

<b>Iniciado em</b>	sexta-feira, 27 out. 2023, 10:52
<b>Estado</b>	Finalizada
<b>Concluída em</b>	sexta-feira, 27 out. 2023, 11:03
<b>Tempo empregado</b>	11 minutos 25 segundos
<b>Notas</b>	12,00/12,00
<b>Avaliar</b>	<b>10,00</b> de um máximo de 10,00( <b>100%</b> )



**Questão 1**

Correto

Atingiu 1,00 de 1,00

**Dado o código abaixo: (Aula\_02\_2\_Cons\_Except\_Func)**

```
01... def Read_Int(pergunta):
02...     Str1=""
03...     Num1=0
04...     Erro1=1
05...
06...     while (Erro1!=0):
07...         try:
08...             Str1=input(pergunta)
09...             Num1=int(Str1)
10...             if (Num1<0):
11...                 raise Exception('Valor inválido')
12...             Erro1 = 0
13...         except ValueError as ve:
14...             print("Erro1: Voce deve digitar valores inteiros: %s" % ve)
15...             Erro1 = 1
16...         except Exception as ex:
17...             print("Erro2: Erro inesperado: %s" % ex)
18...             Erro1 = 2
19...     return (Num1)
```

**Quais as afirmações estão corretas?**

Escolha uma ou mais:

- ☐ a. O método aceita como parâmetro uma pergunta e sempre retorna um valor inteiro negativo
- ☐ b. A linha 07 faz com que o laço seja executado apenas uma vez em qualquer situação
- ☒ c. O método aceita como parâmetro uma pergunta e retorna um valor inteiro quando não houver exceção ✓
- ☒ d. A linha 12 sinaliza a variável Erro1 com valor zero para que haja a saída do laço While ✓
- ☐ e. As atribuições das linhas 15 e 18 servem para o encerramento do laço While
- ☒ f. As linhas 10 e 11 fazem com que nunca seja aceito um valor negativo ✓



Sua resposta está correta.

As respostas corretas são: O método aceita como parâmetro uma pergunta e retorna um valor inteiro quando não houver exceção, A linha 12 sinaliza a variável Erro1 com valor zero para que haja a saída do laço While, As linhas 10 e 11 fazem com que nunca seja aceito um valor negativo



**Questão 2**

Correto

Atingiu 1,00 de 1,00

**Dado o código abaixo:**

```
01... Mat=[[1.0, -2.0, -3.0],  
02...      [4.0, 5.0, -6.0],  
03...      [7.0, 8.0, 9.0]]  
04...  
05... valor=0  
06... for i in range(0,3,1):  
07...     for j in range(0,3,1):  
08...         try:  
09...             x=Mat[i][j]  
10...             if (x < 0):  
11...                 raise Exception()  
12...             except Exception as ex:  
13...                 x=0;  
14...                 valor = valor + x;  
15...                 print("%4.1f" % valor)
```

**Qual a sequência de valores impressos ?**

Escolha uma opção:

☐ a. 2.0  
11.0  
33.0

☐ b. 3.0  
12.0  
31.0

☒ c. 1.0  
10.0  
34.0 ✓

☐ d. 34.0  
10.0  
4.0

☐ e. 3.0  
15.0  
30.0

Sua resposta está correta.

A resposta correta é:

1.0  
10.0  
34.0



**Questão 3**

Correto

Atingiu 1,00 de 1,00

**Dado o vetor de Strings abaixo:**

```
Vect_Str=[]  
Vect_Str.append("Nome=Marilia da Silva Idade=61")  
Vect_Str.append("Nome=Mauricio Andre da Silva Idade=22")  
Vect_Str.append("Nome=John Schmidt Idade=25")
```

**Os dados a referem-se ao nome e idade de diversas pessoas.****quais as alternativas corretas para gravar esse vetor em um arquivo de texto?**

