



Teste de Conhecimento

avale sua aprendizagem



PARADIGMAS DE LING. DE PROGRAMAÇÃO EM PYTHON
EEX0130_202008023386_ESM



Aluno: **ISAAC ALVES PINHEIRO**

Matr.: **202008023386**

Disc.: **LING DE PROG PYTHON**

2021.3 EAD (GT) / EX

Prezado (a) Aluno(a),

Você fará agora seu **TESTE DE CONHECIMENTO!** Lembre-se que este exercício é opcional, mas não valerá ponto para sua avaliação. O mesmo será composto de questões de múltipla escolha.

Após responde cada questão, você terá acesso ao gabarito comentado e/ou à explicação da mesma. Aproveite para se familiarizar com este modelo de questões que será usado na sua AV e AVS.

1. Assinale o grupo de linguagens que são de alto nível e atendem as características do paradigma Orientado a objeto (MODULO 3)

- ☐ Java e Fortran
- ☐ Python, Java, C++ e Pascal
- ☐ Python, Java e Assembly
- ☒ Java, C++, Python e Smalltalk
- ☐ Java, C++ e HTML

Data Resp.: 14/10/2021 14:44:43

Explicação:

2. Qual das linguagens abaixo possui nenhuma abstração (MODULO 1)

- ☐ linguagem JAVA
- ☐ linguagem Assembly
- ☒ linguagem de máquina
- ☐ linguagem Python
- ☐ linguagem HTML

Data Resp.: 14/10/2021 14:45:05

Explicação:

3. (2018 - IF - MT - Informática) Considere o seguinte fragmento de código-fonte escrito na linguagem Python: (MÓDULO 2)

```
def foo(a):  
    return a + a + a  
b = 1  
foo(b)  
foo(b)  
foo(b)
```

Após a execução de todos esses códigos, qual é o valor a ser mostrado mediante o comando abaixo?

```
print(b)
```

- ☐ 9
- ✓ ☒ 1
- ☐ 6
- ☐ 3
- ☐ 0

Data Resp.: 14/10/2021 14:45:23

Explicação:

4. (INMETRO - Pesquisador - Desenvolvimento de Sistemas - 2010)

No que diz respeito à linguagem de programação e às linguagens de script, julgue os itens a seguir. (MÓDULO 1)

```
a, b = 0, 2  
while b < 20:  
    a, b = b, a+b+1  
    print(b)
```

Assinale a opção que apresenta a sequência correta de números impressos no console de um interpretador Python durante a execução dos comandos acima.

- ☐ 1, 1, 2, 3, 5, 8
- ☐ 2, 3, 6
- ☐ 3, 3, 6, 10, 17, 28
- ☐ 2, 2, 4, 6
- ✓ ☒ 3, 6, 10, 17, 28

Data Resp.: 14/10/2021 14:46:18

Explicação:

5. **Ano:** 2010 **Banca:** CETAP **Órgão:** DETRAN-RR **Prova:** CETAP - 2010 - DETRAN-RR - Analista de Sistemas (QConcursos, 2020).

É INCORRETO afirmar que em Python:

- ☐ 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.
- ☐ Um ponto fundamental da sintaxe para métodos é que o primeiro argumento é especial, e convencionou-se utilizar o nome self para ele.
- ☐ É possível acessar os atributos da instância diretamente sem a necessidade de usar um método.
- ✓ ☒ Excetuando-se as classes, tudo é um objeto (tipos, valores, funções, métodos e instâncias), e todos possuem atributos e métodos associados.
- ☐ O método construtor (p.ex.: __init__()) é um método opcional invocado quando a classe é instanciada.

Data Resp.: 14/10/2021 14:46:32

6. Assinale a opção que apresenta a lista que contém apenas as linguagens de programação que são interpretadas em tempo real.

- ☐ C++, JavaScript, Fortran
- ✔ ☒ PHP, Python
- ☐ C, Python
- ☐ C, Pascal, Java
- ☐ C++, Java

Data Resp.: 14/10/2021 14:46:45

Explicação:

7.

