

Curso	Objetivo	Disciplina
Análise e Desenvolvimento de Sistemas	Python Básico e Python para Banco de Dados	Banco de Dados

Variáveis

Todo algoritmo que se possa construir utilizará conjuntos de dados. Tais dados podem ser basicamente números e caracteres isolados ou, de alguma forma, agrupados.

Para que um algoritmo possa ser implantado em um computador é preciso que exista uma forma de armazenamento dos dados que serão manipulados. Assim, chega-se ao conceito de variável. Em programação de computadores uma variável é um elemento da linguagem que ocupa um ou mais bytes na memória RAM do computador. Esse local da memória é capaz de reter, ou seja, armazenar o elemento de dado. No programa a variável é identificada por um nome ou identificador. Desta forma pode-se entender que "do ponto de vista" do programador a variável é um nome que contém um dado; e "do ponto de vista" do computador a variável é um endereço de memória que retém um conjunto de bits que representam esse dado.

Por exemplo, imagine que se queira escrever um algoritmo capaz de calcular a área de um retângulo. Nesse algoritmo haverá três dados. Dois de entrada: a base e a altura do retângulo. O terceiro, o resultado, é a área calculada usando-se os outros dois. Assim sendo pode-se esquematizar um rascunho de algoritmo que seja o seguinte:

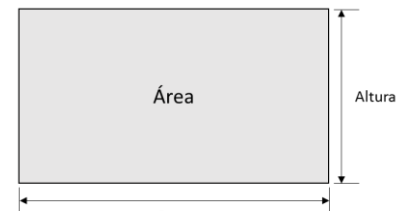


Figura 1- Exemplo de cálculo de área

1. Obter o valor da Base
2. Obter o valor da Altura
3. Calcular $Area = Base \times Altura$
4. Exibir a Área calculada

Os quatro passos sequenciais aqui exibidos representam um algoritmo simples e os identificadores Base, Altura e Área são variáveis.

Figura 2- Primeiro algoritmo

Classificação das Variáveis

Neste texto, quanto à divisibilidade, adota-se a classificação das variáveis como indivisíveis e divisíveis. Uma variável indivisível é aquela que contém um dado único que não pode ser dividido ou acessado em partes. Neste caso sempre que tal variável for acessada ou alterada, trabalha-se com o dado integral nela contido. No exemplo acima as três variáveis são indivisíveis. Variáveis assim são chamadas de Simples

Em contraposição às Variáveis Simples, existem as Variáveis Compostas que são divisíveis, ou seja, é possível acessá-las e alterá-las em partes, ou elementos. Conjuntos de caracteres, também conhecidos como *Strings*, e Listas contendo números diversos são exemplos de Variáveis Compostas.

Essas últimas serão estudadas mais adiante. A seguir será tratado dos tipos de dados que podem estar contidos nas Variáveis Simples.

Tipos Simples de Dados

Em qualquer linguagem de programação os Tipos Simples de Dados geralmente são:

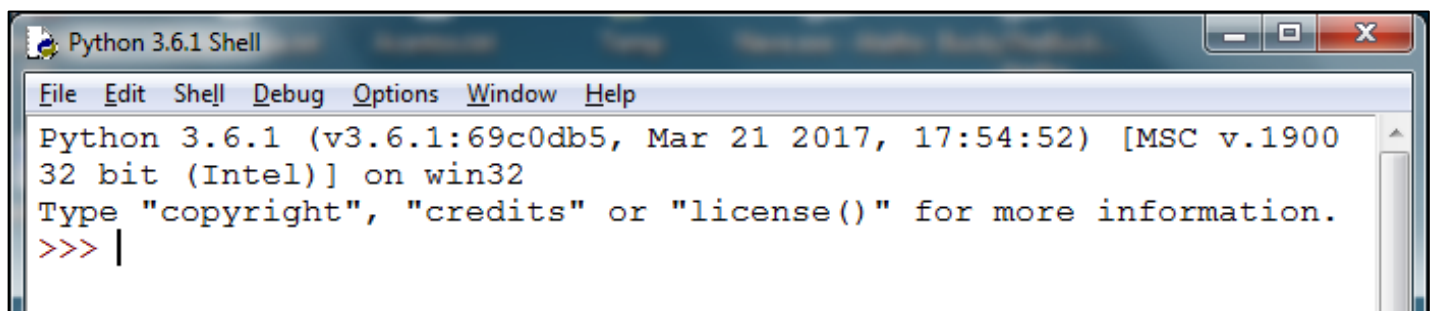
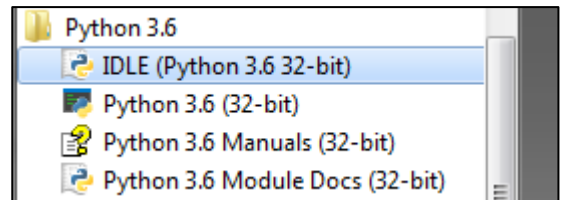
- Número Inteiro;
- Número Real (também conhecido como Número de Ponto Flutuante);
- Caractere;
- Lógico (também conhecido como Booleano);