- ☐ a. 

```
file=open("./texto.txt", "r")  
for i in range(0, len(Vect_Str), 1):  
    file.print("%s\n" % Vect_Str[i])  
    print("Vect_Str[%d] --> %s" % (i, Vect_Str[i]))  
file.close()
```
- ☒ b. 

```
file=open("./texto.txt", "w")  
for i in range(0, len(Vect_Str), 1):  
    file.write("%s\n" % Vect_Str[i])  
    print("Vect_Str[%d] --> %s" % (i, Vect_Str[i]))  
file.close()
```
- ☐ c. 

```
file=open("./texto.txt", "w")  
for i in range(0, len(Vect_Str), 2):  
    file.write("%s\n" % Vect_Str[i])  
    print("Vect_Str[%d] --> %s" % (i, Vect_Str[i]))  
file.flush()
```
- ☐ d. 

```
file=open("./texto.txt", "w")  
for i in range(0, len(Vect_Str), 1):  
    file.console("%s\n" % Vect_Str[i])  
    print("Vect_Str[%d] --> %s" % (i, Vect_Str[i]))  
file.open()
```
- ☒ e. 

```
file=open("./texto.txt", "w")  
for i in range(0, len(Vect_Str), 1):  
    file.write("%s\n" % Vect_Str[i])  
file.close()
```

Sua resposta está correta.

As respostas corretas são:

```
file=open("./texto.txt", "w")  
for i in range(0, len(Vect_Str), 1):  
    file.write("%s\n" % Vect_Str[i])  
    print("Vect_Str[%d] --> %s" % (i, Vect_Str[i]))  
file.close()
```

```
file=open("./texto.txt", "w")  
for i in range(0, len(Vect_Str), 1):  
    file.write("%s\n" % Vect_Str[i])  
file.close()
```



**Questão 4**

Correto

Atingiu 1,00 de 1,00

**Dado o seguinte arquivo de texto:**

```
Nome=Antonio da Silva Idade=18
Nome=Beatriz de Andrade Idade=19
Nome=Carlos Schmidt Idade=20
```

**Quais as alternativas corretas para ler esse arquivo para a memória?**

- ☐ a. 

```
with open("./texto.txt", "w") as file:
    Data=file.readlines()
```
- ☐ b. 

```
with openFile("./texto.txt", "r") as file:
    Data=file.readlines()
```
- ☒ c. 

```
with open("./texto.txt", "r") as file:
    Data=file.readlines()
```
- ☐ d. 

```
with open("./texto.txt", "w") as File:
    Data=file.readLines()
```
- ☐ e. 

```
read open("./texto.txt", "r") as file:
    Data=file.readlines()
```

Sua resposta está correta.

A resposta correta é:

```
with open("./texto.txt", "r") as file:
    Data=file.readlines()
```



## Questão 5

Correto

Atingiu 2,00 de 2,00

Dado o seguinte arquivo de texto formatado em linguagem XML:

```
<banco banco_nome="Banco_Alunos">

  <cidade cidade_key="1" cidade_nome="Sao Bento do Sul" cidade_abrev="SBS"></cidade>
  <cidade cidade_key="2" cidade_nome="Rio negrinho" cidade_abrev="RN"></cidade>

  <aluno aluno_key="1" cidade_key="1" aluno_nome="Joao" aluno_idade="32"></aluno>
  <aluno aluno_key="2" cidade_key="1" aluno_nome="Pedro" aluno_idade="22"></aluno>

</banco>
```

Arraste sobre o código as partes faltantes para ler esse arquivo:

(Aula\_11\_1\_Cons\_ArqXml.py)

```
from xml.dom import  ✓

doc = minidom. ✓ ("./XML_Database/Banco_Alunos.xml")
banco_list=doc. ✓ ('banco')
for node in  ✓ :
    banco_nome = node. ✓ ('banco_nome')
    print("\nBanco:\n%s\n" %  ✓ )

    cidade_list = node.getElementsByTagName( ✓ )
    print("Cidade:")
    for cid in cidade_list:
        cid_key=int(cid.getAttribute( ✓ ))
        cid_nome =  ✓ .getAttribute('cidade_nome')
         ✓ = cid.getAttribute('cidade_abrev')
        print( ✓ %(cid_key, cid_nome, cid_abrev))

    aluno_list =  ✓ .getElementsByTagName('aluno')
    print("Aluno:")
    for alu in aluno_list:
         ✓ = int(alu.getAttribute('aluno_key'))
        cid_key =  ✓ (alu.getAttribute('cidade_key'))
        alu_nome = alu.getAttribute( ✓ )
        alu_idade = int(alu.getAttribute('aluno_idade'))
        print("(%d, %d, %s, %d)" %  ✓ )
```



getAttributebyName

Sua resposta está correta.

