2013

Decode Team

Instituto Politécnico de tomar

Portugol

Equivalências de estruturas entre Portugol e Python

Índice

[Nota Geral: 3](#_Toc359443934)

[Algumas notas sobre Python: 3](#_Toc359443935)

[Estrutura Início 3](#_Toc359443936)

[Início: 3](#_Toc359443937)

[Estrutura Fim 3](#_Toc359443938)

[Fim: 3](#_Toc359443939)

[Variáveis 4](#_Toc359443940)

[Equivalência entre TIPOS de variáveis 4](#_Toc359443941)

[Definição e atribuição de variáveis 4](#_Toc359443942)

[Estruturas input/output 5](#_Toc359443943)

[Input – Ler 5](#_Toc359443944)

[Output – Escrever 5](#_Toc359443945)

[Estruturas de Decisão 6](#_Toc359443946)

[Condição *“if”* e *“if else”* 6](#_Toc359443947)

[Exemplos práticos 6](#_Toc359443948)

[Condição *“while”* 7](#_Toc359443949)

[Condição *“do while”* 8](#_Toc359443950)

[Exemplos práticos 8](#_Toc359443951)

[Estrutura Conector 9](#_Toc359443952)

[Conector 9](#_Toc359443953)

[Funções 10](#_Toc359443954)

[Definir funções 10](#_Toc359443955)

[Definir função 10](#_Toc359443956)

[Exemplos do uso de funções 11](#_Toc359443957)

[Estrutura de retorno 11](#_Toc359443958)

[Return 11](#_Toc359443959)

[Operadores 12](#_Toc359443960)

[Aritméticos 12](#_Toc359443961)

[Lógicos 12](#_Toc359443962)

[Relacionais 12](#_Toc359443963)

[ANEXO 13](#_Toc359443964)

[Algoritmo com o uso da condição “if” 13](#_Toc359443965)

[Fluxograma 13](#_Toc359443966)

[Código 13](#_Toc359443967)

[Esquema detalhado 14](#_Toc359443968)

[Algoritmo com o uso da condição “if else” 15](#_Toc359443969)

[Fluxograma 15](#_Toc359443970)

[Código: 15](#_Toc359443971)

[Esquema detalhado 16](#_Toc359443972)

[Algoritmo com o uso da condição “while” 17](#_Toc359443973)

[Fluxograma 17](#_Toc359443974)

[Código 17](#_Toc359443975)

[Esquema detalhado 18](#_Toc359443976)

[Algoritmo com o uso da condição “ do while” 19](#_Toc359443977)

[Fluxograma 19](#_Toc359443978)

[Código 19](#_Toc359443979)

[Esquema detalhado 20](#_Toc359443980)

[Algoritmo com o uso de uma função 21](#_Toc359443981)

[Fluxogramas 21](#_Toc359443982)

[Código 21](#_Toc359443983)

[Esquema detalhado 22](#_Toc359443984)

# Nota Geral:

Devido à especificação da linguagem, a tradução só é possível depois de ser executado o fluxograma.

# Algumas notas sobre Python:

* É case sensitive.
* Necessita sempre do código indentado pois é a única maneira de o Python saber o que faz está dentro e fora dos ciclos porque não possui um ‘End’.
* As funções são habitualmente definidas na parte inicial do programa, antes das instruções que farão a execução principal do programa.
* Para fazer comentários em Python utiliza-se o #
  + Exemplo: #comentário

# Estrutura Início

## Início:

Início

O Python não tem equivalência ao Inicio.

# Estrutura Fim

## Fim:

Fim

O Python não tem equivalência ao Fim.

# Variáveis

## Equivalência entre TIPOS de variáveis

|  |  |  |
| --- | --- | --- |
| TIPO | Portugol | Python |
| Inteiro | Inteiro | Int |
| Real | Real | Float |
| Texto | Texto | String |
| Lógico | Logico | boolean |

Tabela - Tipos de variáveis

## Definição e atribuição de variáveis

Nota: Ao utilizar variáveis em Python não é necessário informar o tipo desta variável como fazemos em outras linguagens.

Mas por vezes e para evitar que o Python confunda o nome de uma variável que tem um valor inteiro com uma string, basta antes do colocar antes do nome da variável Int ou se for outro tipo colocar o respectivo tipo antes( real, boolean ou float)

Exemplo:

if int(n)%2==0:

* Neste caso indicamos que a variável *n* é do tipo *int*(inteiro) para que o programa não confuda com uma string.

Int(n)

Exemplo:

n= input('Digite um numero: ')

if **int**(n)%2==0:

print('Numero Par')

# Estruturas input/output

## Input – Ler

variavel

<variável>= input('<texto>')

## Output – Escrever

expressao

Para escrever no ecrã:

Print '<texto>'

Ou

Print ( '<texto>',<variável)

# Estruturas de Decisão

## Condição *“if”* e *“if else”*

condição

FALSE

TRUE

Instruções 2

Instruções 1

Para TRUE, escrever:

**if** <condição>**:**

<instruções>

Para FALSE:

Se Instruções 2 for igual a (conector) não fazer nada.

