, listas imutáveis e hashes, que são chamados de dicionários.*

(d) Conforme vimos em aula, listas são um conjunto de objetos, não necessariamente do mesmo tipo, e o primeiro número é zero.

*Os tipos primitivos de Python podem ser simples ou compostos. Os Tipos Simples são: int (boolean), long, float e complex. Os Tipos Compostos são list, tuple, string e dictionary. Para leitura de dados, a linguagem pode utilizar a função embutida raw\_input(), que lê uma string do usuário e armazena em uma variável – seria similar a função scanf() da linguagem C.*

(e) Conforme vimos em aula, elas não possuem restrição de tamanho.

Gabarito: C

17. (CESPE – 2011 – ECT – Analista de Sistemas) A linguagem Python e seu interpretador estão disponíveis para as mais diversas plataformas. Para que seja usado em determinado sistema operacional não suportado, é possível gerar o Python a partir do programa fonte utilizando um compilador C. Nesse caso, o código fonte é traduzido para o formato bytecode, que é multiplataforma e pode ser distribuído de forma independente.

Comentários:

Galera, isso é realmente possível! Questão perfeita!

Gabarito: C

18. (CESPE – 2008 – SERPRO – Analista de Sistemas) Python é uma linguagem livre de alto nível, orientada a objetos e de difícil leitura, pois não permite indentação de linhas de código.

Comentários:

*E o que seria Python? É uma linguagem 1ª de programação de alto nível, interpretada, imperativa, multiparadigma, e de tipagem forte e dinâmica. Vamos falar um pouco sobre essas características! Por que é de alto nível? Porque é uma linguagem com nível de abstração bastante elevado, bem longe do código de máquina e mais próximo da linguagem humana. Easy!*

*Python combina notável poder com sintaxe fácil e clara. Tem interface com muitas bibliotecas, assim como para vários sistemas de janela, e é extensível em C/C++. Também é utilizada como linguagem de extensão para aplicações que necessitam de uma interface programável e é muito comum em aplicativos, jogos, processamento de textos, dados científicos e até motores de busca!*

Conforme vimos em aula, é uma linguagem de fácil leitura justamente por conta de sua indentação.

Gabarito: E

19. (CESPE – 2008 – SERPRO – Analista de Sistemas) O método capitalize da classe String do Python é utilizado para:

- a) remover todos os espaços de uma string.
- b) verificar se todos os caracteres da string são numéricos.
- c) procurar uma substring em uma string retornando seu índice caso seja encontrada.
- d) retornar uma cópia de uma string somente com o primeiro caractere em maiúsculo.
- e) retornar uma cópia de uma string com todos os caracteres em minúsculo.

Comentários:

```
str = "frase com tudo em minúsculo!";  
print "str.capitalize() : ", str.capitalize()
```

Resultado: Frase com tudo em minúsculo!

Conforme mostra o exemplo acima, trata-se de uma função retorna uma cópia de uma string somente com o primeiro caractere em maiúsculo.

Gabarito: D

20. (CESGRANRIO – 2004 – SECAD/TO – Analista de Sistemas) A palavra raise, da linguagem Python, é utilizada para:

- a) converter uma string.
- b) definir uma classe.
- c) gerar uma exceção.
- d) imprimir um resultado.
- e) incorporar um módulo.

Comentários:

```
if alguma coisa:  
    raise Exception('Meu erro!')
```

Conforme mostra o exemplo acima, trata-se de uma palavra para gerar uma exceção. *Simples, não?*

Gabarito: C

21. (CESGRANRIO – 2004 – SECAD/TO – Analista de Sistemas) Um programador de Python recebeu a tarefa de criar uma função chamada calcular que recebe dois parâmetros. Para executar sua atividade, ele deve utilizar a expressão:

- a) def calcular (a,b):
- b) function calcular (a,b):
- c) import calcular (a,b):
- d) procedure calcular (a,b):
- e) sub calcular (a,b):

Comentários:

*Python permite a definição de funções por meio da palavra-chave **def**, seguida do nome da função e parênteses* (Ex: `def NomeFuncao(parametrosFuncao)`). A indentação também é muito importante, visto que – diferente de outras linguagens que possuem blocos limitados por chaves (`{ }`) ou palavras-chaves (*begin/end*) – em Python blocos são delimitados pela própria indentação/tabulação.

Conforme vimos em aula, trata-se da primeira opção.

Gabarito: A

22. (CESPE – 2013 – MPOG – Analista de Sistemas) Em Python, o comando `int("1")` cria um objeto do tipo `int`, que recebe 1 como parâmetro no seu construtor.

Comentários:

Essa é mais uma função embutida! *O que ela faz?* Bem, ela recebe um número como string e o converte em um inteiro (se possível). Esse número serve de parâmetro para o seu construtor.

Gabarito: C

ACERTEI	ERREI