A resposta correta é:

**Dado o seguinte arquivo de texto formatado em linguagem XML:**

```
<banco banco_nome="Banco_Alunos">

    <cidade cidade_key="1" cidade_nome="Sao Bento do Sul" cidade_abrev="SBS"></cidade>
    <cidade cidade_key="2" cidade_nome="Rio negrinho" cidade_abrev="RN"></cidade>

    <aluno aluno_key="1" cidade_key="1" aluno_nome="Joao" aluno_idade="32"></aluno>
    <aluno aluno_key="2" cidade_key="1" aluno_nome="Pedro" aluno_idade="22"></aluno>

</banco>
```



**Arraste sobre o código as partes faltantes para ler esse arquivo:**

**(Aula\_11\_1\_Cons\_ArqXml.py)**

```
from xml.dom import [minidom]

doc = minidom.[parse]("./XML_Database/Banco_Alunos.xml")
banco_list=doc.[getElementsByTagName]('banco')
for node in [banco_list]:
    banco_nome = node.[getAttribute]('banco_nome')
    print("\nBanco:\n%s\n" % [banco_nome])

    cidade_list = node.getElementsByTagName(['cidade'])
    print("Cidade:")
    for cid in cidade_list:
        cid_key=int(cid.getAttribute(['cidade_key']))
        cid_nome = [cid].getAttribute('cidade_nome')
        [cid_abrev] = cid.getAttribute('cidade_abrev')
        print("[%d, %s, %s]" % (cid_key, cid_nome, cid_abrev))

    aluno_list = [node].getElementsByTagName('aluno')
    print("Aluno:")
    for alu in aluno_list:
        [alu_key] = int(alu.getAttribute('aluno_key'))
        cid_key = [int](alu.getAttribute('cidade_key'))
        alu_nome = alu.getAttribute(['aluno_nome'])
        alu_idade = int(alu.getAttribute('aluno_idade'))
        print("%d, %d, %s, %d)" % [(alu_key, cid_key, alu_nome, alu_idade)])
```



**Questão 6**

Correto

Atingiu 1,00 de 1,00

**Dado o vetor de Fichas abaixo: (Aula\_12\_1\_Cons\_ArqBinEscrita)**

```
Vet_Fch = []  
Vet_Fch.append(Ficha(0,"Joao da Silva", 10))  
Vet_Fch.append(Ficha(1,"Pedro da Silva", 11))  
Vet_Fch.append(Ficha(2,"Paulo da Silva", 12))
```

**Considere que existe uma classe chamada Ficha que contém Código, Nome e Idade e seus respectivos métodos get e set a respeito de um funcionário. Quais as alternativas corretas para gravar um vetor de objetos dessa classe utilizando a biblioteca Pickle?**

Escolha uma ou mais:

- ☐ a. 

```
file=open("./texto.bin.txt", "w")  
for Fch in Vet_Fch:  
    pickle.dump(Fch, arquivo)  
    file.flush()  
file.close()
```
- ☒ b. 

```
file=open("./texto.bin.txt", "wb")  
for Fch in Vet_Fch:  
    pickle.dump(Fch, file)  
    file.flush()  
file.close()
```
- ☐ c. 

```
file=open("./texto.bin.txt", "wb")  
for Fch in Vet_Fch:  
    print("Vet_Fch --> [%d, %s, %d]" %  
          (Fch.getCodigo(), Fch.getNome(), Fch.getIdade)  
          pickle.save(Fch, flush)  
file.close()
```
- ☒ d. 

```
file=open("./texto.bin.txt", "wb")  
for Fch in Vet_Fch:  
    print("Vet_Fch --> [%d, %s, %d]" %  
          (Fch.getCodigo(), Fch.getNome(), Fch.getIdade)  
          pickle.dump(Fch, file)  
          file.flush()  
file.close()
```
- ☐ e. 

```
File=open("./texto.bin.txt", "wb")  
for Fch in Vet_Fch:  
    print("Vet_Fch --> [%d, %s, %d]" %  
          (Fch.getCodigo(), Fch.getNome(), Fch.getIdade)  
          pickle.dump(Codigo, file)  
          file.flush()  
file.close()
```

Sua resposta está correta.