Senão, escrever:

**else:**

<instruções>

### Exemplos práticos

#### Condição “if”

n= input('Digite um numero: ')

if int(n)%2==0:

print('Par')

#### Condição “if else”

n= input('Digite um numero: ')

if int(n)%2==0:

print('Numero Par')

else:

print('Numero Impar')

## Condição *“while”*

condição

true

false

instruções

**while** <condição>**:**

   <instruções>

## Condição *“do while”*

instruções

TRUE

condição

FALSE

Instruções

**While True**:

<Instruções>

**If**<condição>:

<Instruções>

**break**

### Exemplos práticos

#### Condição “while”

i=1

while int(i)<=10:

print(i)

i=i+1

#### Condição “do while”

while True:

i=input('Digite um número:')

int(i)<0

if int(i)>=0:

break

# Estrutura Conector

## Conector

Se for uma condição “*do while*” escrever:

While True:

instrução

Senão, escrever:

Break

# Funções

## Definir funções

Exemplo( a , b, . . .)

### Definir função

**def**<nome\_função>:

<instruções>Chamada de funções

Fact(Nome)

NOME(PARAMETRO);

### Exemplos do uso de funções

k=input('Digite um numero: ')

def fact(k):

if k == 0:

return 1

else:

return (int(k) \* fact( int(k) - 1 ))

j=fact(k)

print(fact(k))

# Estrutura de retorno

## Return

expressao

return expressao;

# Operadores

### Aritméticos

|  |  |  |
| --- | --- | --- |
| Nome | Portugol | Python |
| Adição | a + b | a + b |
| Subtração | a – b | a - b |
| Divisão | a / b | a / b |
| Multiplicação | a \* b | a \* b |
| Resto da divisão inteira |  | a % b |
| Potenciação |  | \*\* |

Tabela - Equivalência de operadores aritméticos

### Lógicos

|  |  |  |
| --- | --- | --- |
| Nome | Portugol | Python |
| Disjunção | a && b | and |
| Conjunção | a || b | or |
| Negação | Não | not |

Tabela - Equivalência de operadores lógicos

### Relacionais

|  |  |  |
| --- | --- | --- |
| Nome | Portugol | Python |
| Igual | a == b | a == b |
| Diferente | a != b | a != b |
| Maior | a > b | a > b |
| Maior ou igual | a >= b | a >= b |
| Menor | a < b | a < b |
| Menor ou igual | a <= b | a <= b |

Tabela - Equivalência de operadores relacionais

# ***ANEXO***

Para uma compreensão mais abrangente do uso das estruturas, ficam alguns exemplos mais extensivos, com o uso de várias estruturas em algoritmos completos.

## Algoritmo com o uso da condição “if”

**Problema:** Verificar se um número introduzido pelo utilizador é par.

### Fluxograma

Inicio

n

n%2=0

TRUE

FALSE

“Par”

Fim

### Código

n= input('Digite um numero: ')

if int(n)%2==0:

print('Par')

### Esquema detalhado

**Sem Correspondência**

**n= input('Digite um numero: ')**

**if int(n)%2==0:**

**print('Par')**

**Sem Correspondência**

**Sem Correspondência**



## Algoritmo com o uso da condição “if else”

**Problema:** Verificar se um número introduzido pelo utilizador é par ou ímpar.

### Fluxograma

### 

TRUE

Fim

“Par”

n%2=0

n

Inicio

FALSE

“Impar”

### Código:

n= input('Digite um numero: ')

if int(n)%2==0:

print('Numero Par')

else:

print('Numero Impar')

### Esquema detalhado

**Sem Correspondência**

**n= input('Digite um numero: ')**

**if int(n)%2==0:**

**else:**

**print('Numero Impar')**

**print('Numero Par')**

**Sem Correspondência**

**Sem Correspondência**

“Impar”

FALSE

TRUE

Fim

“Par”

n%2=0

n

Inicio

## Algoritmo com o uso da condição “while”

**Problema:** Escrever um número de 1 a 10.

### Fluxograma

Inicio

i=1

i <= 10

FALSE

TRUE

i

i=i + 1

Fim

### Código

i=1

while int(i)<=10:

print(i)

i=i+1

### Esquema detalhado

**Sem Correspondência**

**i= 1**

**while int(i)<=10:**

**print(i)**

**i=i+1**

**Sem Correspodência**

i=i + 1

i=1

Fim

i

i <= 10

Inicio

FALSE

TRUE

## Algoritmo com o uso da condição “ do while”

**Problema:** Pedir um número positivo.

### Fluxograma

Inicio

i = 0

i

TRUE

i < 0

FALSE

Fim

### Código

i=0

while True:

i=input('Digite um número:')

int(i)<0

if int(i)>=0:

break

### Esquema detalhado

**Sem Correspondência**

**i=0**

**While True:**

**i=input('Digite um número:')**

**int(i)<0**

**if int(i)>=0:**

**break**

**Sem Correspondência**

Inicio

i = 0

i

TRUE

i < 0

FALSE

Fim

## Algoritmo com o uso de uma função

**Problema:** Factorial de um número.

### Fluxogramas

**Código principal**

**Função fact(k)**

fact(k)

Inicio

TRUE

FALSE

k > 2

i = 5

j = fact ( i )

k

k \* fact(k - 1)

j

Fim

Fim

### Código

def fact(k):

if k >2 :

return (k \* fact(k - 1) )

else:

return k

i=5

j=fact(i)

print (j)

### Esquema detalhado

**Sem Correspondência**

**i=5**

**j=fact(i)**

**print (j)**

**Sem Correspondência**

Inicio

i = 5

j = fact ( i )

j

Fim

**def fact(k):**

**if k >2 :**

**return (k \* fact(k - 1) )**

**else:**

**return k**

**Sem Correspondência**