A partir deste ponto é preciso tomar o cuidado de considerar que cada linguagem de programação trata este aspecto de modo bastante específico. Sendo assim, convém começar a tratar da nossa linguagem alvo que é o Python. Nela, os Tipos Simples de Dados são:

- Número Inteiro (int);
- Número Real (float);
- Número Complexo (complex);
- Lógico (bool);

Nesta linguagem não existe o tipo Caractere mencionado acima, na forma de tipo simples de dado. É claro que é possível trabalhar com caracteres em Python, porém isso é feito usando-se *strings*, que são um tipo composto.

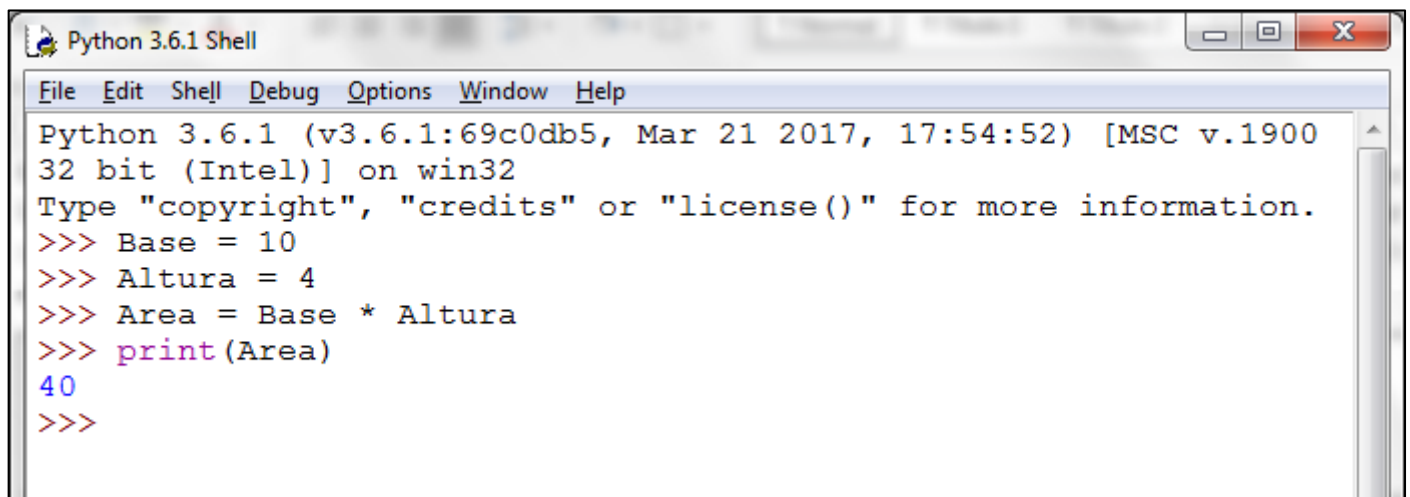
Agora é hora de começar a trabalhar com Python. Para isso, supondo que você já tenha o Python instalado, abra o ambiente IDLE. No sistema operacional Windows procure o grupo de programas Python 3.6 e execute o IDLE conforme mostrado ao lado. A tela que será aberta será semelhante à mostrada abaixo.



Na maioria das linguagens, para testar qualquer elemento é preciso escrever um programa completo e executá-lo. Em Python isso não é necessário. O ambiente IDLE acima é interativo, de modo que é possível digitar um comando e, ao apertar Enter, de imediato verificar sua execução. Assim sendo, experimente digitar as três linhas a seguir:

```
>>> Base = 10
>>> Altura = 4
>>> Area = Base * Altura
>>> print(Area)
```

O resultado que você obterá deve ser semelhante à próxima figura.



