1)

Python foi criado no início dos anos \_\_\_\_\_ por \_\_\_\_\_\_\_ van Rossum no Stichting Mathematisch Centrum (CWI), na Holanda, como sucessor de uma linguagem chamada \_\_\_\_\_\_\_. \_\_\_\_\_\_\_\_ é o principal autor do Python, embora inclua muitas contribuições de outros pesquisadores.

Escolha a opção que completa corretamente as lacunas.

**Alternativas:**

* a)

70; Guido; PSF; Guido

* b)

80; Mark; HIJ; Mark

* c)

90; Guido; ABC; Guido

Alternativa assinalada

* d)

90; Mark; ABC; Mark

* e)

80; Guido; ABC; Guido

2)

Uma das grandes características da linguagem é sua \_\_\_\_\_\_\_\_\_. Uma das principais ideias de Guido é que o código é \_\_\_\_\_\_\_ com muito mais frequência do que está \_\_\_\_\_\_\_\_. Tal aspecto é tão relevante que, um código que segue as regras do idioma python é chamado de “\_\_\_\_\_\_\_\_\_\_\_”. Essas regras são definidas pelo \_\_\_\_\_\_\_ (Python Enhancement Proposal) e dizem respeito a formatação, identação, parâmetros em funções, e tudo mais que possa estar relacionado a sintaxe do código.

Escolha a opção que preenche corretamente as lacunas.

**Alternativas:**

* a)

sintaxe; escrito; especificado; pythonic code; PEP 8

* b)

interface; lido; escrito; python code; PEP 8

* c)

interface; lido; escrito; python code; PEP 10

* d)

sintaxe; lido; escrito; pythonic code; PEP 8

Alternativa assinalada

* e)

sintaxe; lido; escrito; python code; PEP 8

3)

Variáveis são espaços alocados na memória \_\_\_\_\_\_\_, para guardar valores temporariamente. Em Python, esses espaços não precisam ser \_\_\_\_\_\_\_\_\_, ou seja, a variável pode ser alocada sem especificar o \_\_\_\_\_\_\_\_ que ela aguardará. O interpretador Python é capaz de determinar o tipo de dado da variável com base no seu \_\_\_\_\_, ou seja, as variáveis são tipadas \_\_\_\_\_\_\_\_\_ nessa linguagem.

Escolha a opção que completa corretamente as lacunas.

**Alternativas:**

* a)

RAM; especificados; valor; valor; dinamicamente

* b)

RAM; tipados; tipo de dado; valor; dinamicamente

Alternativa assinalada

* c)

RAM; especificados; valor; valor; aleatoriamente

* d)

RAM; tipados; tipo de dado; valor; aleatoriamente

* e)

cache; tipados; valor; valor; dinamicamente

4)

Para saber o tipo de dado que uma variável guarda, o interpretador Python, conta com a função type().

O código a seguir representa criação de variáveis na linguagem Python.

x = 10  
nome = 'aluno'  
nota = 8.75  
fez\_inscricao = True

O código a seguir representa a utilização da função type()

1. print(type(x))  
2. print(type(nome))  
3. print(type(nota))  
4. print(type(fez\_inscricao))

Escolha a opção que representa a saída dos comandos das linhas 1 a 4.

**Alternativas:**

* a)

int

str

float

bool

* b)

integer

str

decimal

bool

* c)

<class 'integer'>  
<class 'string'>  
<class 'float'>  
<class 'bool'>

* d)

integer

string

float

bool

* e)

<class 'int'>  
<class 'str'>  
<class 'float'>  
<class 'bool'>

Alternativa assinalada

5)

A linguagem de programação Python suporta as quatro operações matemáticas básicas, além de outras. O código a seguir é um trecho de um programa em Python.

1. x = 4

2. y = 5

3. print(x / y)

4. print(x // y)

Considerando o trecho de código em Python, escolha a opção que representa o que será impresso pelas linhas 3 e 4.

