

Unidade 5: Introdução à Programação com Python

Prof. Daniel Caetano

Objetivo: Explicitar os elementos básicos envolvidos na programação com a linguagem Python.

Bibliografia: ASCENCIO, 2007; MEDINA, 2006; SILVA, 2010; SILVA, 2006.

INTRODUÇÃO

Nas aulas anteriores estudamos os conceitos básicos de programação e, em algum detalhe, a linguagem Português Estruturado (ou Portugol). Nesta aula os mesmos conceitos serão apresentados com relação à linguagem Python.

Como será possível ver, a linguagem Python não difere, essencialmente, do Portugol; entretanto, o Python é uma linguagem mais prática, exigindo menos digitação do programador.

Ao final desta aula, você saberá como declarar variáveis, como ler e escrever valores, como executar expressões aritméticas e armazenar seus resultados em variáveis, usando a linguagem Python.

1. INTRODUÇÃO À LINGUAGEM Python

A linguagem *portugol* é importante para aprendermos, mas profissionalmente ela não é usada. A função do *portugol* é proporcionar uma transição suave entre os fluxogramas e outras linguagens de programação profissionais, como Python.

A grande maioria das regras permanece a mesma, de maneira que os conceitos vistos anteriormente para o *portugol* serão rerepresentados aqui com o formato de Python.

A estrutura de um programam mínimo em Python tem a seguinte "cara":

```
# Descrição do Programa  
# Aqui será escrito nosso programa
```

Observe que toda linha que começar com um **#** será ignorada pelo computador: trata-se do que chamamos de **comentário**. A linguagem Python considera que o programa começa no início do arquivo e acaba no fim desse arquivo, não há marcadores de início e fim.

Uma coisa **muito importante** neste momento é que **o Python é sensível a maiúsculas e minúsculas**. Isso significa que os textos **devem ser digitados exatamente na forma como foram apresentados**.

NOTA: Os únicos lugares onde não é necessário uma preocupação maior com maiúsculas e minúsculas são nos comentários (linhas que se iniciam com #) ou em textos que serão impressos, representados entre aspas: "tExTo".

2. VARIÁVEIS

Diferentemente do *portugol*, em que a declaração de variáveis é obrigatória, no Python essa declaração é opcional. É de bom tom fazê-la, no entanto, para garantir que nenhuma variável esteja “indefinida” e tentemos imprimi-la sem querer. A declaração de variáveis em Python é, simplesmente, armazenar um valor do tipo desejado nessa variável. Em geral usa-se o seguinte:

Variáveis Inteiras:

```
nome_da_variavel = 0
```

Variáveis Reais:

```
nome_da_variavel = 0.0
```

Variáveis String:

```
nome_da_variavel = ""
```

Variáveis Lógicas:

```
nome_da_variavel = false
```

Vamos apresentar um programa exemplo que declara a variável **idade** para inteiros:

```
# Exemplo de declaração de variável
```

```
idade = 0
```

Da mesma forma que em *portugol*, podemos declarar variáveis em várias linhas:

```
# Exemplo com várias variáveis
```

```
idade = 0
```

```
dia = 0
```

```
nota = 0.0
```

Pode-se declarar mais de uma variável por linha, separando-se as definições pelo símbolo ; (ponto-e-vírgula):

```
var_ineira = 0; var_real = 0.0
```

```
# Declarando mais de uma variável na mesma linha
idade = 0; dia = 0; nota = 0;
```

3. OPERADOR DE ATRIBUIÇÃO

Tudo que foi dito para o operador atribuição de *portugol* vale para Python, as únicas diferenças é que usaremos o sinal de igualdade = para representar a atribuição e a linha precisa ser finalizada com um ponto-e-vírgula:

```
nome_da_variavel = expressao;
```

Da mesma forma, o lado esquerdo só pode ser uma variável, e o lado direito é uma expressão que sempre será processada **antes** que a atribuição ocorra.

```
# Guardando um valor diferente na variável
# Declarando a variável
idade = 0

# Trocando o valor da variável
idade = 20
```