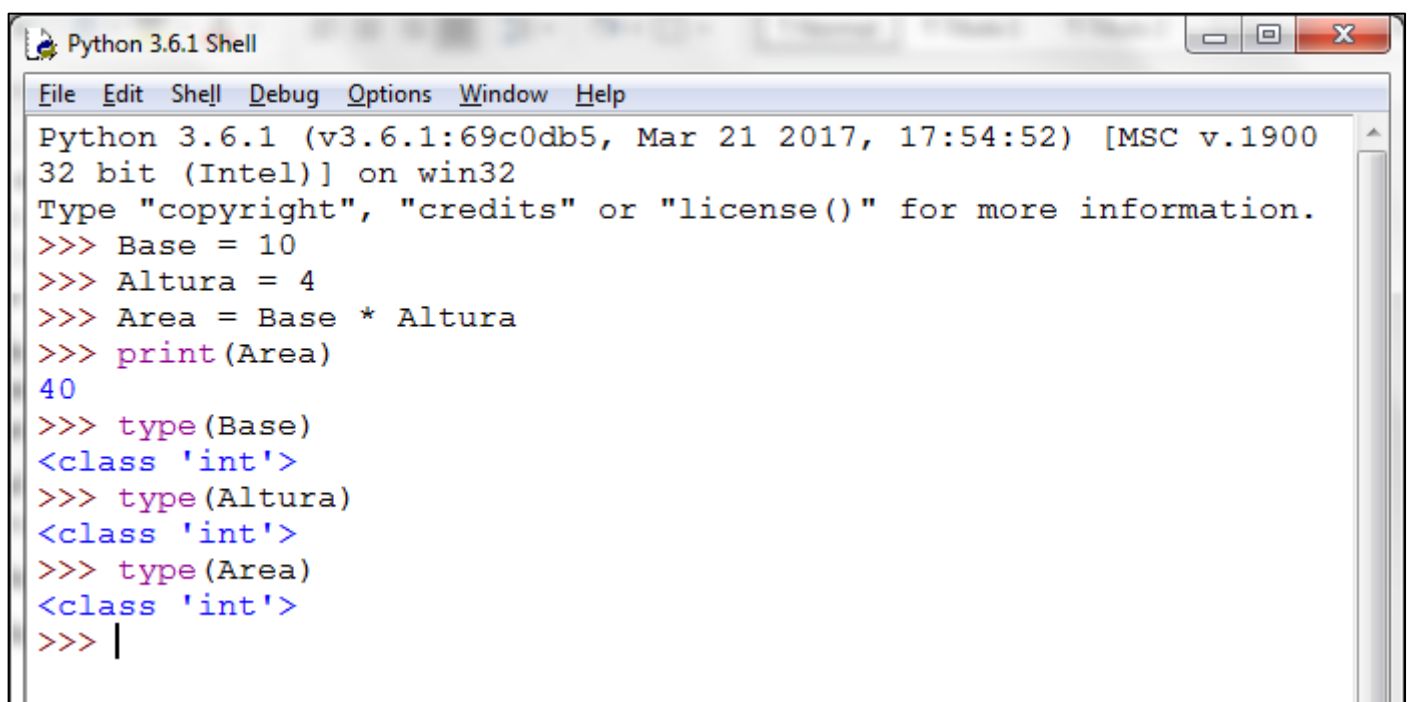
Os quatro comandos acima são a tradução para Python do algoritmo da Figura 2. Os dois primeiros comandos atribuem valores fixos às variáveis Base e Altura. O terceiro comando contém do lado direito uma expressão aritmética de multiplicação cujo resultado é calculado e armazenado na variável Area. Por fim, o comando print é usado para exibir na tela o conteúdo da variável Area.

Agora vamos tratar de um aspecto importante do Python e que quem já conhece alguma outra linguagem de programação talvez tenha estranhado. As variáveis em Python não precisam ser explicitamente declaradas no início do programa para serem usadas. Isso é assim na maioria das linguagens, mas em Python não.

Para que uma variável comece a existir, basta que a ela se atribua um valor inicial. Neste momento o Python cria esse identificador e reserva um espaço de memória para armazenar o dado contido.

E qual é o tipo de dados dessas variáveis criadas?

No caso deste exemplo, todos são números inteiros, formalmente: são do tipo 'int'. Para constatar isso utilize o comando **type**. Veja a figura abaixo onde foi usado o comando type três vezes com o qual verifica-se que, de fato, as variáveis criadas são do tipo 'int'.

