Disciplina:

Python

Professor: Nelson Júnior

PUC Minas

EMENTA

- ✓ Introdução O que é Python? Por que usar Python?
- ✓ Variáveis
- ✓ Primeiros Trabalhos
- ✓ Manipulação de Arquivos
- ✓ Conexões com Bancos de Dados e arquivos.

PUC Minas

PONTUAÇÃO

- ✓60 Pontos em Exercício Canvas.
- ✓40 Prova.

Python

Python é uma linguagem de programação relativamente **simples** que foi criada por **Guido van Rossum** em 1991, ela é de **alto nível**, **interpretada** e de **alta produtividade**

- **Simples**
  - Elegante - *Menos linhas de código comparando como Java, C, C++*
  - Documentação Gratuita e de fácil acesso
- **Alto nível**
  - Abstração elevada
  - Longe do código de máquina
  - Próximo à linguagem humana – *É como escrever uma carta*

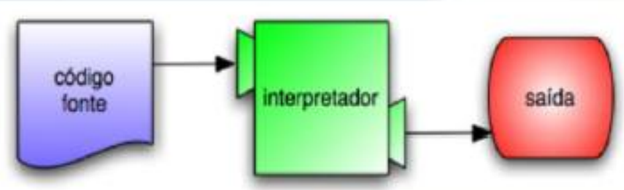
Python

Python é uma linguagem de programação relativamente **simples** que foi criada por **Guido van Rossum** em 1991, ela é de **alto nível**, **interpretada** e de **alta produtividade**

- **Interpretada**

O código fonte é executado por um programa de computador, evita "codifica-compila-roda"

Compilada



Interpretada

- **Alta Produtividade**
  - Imperativa
  - Orientada a objetos
  - Funcional

PUC Minas

Quem usa Python?

Variáveis

python.org.br

Variáveis String

São variáveis do tipo texto, o texto fica entre aspas “ ”

CÓDIGO

```
a = "Olá mundo"
b = "Hello World"
c = "Olá PET-SI"
d = "Olá UFRRJ"
e = "Curso"
f = "Python"

print(a)
print(b)
print(c)
print(d)
print(e+" de "+f)
```

SAÍDA

```
Olá mundo
Hello World
Olá PET-SI
Olá UFRRJ
Curso de Python
```

CÓDIGO

```
a = "PET-SI"
b = " está oferecendo um
minicurso de Python"
c = " para alunos da"
d = " UFRRJ"

print("O "+a+b+c+d)
```

SAÍDA

```
O PET-SI está oferecendo um
minicurso de Python para
alunos da UFRRJ
```

PUC Minas

Variáveis

python.org.br

Variáveis String – Leitura

CÓDIGO

```
nome = input('Entre com o seu nome: ')
print(nome)
```

```
Entre com o seu nome: fulano de tal
fulano de tal
```

CÓDIGO

```
nome = input('Digite seu nome: ')
curso = input('Digite seu curso: ')

print('Olá '+nome+', ficamos felizes em conhecer você, gostamos muito do curso de '+curso)
```

```
Digite seu nome: Lucas
Digite seu curso: Sistemas de Informação
Olá Lucas, ficamos felizes em conhecer você, gostamos muito do curso de Sistemas de Informação
```

Variáveis – Exemplos

python.org.br

CÓDIGO

```
a = "abcdefghijkl"
print(a[9])
print(a[0])

print(a[3])
print(a[3:])
```

SAÍDA

```
j
a
d
defghijkl
```

CÓDIGO

```
nome = input("Digite seu primeiro nome: ")
print("A primeira letra do seu nome é: "+nome[0])
```

```
Digite seu primeiro nome: Lucas
A primeira letra do seu nome é: L
```

Variáveis

python.org.br

Variáveis Numéricas

O Python possui alguns tipos numéricos pré-definidos:

- **Inteiros (*int*)**
- Ponto flutuante(*float*)
- Booleanos (*bool*)
- Complexos (*complex*)