4. OPERADORES MATEMÁTICOS

Os operadores matemáticos do C/C++ são, basicamente, os mesmos do *portugol*. Os símbolos usados para cada um deles estão indicados a seguir, bem como sua prioridade:

<u>Operação</u>	<u>Sinal</u>	<u>Prioridade</u>
Adição:	+	1
Subtração:	-	1
Multiplicação:	*	2
Divisão:	/	2
Resto da Divisão:	%	2
Divisão Inteira:	//	2
Potência:	**	3

OPCIONAL: Existem, porém, alguns operadores especiais, usados em situações específicas:

<u>Operador</u>	<u>Uso</u>	<u>Equivalência</u>
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
**=	x **= y	x = x ** y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Exemplos de operações:

```
X = 37 % 7
X = A + B * 2
X = (A + B) * 2
```

```
# Exemplo de operação
idade = 20
idade = idade + 10
```

Ou, de outra forma:

```
# Exemplo de operação
idade = 20
idade += 10
```

5. SAÍDA DE DADOS

No Python existem diversas formas de realizar a saída de dados. Usaremos com a função **print** para isso. A sintaxe é a seguinte:

```
print ( dado_a_ser_escrito_na_tela )
```

O **dado a ser escrito** pode ser um número:

```
# Escrevendo um número na tela
print( 10 )
```

Um texto (lembrando que texto deve SEMPRE vir entre aspas):

```
# Escrevendo um texto

print( "Um texto qualquer" )
```

Ou, ainda, pode ser um nome de variável, situação na qual o **valor da variável** será impresso:

```
# Escrevendo o valor de uma variável

idade = 20
idade = idade + 10
print ( idade )
```

O comando **print** aceita, ainda, que indiquemos uma expressão matemática, quando então ele **imprime o resultado da expressão**:

```
# Escrevendo o resultado de uma expressão

print( (10+20)*3 )
```

Podemos compor uma linha com várias instruções **print**:

```
# Escrevendo várias vezes...

idade = 20
print( "Minha idade é: " )
print( idade )
```

Ou podemos pedir que um único **print** imprima várias coisas, apresentando todos os valores na mesma linha:

```
# Escrevendo múltiplos valores na mesma linha

idade = 20
print( "Minha idade é: ", idade )
```

Observe a repetição do sinal `,` entre os diferentes dados a imprimir.

6. ENTRADA DE DADOS

A entrada de dados no Python é feita com a função **input**. Ele tem a seguinte sintaxe:

variável = **input** ("Pergunta para o usuário")

```
# Lendo um dado digitado pelo usuário

idade = 0
idade = input( "Digite sua idade: " )
print( "Sua idade é: ", idade )
```

O valor digitado pelo usuário irá ser armazenado na variável. Se ela for omitida, o valor será desprezado.

7. EXEMPLO

Como exemplo, vamos apresentar o programa que calcula a média de dois números no formato em Python:

```
# Calcula a media de dois números

n1 = 0; n2 = 0; m = 0

print( "Calcula a média de dois números" )
n1 = input( "Digite o primeiro número: " )
n2 = input( "Digite o segundo número: " )
m = (n1+n2)/2
print( "A média é: ", m )
```

Observe que mesmo tendo declarado os valores n1, n2 e m com inteiros, o valor de m será real no caso de pedirmos a média entre 2 e 3. Isso ocorre porque, em Python, a variável não tem um tipo fixo: o tipo da variável é definido pelo valor armazenado nela.

8. BIBLIOGRAFIA

ASCENCIO, A.F.G; CAMPOS, E.A.V. **Fundamentos da Programação de Computadores**. 2ed. Rio de Janeiro, 2007.

MEDINA, M; FERTIG, C. **Algoritmos e Programação: Teoria e Prática**. 2ed. São Paulo: Ed. Novatec, 2006.

SILVA, I.C.S; FALKEMBACH, G.M; SILVEIRA, S.R. **Algoritmos e Programação em Linguagem C**. 1ed. Porto Alegre: Ed. UniRitter, 2010.