2. Ano: 2012 Banca: FCC Órgão: TRE-CE Prova: FCC - 2012 - TRE-CE - Analista Judiciário - Análise de Sistemas e (QConcursos, 2020)
Considere:

```
class A:
    a = 1

class B(A):
    _c = 3

    def __init__(self):
        print(self.a)
        print(self._c)

a = A()
print(isinstance(a,B))

b = B()
print(isinstance(b,B))

b = A()
print(isinstance(b,A))
```

Em relação à sintaxe em Python, é INCORRETO afirmar que:

- ☐ A linha 5 exibe um método construtor.
- ☐ Na linha 13, o objeto b é uma instância da classe A.
- ☐ Na linha 4, o atributo é considerado privado.
- ✔ ☒ Na linha 9, o objeto a é uma instância da classe B.
- ☐ Na linha 3, a classe B está herdando A.

Data Resp.: 14/10/2021 14:47:02

Explicação:

O objeto a da classe A que possui a implementação da subclasse B, logo objeto a é uma instancia de B

8.

Considere o código a seguir, onde temos um servidor Flask escutando na porta 5000, e responda:

```
1 from flask import Flask, request
2
3 app = Flask(__name__)
4
5 @app.route('/teste', methods=['POST'])
6 def teste_post():
7     if request.method == 'POST':
8         return "Requisição POST"
9     else:
10        return "Requisição GET"
11
12 if __name__ == '__main__':
13     app.run()
```

O que será apresentado no navegador se acessarmos a URL `http://127.0.0.1:5000/teste?`

- ✓ ☒ Erro 405 - Método não permitido.
- ☐ Uma página em branco.
- ☐ Erro 404 - Página não encontrada.
- ☐ Requisição POST.
- ☐ Requisição GET.

Data Resp.: 14/10/2021 14:47:11

Explicação:

foi implementado somente o método post, não o método get "?"

9. Considere o trecho de código apresentado a seguir.

```
def multiplicar_por(multiplicador):
    def multi(multiplicando):
        return multiplicando * multiplicador
    return multi

multi_10 = multiplicar_por(10)
print(multi_10(5))
```

Para que as duas últimas linhas não precisem ser modificadas, poderemos trocar a função por qual instrução lambda?

- ✓ ☒ `multiplicar_por = lambda m1: lambda m2: m2 * m1`
- ☐ `multiplicar_por = lambda m1, lambda m2: m2 * m1`
- ☐ `multiplicar_por = lambda m1, m2: m2 * m1`
- ☐ `multiplicar_por = lambda (m1, lambda m2: m2): m2 * m1`
- ☐ `multiplicar_por = lambda m1: m1 * lambda m2: m2 * m1`

Data Resp.: 14/10/2021 14:47:32

Explicação:

m1=10, m2=5

A única opção que apresenta sintaxe correta com a multiplicação é:

multiplicar_por = lambda m1: lambda m2: m2 * m1

sendo multiplicar_por o nome da função

10. Considere o trecho de código de uma aplicação Flask, apresentado a seguir.

```
@app.route('/logar', methods=['GET', 'POST'])
def logar():
    if request.method == 'POST':
        return "Recebeu post !"
    else:
        return "Recebeu get !"
```

Qual será o retorno para uma chamada efetuada, com a digitação do endereço **http://127.0.0.1:5000/logar**, no navegador, pelo usuário?

- ☐ Mensagem de erro por métodos múltiplos
- ☐ Mensagem de erro por ausência de parâmetro
- ☐ Mensagem de erro para página não encontrada.
- ☐ Apresentação do texto "Recebeu post !"
- ☒ Apresentação do texto "Recebeu get !"

Data Resp.: 14/10/2021 14:47:46

Explicação:

Isso cria uma rota /LOGAR que aceita as solicitações GET e POST. As solicitações GET são aceitas por padrão

Se você não especificar um argumento de métodos para LOGAR (), o padrão é aceitar apenas solicitações GET

 Não Respondida  Não Gravada  Gravada

Exercício iniciado em 19/09/2021 12:32:35.