Advanced Institute for Artificial Intelligence

Python

Advanced Institute for Artificial Intelligence

https://advancedinstitute.ai

Sumario

Introdução

- Python é uma linguagem interpretada
- Caminho do Python no Sistema
  - which python
- Versão do Python
  - python -V

Usando Python

- Iniciando interpretador Python
  - python
  - Python 3.6.8 —Anaconda, Inc.— (default, Dec 30 2018, 01:22:34)
  - [GCC 7.3.0] on linux
  - Type "help", "copyright", "credits" or "license" for more information.
  - >>> Esse é o prompt para receber comandos python
- Ctrl+D sai do interpretador

Usando Python

Comando print

- print "hello world"
- Em python 2 é possível utilizar dessa forma:
- print "hello world"
- Em python 3 é obrigatório utilizar ( )
- print ("hello world")

Comentários no código

- : comentando uma linha
- "' : começar e terminar bloco de comentário
- """ : começar e terminar bloco de comentário

Usando Python

Indentação

- O controle de início e fim deblocos de código é feito por meio de Indentação
- Indentação pode ser controlada por um tamanho fixo de espaços em branco
- Exemplo

```
print ("teste")
if (i == 0):
        print ("0")
else:
        print ("outro valor")
        if (i >= 0):
                print (>=0")
```

Tipos de dados - Números

Existem três tipos numéricos em python: números inteiros, números de ponto flutuante e números complexos.

- Booleanos são um subtipo de números inteiros.
- Inteiros têm precisão ilimitada.
- Números de ponto flutuante são geralmente implementados usando tipo Double em C

Tipos de dados - Strings

Strings podem ser manipuladas de diversas maneiras em Python

- podem ser representadas usando aspas simples ' ' ou aspas duplas " "
- É possível utilizar catacteres escape

Funções

- A palavra-chave def é usada para definir funções
- Deve ser definida antes de ser utilizada
- O valor de retorno padrão é None

Função

Argumento pode ser gerado da seguinte forma:

- nome de variável
- nome de variável e tipo padrão

Escopo de variável

- variáveis possuem escopo local ao bloco onde são criadas
- Pode ser definidas variáveis globais

Funções

Função sem argumentos:

```
def greeting():
    print("hello world")


greeting()
```

Argumento de Função

```
def numsquare(num):
        return num * num

number=10
numsquare(number)

def numsquare(num=10):
        return num * num

numsquare()
```

Obtendo dados do usuário

função input() é utilizada para aguardar um valor digitado no terminal pelo usuário.

```
usrip = input("número inteiro: ")
usrnum = int(usrip)
sqrnum = numsquare(usrnum)
print("Square of entered number is: ".format(sqrnum))

usrip = input("float: ")
usrnum = float(usrip)
sqrnum = numsquare(usrnum)
print("Square of entered number is: ".format(sqrnum))

usrname = input("nome: ")
print("nome: ",usrname)
```

Usando bibliotecas adicionais

A palavra reservada import permite adicionar pacotes que não são nativos do Python

```
import subprocess
# Executa um comando linux no terminal
subprocess.call('date')
```

A palavra reservada from premite importar apenas parte de um pacote

exemplo:

```
from sklearn.model_selection import train_test_split
```

Estruturas e Função de Controle

Argumento pode ser gerado da seguinte forma:

- if
- for
- while

Estruturas e Função de Controle

if

- As instruções if avaliam uma condição, caso seja verdadeira executa o bloco seguinte
- Pode ser combinado com uma estrutura else, que é executada quando a condição não é verdadeira no bloco if

Exemplo:

```
var = 100
if (var==100):
        print("100")
else:           print("not 100")
```

Estruturas e Função de Controle

for

- executam um certo bloco de código para um número conhecido de iterações.
- Um bloco de código pode ser executado para o número de itens existentes em uma lista, dicionário, variável de sequência ou tupla
- Um bloco de código pode ser executado em um intervalo contado de etapas

Exemplo:

```
a=(10,20,30,40,50)
for b in a:
print "square of " + str(b) + " is " +str(b*b)
```

Estruturas e Função de Controle

while

- O loop while é executado enquanto uma declaração condicional retorna true
- A instrução condicional é avaliada toda vez que um bloco de código é executado
- A execução para no momento em que a instrução condicional retorna false.

Exemplo:

```
count = 0
while (count < 9):

print("iteração",count)
count+=1
```