



Introdução

GLEISON ANDRADE

Por que Python?



Por que Python?

- Simplicidade e clareza;
- Clara e objetiva;
- Software Livre;
- Pode ser utilizada em quase todos os S.O.;
- Grande crescimento em diversas áreas da T.I.;

Por que Python?

- Legibilidade
 - Outras Linguagens utilizam inúmeras marcações, como ponto e vírgula (;);
 - Marcadores de início e fim de blocos ou palavras especiais (begin/end);
- Linguagem completa
 - Banco de dados;
 - Construir interfaces gráficas;
 - Jogos.

Preparando Ambiente

- Instalar o interpretador Python;
 - É um programa que aceita comandos escrito em Python e os executa, linha a linha;
 - Ele traduz nossos programas em um formato que pode ser executado pelo computador;
 - Verificar se o que foi escrito está correto, caso não esteja, mensagens de erros são mostradas.

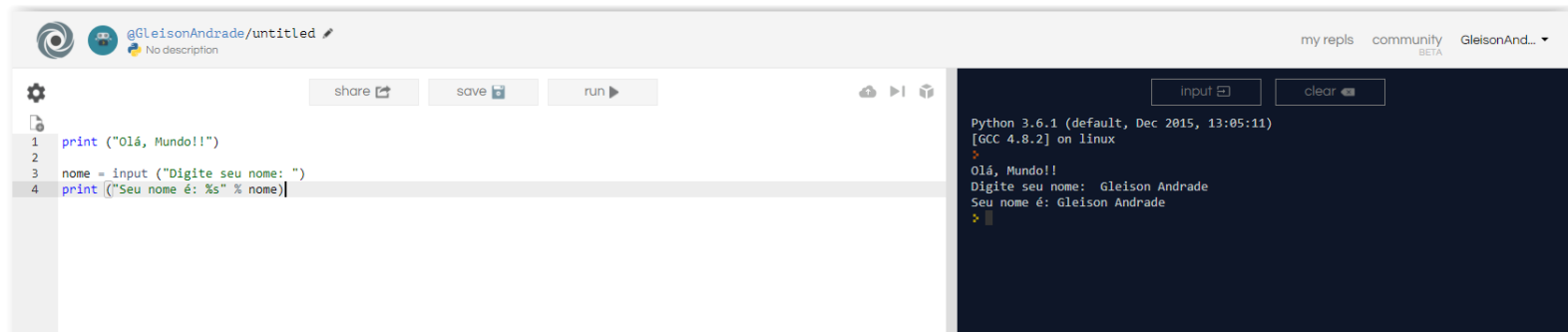
Instalando no Windows

[HTTPS://WWW.PYTHON.ORG](https://www.python.org)



Interpretador online

(<https://repl.it/languages/python3>)



The screenshot shows the Repl.it online Python interpreter interface. The top bar displays the user profile '@GleisonAndrade/untitled' and navigation links for 'my repls', 'community', and the user's name 'GleisonAnd...' with a dropdown arrow. Below the top bar, there are buttons for 'share', 'save', and 'run'. The main area is split into two panels. The left panel contains a code editor with the following Python code:

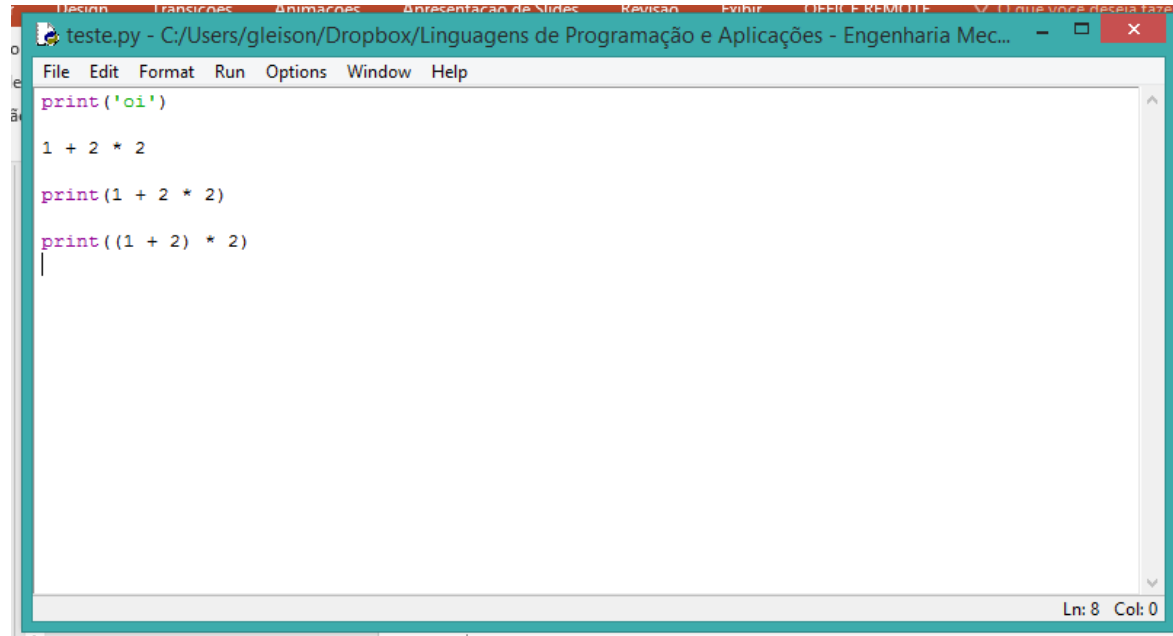
```
1 print("Olá, Mundo!!")
2
3 nome = input("Digite seu nome: ")
4 print("Seu nome é: %s" % nome)
```

The right panel shows the output of the code execution. It includes a 'Python 3.6.1 (default, Dec 2015, 13:05:11) [GCC 4.8.2] on linux' header, a prompt character '>', and the output of the program:

```
Olá, Mundo!!
Digite seu nome: Gleison Andrade
Seu nome é: Gleison Andrade
```

There are also 'input' and 'clear' buttons above the output area.