As respostas corretas são:

```
file=open("./texto.bin.txt", "wb")  
for Fch in Vet_Fch:  
    pickle.dump(Fch, file)  
    file.flush()  
file.close()
```





```
,
file=open("./texto.bin.txt", "wb")
for Fch in Vet_Fch:
    print("Vet_Fch --> [%d, %s, %d]" %
          (Fch.getCodigo(), Fch.getNome(), Fch.getIdade()))
    pickle.dump(Fch, file)
    file.flush()
file.close()
```



## Questão 7

Correto

Atingiu 2,00 de 2,00

Dado o código abaixo que lê arquivos binários:  
(Aula\_12\_2\_Cons\_ArqBinLeitura.py)

```
Ficha.py

#####

class Fichal(object) ✓):
    __Codigo=None
    __Nome=None
    __Idade=None

    def __init__ ✓ (self, Codigo1=0, Nome1="", Idade1=0 ✓):
        self.__Codigo = Codigo1
        self.__Nome = Nome1 ✓
        self.__Idade = Idade1

    def getCodigo(self):
        return(self.__Codigo) ✓

    def setCodigo(self, Codigo1):
        self.__Codigo=Codigo1

    def getNome(self):
        return(self.__Nome)

    def setNome(self, Nome1):
        self.__Nome ✓ =Nome1

    def getIdade(self):
        return(self.__Idade)

    def setIdade(self, Idade1):
        self.__Idade=Idade1

#####

Programa_Principal.py
```



```
import sys
import pickle ✓
from Ficha import Ficha

#####

Vet_Fch = []

file= open ✓ ("./texto.bin.txt", "rb")

i=0
while (True):
    try:
        Fch=pickle.load ✓ (file)
        print("Vet_Fch[%d] --> [%d, %s, %d]" %
              (i, Fch.getCodigo(), Fch.getNome(), Fch.getIdade()))
        Vet_Fch.append ✓ (Fch)
        i += 1
    except EOFError ✓ as eof:
        print("Warning: fim de arquivo: %s" % eof)
        break
    except IOError as io:
        print("Erro: Erro de entrada e saída na leitura: %s" % io)
        break
    except Exception as ex:
        print("Erro: Erro inesperado: %s" % ex)
        break

file.close ✓ ()

#####
```



Arraste sobre o código as partes faltantes:

Sua resposta está correta.

A resposta correta é:

**Dado o código abaixo que lê arquivos binários:**  
**(Aula\_12\_2\_Cons\_ArqBinLeitura.py)**

Ficha.py



```
#####

class Ficha([object]):
    __Codigo=None
    __Nome=None
    __Idade=None

    def __init__(self, [Codigo1=0, Nome1="", Idade1=0 ]):
        self.__Codigo = Codigo1
        self.__Nome = [Nome1]
        self.__Idade = Idade1

    def getCodigo(self):
        return([self.__Codigo])

    def setCodigo(self, Codigo1):
        self.__Codigo=Codigo1

    def getNome(self):
        return(self.__Nome)

    def setNome(self, Nome1):
        [self.__Nome]=Nome1

    def getIdade(self):
        return(self.__Idade)

    def setIdade(self, Idade1):
        self.__Idade=Idade1

#####
```

## Programa\_Principal.py

```
import sys
import [pickle]
from Ficha import Ficha

#####

Vet_Fch = []

file=[open]("./texto.bin.txt", "rb")

i=0
while (True):
    try:
        Fch=pickle.[load](file)
        print("Vet_Fch[%d] --> [%d, %s, %d]" %
              (i, Fch.getCodigo(), Fch.getNome(), Fch.getIdade()))
        Vet_Fch.[append](Fch)
        i += 1
    except [EOFError] as eof:
        print("Warning: fim de arquivo: %s" % eof)
        break
    except IOError as io:
        print("Erro: Erro de entrada e saída na leitura: %s" % io)
        break
    except Exception as ex:
        print("Erro: Erro inesperado: %s" % ex)
        break

file.[close]()

#####
```



Arraste sobre o código as partes faltantes:

**Questão 8**

Correto

Atingiu 1,00 de 1,00

Dado o seguinte vetor que representa uma lista de alunos em Python:

```
Data = {}  
Data['Aluno'] = []  
Data['Aluno'].append({  
    'Nome': 'Joao',  
    'Idade': '28',  
    'Cidade': 'Rio Negrinho'  
})  
Data['Aluno'].append({  
    'Nome': 'Pedro',  
    'Idade': '17',  
    'Cidade': 'Sao Bento do Sul'  
})  
Data['Aluno'].append({  
    'Nome': 'Maria',  
    'Idade': '37',  
    'Cidade': 'Campo Alegre'  
})
```

Quais as alternativas corretas para gravar essa lista em um arquivo JSON?