**Alternativas:**

* a)

Serão impressos os valores 4 (linha 3) e 5 (linha 4)

* b)

Serão impressos os valores 5 (linha 3) e 4 (linha 4)

* c)

Serão impressos os valores 0 (linha 3) e 1 (linha 4)

* d)

Serão impressos os valores 0.8 (linha 3) e 0 (linha 4)

Alternativa assinalada

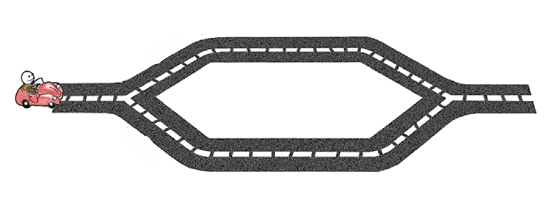
* e)

Serão impressos os valores 0 (linha 3) e 0.8 (linha 4)

6)

Existem diversos elementos que fazem parte das técnicas de programação, independente da linguagem de programação. A Figura 1 apresenta um elemento da programação.

Figura 1 -  Elemento da programação.



Fonte: GRIFFITHS e BARRY, p. 13, 2009.

Escolha a opção que contem a técnica de programação ilustrada na Figura 1.

**Alternativas:**

* a)

Estrutura de repetição.

* b)

Estruturas de decisão.

Alternativa assinalada

* c)

Função.

* d)

Recursividade.

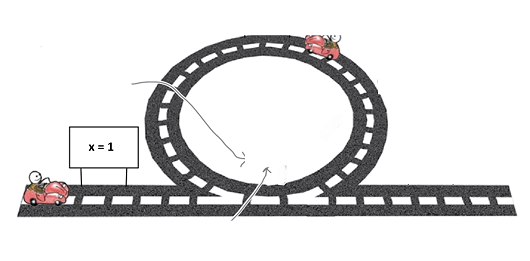
* e)

Módulos.

7)

No desenvolvimento de algoritmos diversos elementos de programação são utilizados para alcançar o resultado esperado. A Figura 1 ilustra um elemento de programação.

Figura 1: Elemento de programação.



Fonte: GRIFFITHS e BARRY, p. 28, 2009.

Escolha a opção que contém o elemento de programação ilustrado na Figura 1.

**Alternativas:**

* a)

Estrutura de decisão.

* b)

Módulo.

* c)

Função.

* d)

Estrutura de repetição.

Alternativa assinalada

* e)

Recursividade.

8)

A condição para um aluno ser aprova é ter média final, de 7 pontos ou superior. Essa condição pode ser implementada na linguagem Python, por meio de uma estrutura condicional.

Considerando a linguagem Python, escolha a opção que implementa corretamente a condição para o aluno ser aprovado.

**Alternativas:**

* a)

if media\_final > 7:

print("aprovado")

* b)

elif media\_final >= 7:

print("aprovado")

* c)

if media\_final >= 7:

print("aprovado")

Alternativa assinalada

* d)

if media\_final < 7:

print("aprovado")

* e)

elif media\_final < 7:

print("aprovado")

9)

A condição para que um aluno seja aprovado é que ele tenha média final de 7 ou superior e tenha 5 ou menos faltas. Tal condição pode ser implementada através de uma estrutura condicional na linguagem Python.

Considerando a linguagem Python, escolha a opção que implementa corretamente a condição para o aluno ser aprovado.

**Alternativas:**

* a)

if media\_final >= 7 and faltas <= 5:

print("aprovado")

Alternativa assinalada

* b)

if media\_final > 7 and faltas < 5:

print("aprovado")

* c)

if media\_final > 7 and faltas <= 5:

print("aprovado")

* d)

if media\_final >= 7 and faltas < 5:

print("aprovado")

* e)

if media\_final >= 7 or faltas <= 5:

print("aprovado")

10)

O primeiro nome do criador da linguagem Python é Guido. Ao atribuir esse nome a uma variável na linguagem Python, estamos criando um objeto iterável, pois textos são sequência de caracteres, ou seja, são objetos em que se pode iterar sobre cada letra.