Editando um arquivo

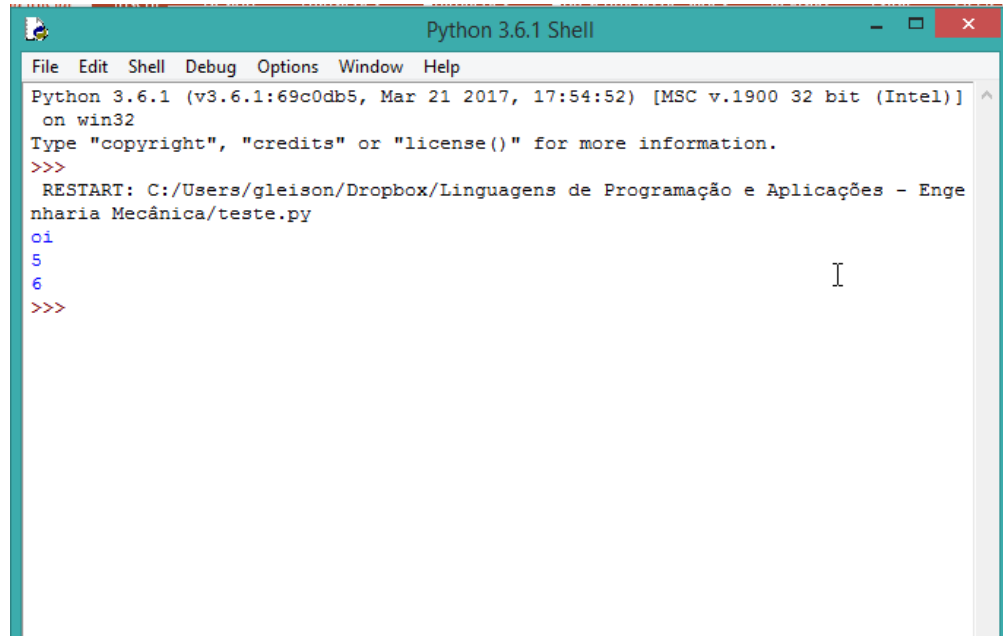


The image shows a screenshot of a Python IDE window. The title bar reads "teste.py - C:/Users/gleison/Dropbox/Linguagens de Programação e Aplicações - Engenharia Mec...". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following text:

```
print('oi')  
  
1 + 2 * 2  
  
print(1 + 2 * 2)  
  
print((1 + 2) * 2)  
|
```

The status bar at the bottom right indicates "Ln: 8 Col: 0".

Editando um arquivo



The image shows a screenshot of a Python 3.6.1 Shell window. The window has a teal title bar with the text "Python 3.6.1 Shell" and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content:

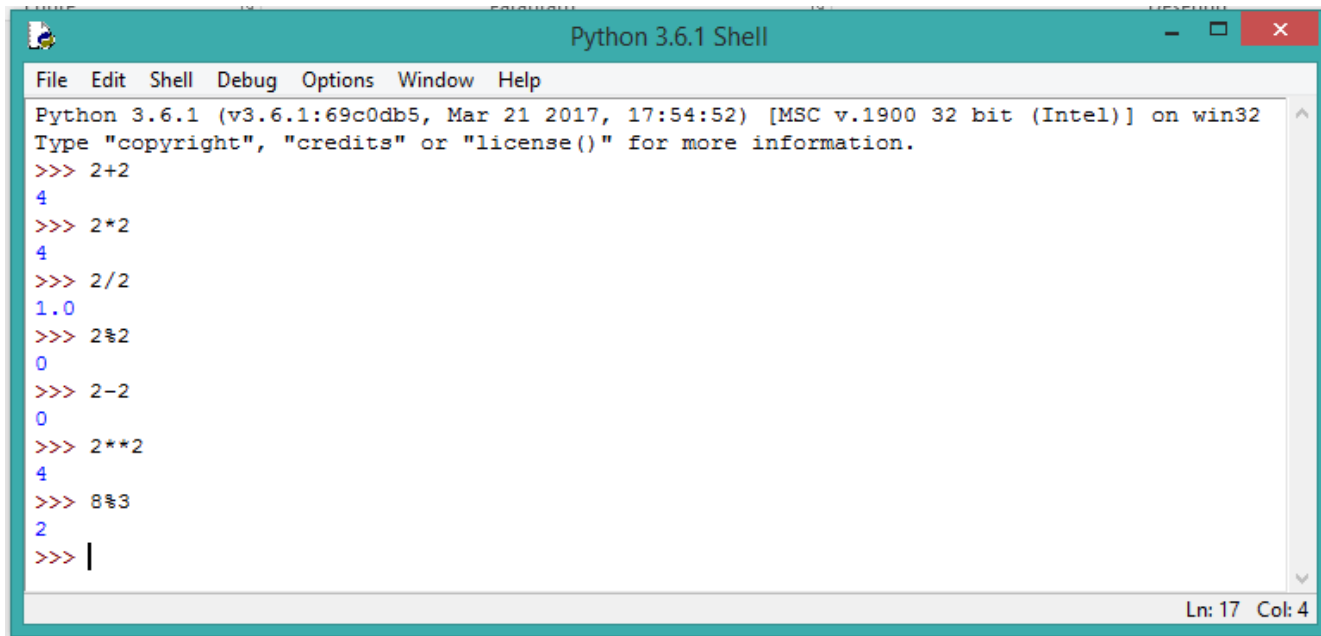
```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
  RESTART: C:/Users/gleison/Dropbox/Linguagens de Programação e Aplicações - Enge
nharia Mecânica/teste.py
01
5
6
>>>
```

A cursor is visible on the line containing the number "6".