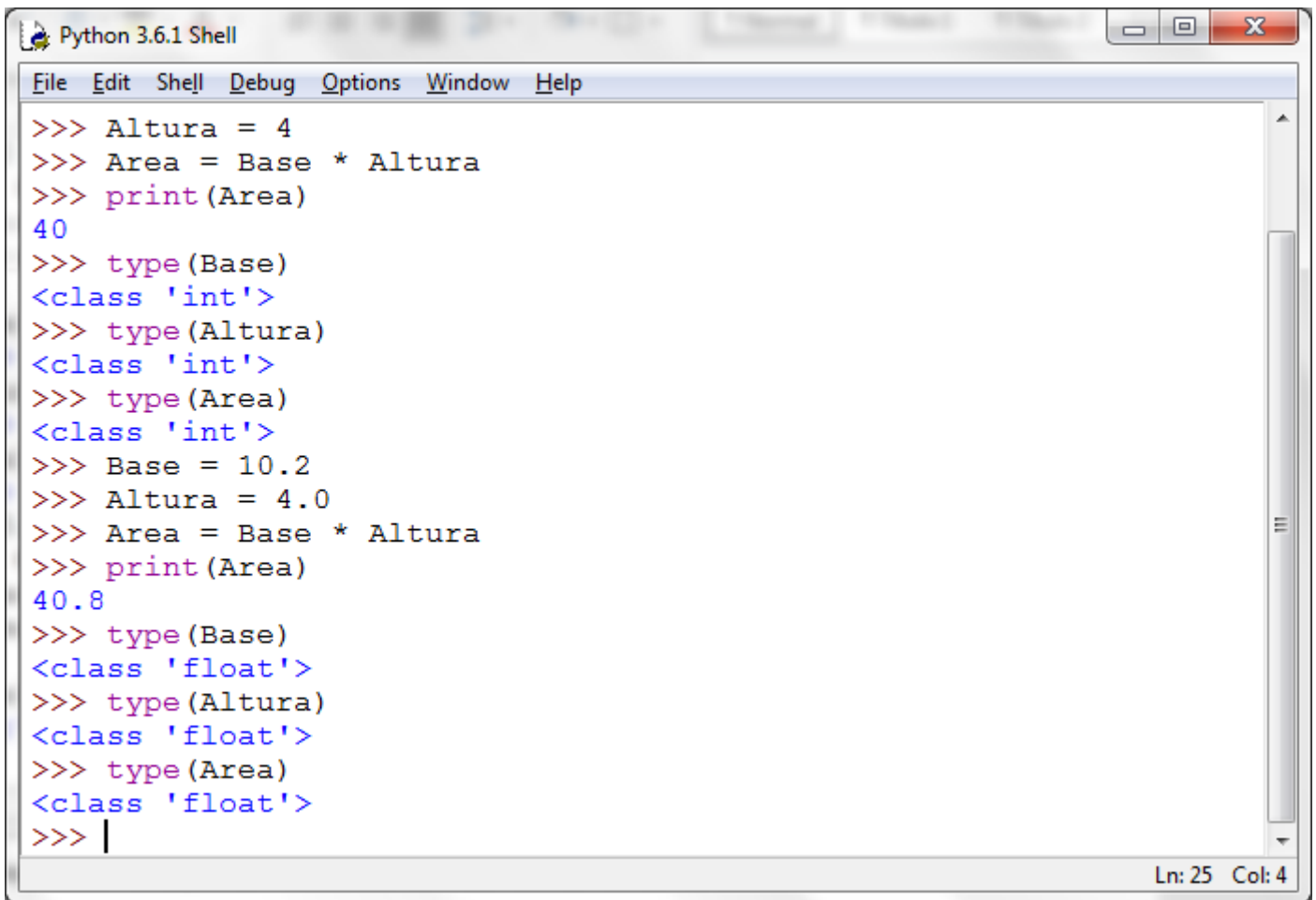
A screenshot of a Python 3.6.1 Shell window. The window title is "Python 3.6.1 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The shell displays the following text:

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> Base = 10
>>> Altura = 4
>>> Area = Base * Altura
>>> print(Area)
40
>>> type(Base)
<class 'int'>
>>> type(Altura)
<class 'int'>
>>> type(Area)
<class 'int'>
>>> |
```

Caso queiramos criar uma variável do tipo 'float', ou seja, capaz de conter um número real, basta atribuir a ela um valor que contenha a parte decimal. Poderíamos fazer:

```
>>> Base = 10.2
>>> Altura = 4.0
>>> Area = Base * Altura
>>> print(Area)
```

Veja na figura a seguir. Esses quatro comandos acima foram usados e a área calculada e exibida foi de 40,8, um número real, portanto.