- ☐ a. `with open('Texto_Json.txt', 'r') as outfile:  
 json.dump(Data, outfile, indentation=4)`
- ☒ b. `with open('Texto_Json.txt', 'w') as outfile:  
 json.dump(Data, outfile, indent=4)`
- ☐ c. `with open('Texto_Json.txt', 'w') as file:  
 json.flush(Data, outfile, indent=4)`
- ☐ d. `with open('Texto_Json.txt', 'w') as outfile:  
 json.write(Data, outfile, indent=4)`
- ☐ e. `with open('Texto_Json.txt', 'r') as outfile:  
 json.save(Data, file, indent=4)`

Sua resposta está correta.

A resposta correta é:

```
with open('Texto_Json.txt', 'w') as outfile:  
    json.dump(Data, outfile, indent=4)
```



**Questão 9**

Correto

Atingiu 1,00 de 1,00

**Dado o seguinte arquivo JSON que representa uma lista de alunos:**

```
{
  "Aluno": [
    {
      "Nome": "Joao",
      "Idade": "28",
      "Cidade": "Rio Negrinho"
    },
    {
      "Nome": "Pedro",
      "Idade": "17",
      "Cidade": "Sao Bento do Sul"
    },
    {
      "Nome": "Maria",
      "Idade": "37",
      "Cidade": "Campo Alegre"
    }
  ]
}
```

**Quais as alternativas corretas para ler esse arquivo JSON para a memória?**

- ☐ a. `with open('Texto_json.txt', 'w') as json_file:`  
`Data = json.Load(json_file)`
- ☐ b. `with open('Texto_json.txt', 'r') as json_file:`  
`Data = json.load(file)`
- ☐ c. `with open('Texto_json.txt', 'w') as json_file:`  
`Data = json.read(json_file)`
- ☐ d. `with open('Texto_json.txt', 'r') as json:`  
`Data = json.read(file)`
- ☒ e. `with open('Texto_json.txt', 'r') as json_file:`  
`Data = json.load(json_file)`

Sua resposta está correta.

A resposta correta é:

```
with open('Texto_json.txt', 'r') as json_file:
    Data = json.load(json_file)
```



**Questão 10**

Correto

Atingiu 1,00 de 1,00

**Dado o seguinte vetor que representa uma lista de alunos em Python:**

```
Data = {}
Data['Aluno'] = []
Data['Aluno'].append({
    'Nome': 'Joao',
    'Idade': '28',
    'Cidade': 'Rio Negrinho'
})
Data['Aluno'].append({
    'Nome': 'Pedro',
    'Idade': '17',
    'Cidade': 'Sao Bento do Sul'
})
Data['Aluno'].append({
    'Nome': 'Maria',
    'Idade': '37',
    'Cidade': 'Campo Alegre'
})
```

**Quais as alternativas corretas para imprimir essa lista no console?**

☒ a. 

```
for p in Data['Aluno']:
    print('Nome: ' + p['Nome'])
    print('Idade: ' + p['Idade'])
    print('Cidade: ' + p['Cidade'])
```

☐ b. 

```
for p in Date['Aluno']:
    print('Nome: ' + p['Nome'])
    print('Idade: ' + p['Idade'])
    print('Cidade: ' + p['Cidade'])
```

☐ c. 

```
for p in Aluno['Data']:
    print('Nome: ' + p['Nome'])
    print('Idade: ' + p['Idade'])
    print('Cidade: ' + p['Cidade'])
```

☐ d. 

```
for p in Data['Aluno']:
    write('Nome: ' + p['Nome'])
    write('Idade: ' + p['Idade'])
    write('Cidade: ' + p['Cidade'])
```

☐ e. 

```
for p in Data['Aluno']:
    print('Nome: ' + Nome)
    print('Idade: ' + Idade)
    print('Cidade: ' + Cidade)
```

**Sua resposta está correta.****A resposta correta é:**

```
for p in Data['Aluno']:
    print('Nome: ' + p['Nome'])
    print('Idade: ' + p['Idade'])
    print('Cidade: ' + p['Cidade'])
```