Erros Comuns

- Letras maiúsculas e minúsculas são diferentes;
 - Ex: `print` e `Print`
- Aspas devem ser abertas e fechadas;
- Parênteses não são opcionais em Python;
- Espaços são muito importantes.

Operações no Interpretador Python



The image shows a screenshot of the Python 3.6.1 Shell window. The window has a teal title bar with the text "Python 3.6.1 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window is a text editor with a white background and a vertical scrollbar on the right. It contains 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.
>>> 2+2
4
>>> 2*2
4
>>> 2/2
1.0
>>> 2%2
0
>>> 2-2
0
>>> 2**2
4
>>> 8%3
2
>>> |
```

At the bottom right of the window, there is a status bar that reads "Ln: 17 Col: 4".

Operações no Interpretador Python

- Adição (+)
- Subtração (-)
- Multiplicação (*)
- Divisão (/) diferente de \
- Resto da divisão (%)
- Exponenciação ou potenciação (**)

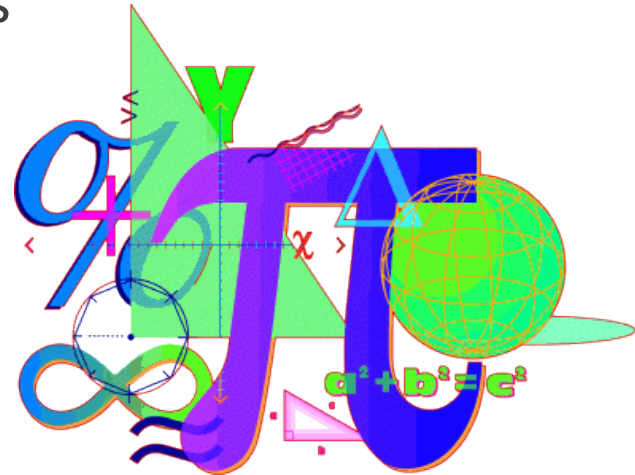