Considerando a linguagem Python, escolha a opção que implementa corretamente a estrutura para imprimir cada letra do criado da linguagem.

**Alternativas:**

* a)

nome = 'Guido'

for i in range(nome):

print(i)

* b)

nome = 'Guido'

for(i=0; i <=5; i++):

print(i)

* c)

nome = 'Guido'

for i=0; i<= 5 in nome:

print(i)

* d)

nome = 'Guido'

for i in nome:

print(i)

Alternativa assinalada

* e)

nome = 'Guido'

for i in nome{

print(i)

}

11)

Uma função \_\_\_\_\_ é um objeto que está \_\_\_\_\_\_ ao núcleo do \_\_\_\_\_\_\_\_\_, ou seja, não precisa ser feita nenhuma instalação adicional, já está pronto para uso.

Escolha a opção que preenche corretamente as lacunas.

**Alternativas:**

* a)

Python; integrado; módulo

* b)

built-in; integrado; interpretador

Alternativa assinalada

* c)

Python; adptado; interpretador

* d)

built-in; integrado; módulo

* e)

built-in; adptado; interpretador

12)

Em Python, uma função built-in é um objeto que está integrado ao núcleo do interpretador, ou seja, não precisa ser feita nenhuma instalação adicional, já está pronto para uso.

Considerando as funções built-in em Python, escolha a opção correta.

**Alternativas:**

* a)

A função printf() é usada para imprimir um valor na tela.

* b)

A função type() é usada para saber qual o tipo de um objeto.

Alternativa assinalada

* c)

A função input() é usada para abrir um arquivo.

* d)

A função int() é usada para imprimir um valor numérico.

* e)

A função float() é usada para imprimir um valor numérico com casas decimais.

13)

As funções ajudam a dividir um programa em blocos menores e modulares.  À medida que um programa aumenta, as funções o tornam mais organizado e gerenciável.

Considerando a linguagem Python, escolha a opção que implementa corretamente uma função que recebe um valor e retorna o dobro.

**Alternativas:**

* a)

def funcao(valor):

return 2 \* valor

Alternativa assinalada

* b)

def public funcao(valor):

return 2 \* valor

* c)

def void funcao(valor):

return 2 \* valor

* d)

def int funcao(valor):

return 2 \* valor

* e)

def public int funcao(valor):

return 2 \* valor

14)

As funções em Python são definidas usando a palavra-chave do bloco "def", seguida do nome da função como o nome do bloco.

O código a seguir é uma função implementada em Python.

def calcular(v1, v2=0):    return v1 + v2

I. (   ) Para funcionar corretamente (sem erro), a função deve ser invocada passando como parâmetros dois valores.

II. (   ) Como existe um valor default para o segundo parâmetro, a função pode ser invocada passando somente um valor.

III. (    ) Caso seja invocada como calcular(4), o resultado será o próprio valor passado.

Escolha a opção correta.

**Alternativas:**

* a)

V - V - F

* b)

V - F - V

* c)

V - V - V

* d)

F - V - F

* e)

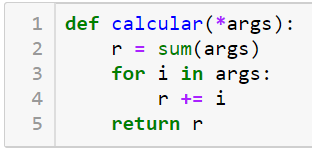
F - V - V

Alternativa assinalada

15)

Funções em Python, podem ser argumentos definidos ou indefinidos. Ao invocar uma função, os parâmetros podem ser passados de forma posicional ou nominal.

O código a seguir é uma implementação em Python



Considerando o código apresentado, escolha a opção que representa o retorno da função para a chamada: calcular(1, 4, 5)

**Alternativas:**

* a)

A função irá retornar o valor 10.

* b)

A função irá retornar o valor 20.

Alternativa assinalada

* c)

A função irá retornar o valor 5.

* d)

A função irá retornar o valor 15.

* e)

A função irá retornar o valor 25.