```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
>>> Altura = 4
>>> Area = Base * Altura
>>> print(Area)
40
>>> type(Base)
<class 'int'>
>>> type(Altura)
<class 'int'>
>>> type(Area)
<class 'int'>
>>> Base = 10.2
>>> Altura = 4.0
>>> Area = Base * Altura
>>> print(Area)
40.8
>>> type(Base)
<class 'float'>
>>> type(Altura)
<class 'float'>
>>> type(Area)
<class 'float'>
>>> |
Ln: 25 Col: 4
```

No exemplo acima, propositalmente, foram utilizados os mesmos nomes de variáveis e em seguida foi utilizado o comando `type` três vezes para verificar de que tipo é cada variável. Perceba que se antes elas eram `'int'`, após o uso com números reais elas passaram a ser `'float'`.

Na linguagem Python essa é uma característica marcante. As variáveis podem mudar de tipo, e isso pode ocorrer sempre que houver uma atribuição de valor à variável. No momento de atribuição de um valor à variável, o Python verificará qual a opção de tipo mais adequada e fará com que a variável seja desse tipo.

Para quem já possui alguma experiência em programação isso pode parecer bastante estranho, mas Python é assim mesmo.

Experimente outros tipos de variáveis, com as sugestões abaixo.

```
>>> X = 1.0
>>> type(X)
>>> Y = 18
>>> type(Y)
>>> Z = X + Y
>>> type(Z)           Z é a soma de um float com um int. Qual é o tipo de Z?
>>> a = 5 + 3j
>>> type(a)
>>> type(A)           Experimente usar o type com a variável A (maiúsculo) e
                        relate o que ocorreu
```

Expressões aritméticas

As expressões aritméticas são construídas utilizando-se variáveis, operadores aritméticos e funções matemáticas.

Os operadores aritméticos disponíveis em Python são os indicados na tabela abaixo. Execute todos os exemplos da tabela com os valores: A = 14 e B = 5. Siga o esquema abaixo e compare os resultados que você obteve com os resultados esperados indicados na tabela

```
>>> A = 14
>>> B = 5
>>> C = A + B
>>> print(C)
```

Operação	Operador	Exemplo	Resultado esperado
Adição	+	C = A + B	19
Subtração	-	C = A - B	9
Multiplicação	*	C = A * B	70
Divisão	/	C = A / B	2.8
Divisão inteira	//	C = A // B	2
Módulo (resto da divisão de inteiros)	%	C = A % B	4
Potenciação	**	C = A ** B	537824

As expressões podem conter vários operadores e as regras de precedência da álgebra são estritamente respeitadas. Então em uma expressão como $R = 2 * A + B$ primeiro será calculada a multiplicação e depois a adição. Caso queira, nesta expressão, primeiro calcular a adição e depois a multiplicação, então é necessário usar parênteses: $R = 2 * (A + B)$.

Questões

1. Faça $X = 0.0$ e $Y = 18$. Verifique tipo de dado que o Python atribuiu a cada um. Faça $Z = X + Y$ e verifique o resultado calculado e armazenado em Z. Verifique qual foi o tipo de dado da variável Z.
2. Atribua um valor qualquer a uma variável a (minúsculo). Use o comando type ou o comando print com a variável A (maiúsculo). Relate o que aconteceu.
3. Faça $S = \text{"FATEC SÃO CAETANO DO SUL"}$. Verifique qual é o tipo de dado da variável S.
4. Reproduza todos os exemplos de operações aritméticas acima e compare os valores obtidos por você com os valores esperados constantes na tabela.
5. Escreva a sequência de comandos necessária para cálculo a área de um triângulo de base 9 e altura 6.
6. Escreva a sequência de comandos para calcular o salário bruto de um profissional que ganha por hora, sabendo que ele ganha R\$ 14,25/h e trabalhou 163 horas normais e 20 horas extras (pagam o dobro).
7. Escreva em Python as seguintes expressões aritméticas e teste-as para os valores $A = 4$, $B = 5$, $C = 1$

$$R = \frac{A + B}{2} \qquad R = \frac{3A + 2B}{A + B} \qquad x = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \qquad Z = 7.6A - B^{1,7}$$

8. Pesquise o que é o comando *input* e descubra como usá-lo.