Elas suportam as operações matemáticas básicas

```
>>> a, b = 1, 2.5          # atribui 1 a "a" e 2.5 a "b"
>>>                        # um inteiro e um ponto flutuante
>>> c = True               # booleano
>>> z = 3 + 4j             # complexo
>>>
>>> a + b                  # resultado em ponto flutuante
3.5
>>> int(a + b)             # resultado inteiro
3
>>> b * z                  # resultado complexo
(7.5+10j)
>>> type(z)                # mostra a tipagem da variável
<type 'complex'>
```

EXERCÍCIOS

Exercícios

python.org.br

Exercício 1:

Faça um programa que mostre o tradicional “Hello World!” na tela

Exercício 2:

Faça um programa que peça um número e então mostre a mensagem: *O número informado foi [número]*.

Exercício 3: (Sem estruturas de repetição)

Faça um programa que peça 5 itens e suas respectivas quantidades e mostre na tela a lista de itens com a quantidade,

Item 1 – Quantidade: V
Item 2 – Quantidade: W
Item 3 – Quantidade: Y
Item 4 – Quantidade: X
Item 5 – Quantidade: Z

Exercício

python.org.br

Criar uma lista de compra com as seguintes regras:

- É necessário um total de 5 frutas;

- A **primeira** fruta deve **custar 1,00**;
- A **segunda** fruta deve **custar o dobro do valor da primeira**;
- A **terceira** fruta deve **custar metade do valor da primeira**;
- A **quarta** fruta deve **custar 3 vezes o valor da terceira fruta**;
- A **quinta** fruta deve **custar metade do valor da quarta**;

- Cada fruta deve possuir uma variável;
- Usar a menor quantidade possível de variáveis;
- Todas as frutas e seus valores devem ser impressos no seguinte formato:

“A fruta _____ custa _____”

Dicionários

(Dicionários)

- Dicionários são coleções de elementos onde é possível utilizar um índice de qualquer tipo **imutável**.

- Os dicionários implementam mapeamentos que são coleções de associações entre pares de valores
  - O primeiro elemento é a **chave**
  - O segundo elemento é o **conteúdo/valor**

```
DICIONARIO = {"ALAN":'001',"AMARILDO":'002',"ANA":'003',"ARISTIDES":'004'}
```

- As chaves dos dicionários são armazenadas por tabelas de espalhamento (Hash Tables)
- Diferente de listas, não existe uma ordem específica de armazenamento no dicionário

Dicionários

python.org.br

Criação do Dicionário

dic = {"Nome":'Larissa',"Sobrenome":'Maria'}

dic = {"Alan":'001',"Amarildo":'002',"Ana":'003',"Aristides":'004'}

Operações com Dicionário

print(dic["Nome"]) - Imprime o conteúdo da chave Nome

print(dic["Sobrenome"]) – Imprime o conteúdo da chave Sobrenome

print(dic.**keys**()) – Imprime apenas as chaves

print(dic.**values**()) – Imprime apenas os conteúdos

print(dic.**items**()) – Imprime as chaves e conteúdos

Inserindo um novo item no dicionário

dic["Idade"] = '18‘

Alterando o valor das chaves

dic[“Nome"] = ‘Rose'

Dicionários

python.org.br

Função GET: retorna o valor da chave e NONE caso não exista

```
print(dic.get('Larissa'))
print(dic.get('Rose'))
```

Função DEL: Apaga determinado item do dicionário

```
del dic["Nome"]
```

Função CLEAR: Apaga todo o dicionário

```
dic.clear()
```

Função COPY: Copia o conteúdo de um dicionário para outro

```
dic2 = dic.copy()
```

Dicionários – Exemplos

python.org.br

CÓDIGO

```
listatel = {"ana":210012,"bianca":210045,"camila":210019}

print(listatel["ana"])
print(listatel["bianca"])
print(listatel["camila"])

print(listatel.keys())
print(listatel.values())
```

SAÍDA

```
210012
210045
210019
dict_keys(['camila', 'bianca', 'ana'])
dict_values([210019, 210045, 210012])
```

EXERCÍCIOS

PUC Minas

Exercícios - Dicionários

python.org.br

Exercício:

Faça um dicionário que contenha os dados de uma pessoa, são os seguintes dados: (Preencha os dados iniciais como preferir)

- Nome
- Ultimo Nome
- Idade
- Curso
- Endereço

a) Imprima o dicionário completo
b) Imprima cada um dos 5 itens separadamente
c) Exclua a chave **CURSO** do dicionário
d) Altere o **ULTIMO NOME** para Lopes
e) Imprima novamente o dicionário completo
f) Imprima apenas o endereço
g) Crie uma cópia do dicionário e altere **Nome** e **Idade**
h) Imprima o segundo dicionário completo

OPERADORES ARITMÉTICOS

- Operador: Lógico

```
> and    "E" operador lógico
> or     "ou" operador lógico
> not    "Negação" operador lógico
>!=      "Diferente" operador
         lógico
```

OPERADORES ARITMÉTICOS

- Operador: Lógico

```
>    Maior
<    Menor
>=   Maior igual
<=   Menor igual
=    Atribuição
==   Igualdade
```

EXERCÍCIOS

CONTROLE DE FLUXO: IF / ELIF/ ELSE

- If simples

```
> a = 12
> if a > 10:
..    print "%s maior que 10" % a
..
```

CONTROLE DE FLUXO: IF / ELIF/ ELSE

- If com exceção

```
> a = 3
> if a >= 10:
..    print                    que 10” % a
.. else:
..    print “%s menor que 10” % a
```

CONTROLE DE FLUXO: IF / ELIF/ ELSE

- If com outro if na exceção

```
> If a  ==  b :
>       print (“igual”)
> else:
>       Print (“diferente”)
```

CONTROLE DE FLUXO: IF / ELIF/ ELSE

- If com outro if na exceção

```
> If a  ==  b :
>       print (“igual”)
> elif  a == c :
        print (“ A é igual a C “)
> else:
>       print (“diferente”)
```

Não existe CASE/ SWIT em python

logo se usa ELIF

PRATICANDO

- Calculo de notas, DADOS :

- Nota 1 + Nota 2 / 2
- Se média <= 3 imprimir reprovado
- Se >3 e < 5 imprimir optativa
- Se >=5 imprimir aprovado

REPETIÇÃO (LOOP) : FOR

- Somando a lista

```
> total = 0
> for numero in numeros:
..      total = total + numero
..
> print total
```

REPETIÇÃO (LOOP) : WHILE

Condição de parada normal

```
x = 100
while x > 0:
    print "x > 0"
    x = x - 1
```

Condição de parada dentro do break

```
i = 0
while True:
    print "Não vou parar nunca!"
    i = i + 1
    if i > 100:
        break
```

EXERCÍCIOS

EXERCÍCIOS

- Faça um programa que permita somar apenas os números pares da sequencia de inteiros contida no intervalos [x, y].

EXERCÍCIOS

- Leia um número e calcule e imprima sua tabuada

FUNÇÕES

- Criando e usando uma função

```
> def somaTres(numero):
..      return numero + 3
..
> print somaTres(3)
```

FUNÇÕES

FUNÇÃO RANGE

#Gerando sequencia de números em uma lista

```
> numeros = range(1,101)
> print numeros
```

Acessando um item da lista

> print numeros [10]

Acessando partes da lista

> print numeros [50:60]

FUNÇÕES

Números aleatórios

Importar a bibliotéca “random”

```
from random import*
aux = random()  # gera números aleatórios
print(int(aux*10))
```

FUNÇÕES

Números aleatórios

```
print uniform(10, 20)
print randint(100, 1000)
print randrange(100, 1000, 2)
```

A função **random**() retorna um float x tal que 0 <= x < 1.
A função **uniform(10,20)** retorna um float x tal que 10 <= x < 20.
A função **randint(100,1000)** retorna um inteiro x tal que 100 <= x < 1000.
A função **randrange(100,1000,2)** retorna um inteiro x tal que 100 <= x < 1000 e x é par (ou seja, passo 2).

EXERCÍCIOS

Escreva uma aplicação de dicionário com três funções: adicionar um termo ao dicionário, procurar um termo no dicionário e listar todos os termos existentes.