Parênteses

- Alterar ordem de execução/prioridade.
- $100 + (200 * 300 / 200) = 100 + \left(\frac{200*300}{200}\right)$
- $2 * 2 + 10 \Rightarrow$ resultado: 14
- $2 * (2 + 10) \Rightarrow$ resultado: 24

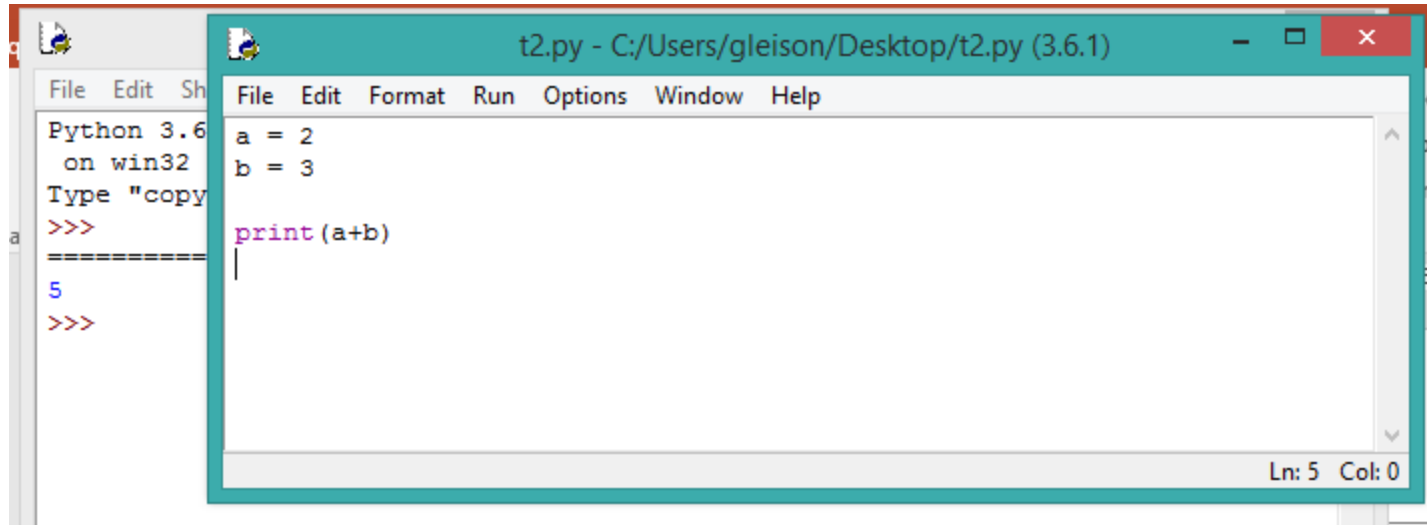
Variáveis e entrada de dados

Conceito

- Representar incógnitas em uma equação do tipo $x + 1 = 2$, onde devemos determinar o valor de x ;
- Em programação variáveis são utilizadas para armazenar valores e para dar nome a uma área de memória;



Exemplo



The image shows a screenshot of a Python 3.6 IDE. The main window, titled "t2.py - C:/Users/gleison/Desktop/t2.py (3.6.1)", contains the following code:

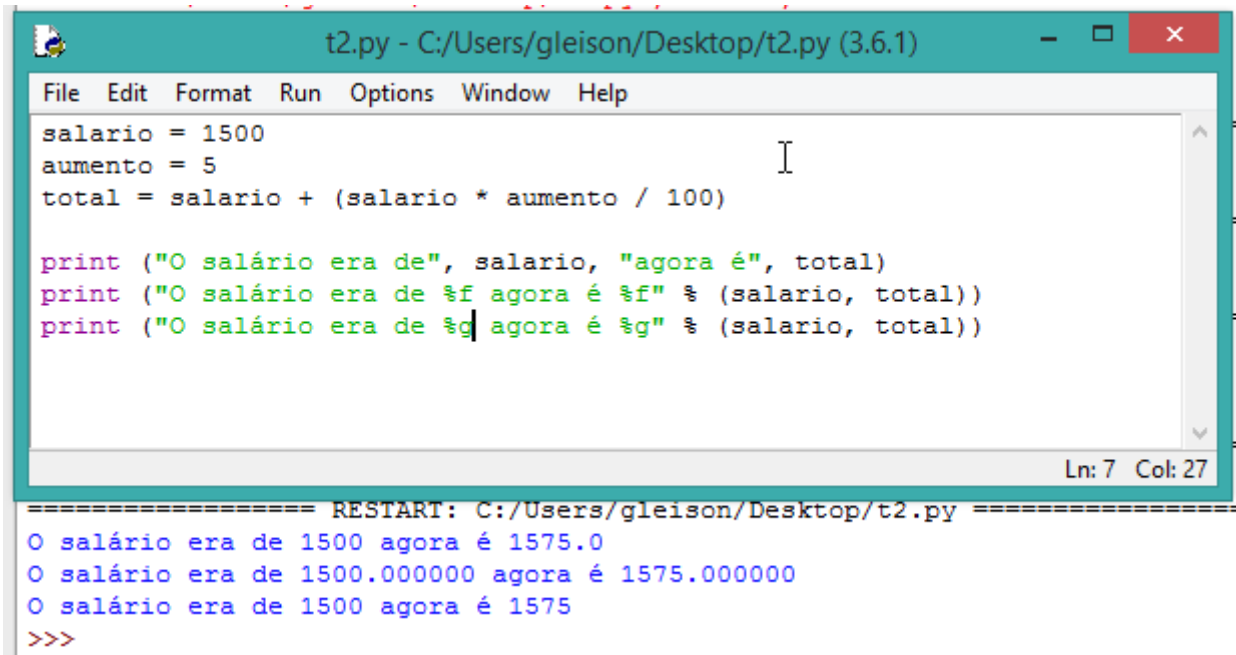
```
File Edit Format Run Options Window Help
a = 2
b = 3
print(a+b)
```

The status bar at the bottom right of the editor window indicates "Ln: 5 Col: 0".

To the left, a smaller window shows the Python 3.6 shell output:

```
Python 3.6
on win32
Type "copy"
>>>
=====
5
>>>
```


Exemplo 2



The screenshot shows a Python IDE window titled "t2.py - C:/Users/gleison/Desktop/t2.py (3.6.1)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The editor contains the following Python code:

```
salario = 1500
aumento = 5
total = salario + (salario * aumento / 100)

print ("O salário era de", salario, "agora é", total)
print ("O salário era de %f agora é %f" % (salario, total))
print ("O salário era de %g agora é %g" % (salario, total))
```

The status bar at the bottom right of the editor indicates "Ln: 7 Col: 27". Below the editor, the output of the script is displayed, preceded by a restart command:

```
===== RESTART: C:/Users/gleison/Desktop/t2.py =====
O salário era de 1500 agora é 1575.0
O salário era de 1500.000000 agora é 1575.000000
O salário era de 1500 agora é 1575
>>>
```

Nomes de variáveis

Tabela 3.1 – Exemplo de nomes válidos e inválidos para variáveis

Nome	Válido	Comentários
a1	Sim	Embora contenha um número, o nome a1 inicia com letra.
velocidade	Sim	Nome formado por letras.
velocidade90	Sim	Nome formado por letras e números, mas iniciado por letra.
salário_médio	Sim	O símbolo sublinha (<code>_</code>) é permitido e facilita a leitura de nomes grandes.
salário médio	Não	Nomes de variáveis não podem conter espaços em branco.
b	Sim	O sublinha (<code></code>) é aceito em nomes de variáveis, mesmo no início.
1a	Não	Nomes de variáveis não podem começar com números.

Tipos de variáveis

- Em Python existem vários tipos de dados, mas os mais comuns são números inteiros, números de ponto flutuante e strings.
- Além de poder armazenar números e letras, as variáveis em Python também podem armazenar valores como verdadeiro(true) ou falso(false). Esses tipos são conhecidos como lógicos.

Variáveis numéricas

- Dizemos que uma variável é numérica quando armazena números inteiros ou de ponto flutuante;
- Os números inteiros são aqueles sem parte decimal, como: 1, 0, -5, 550, -47, 300000;
- Números de ponto flutuante ou decimais são aqueles com parte decimal, como 1.0, 5.478, 30000.4

Variáveis do tipo Lógico

- Quando se deseja armazenar valores do tipo verdadeiro ou falso;
- Chamado de tipo lógico ou booleano;
- Em Python **True** e **False**;
- Ex: resultado = True
aprovado = False

Operadores Relacionais

Tabela 3.2 – Operadores relacionais

Operador	Operação	Símbolo matemático
==	igualdade	=
>	maior que	>
<	menor que	<
!=	diferente	≠
>=	maior ou igual	≥
<=	menor ou igual	≤

```
>>> 1 == 1
True
>>> 1 == 2
False
>>> 1 >= 2
False
>>> 1 <= 2
True
>>> 1 != 2
True
>>> 1 != 1
False
```

Operadores Lógicos

Tabela 3.3 – Operadores lógicos

Operador Python	Operação
not	não
and	e
or	ou

Operadores Lógicos - NOT

Tabela 3.4 – Tabela verdade do operador not (não)

V_1	$\text{not } V_1$
V	F
F	V

Operadores Lógicos - AND

Tabela 3.5 – Tabela verdade do operador and (e)

V_1	V_2	$V_1 \text{ and } V_2$
V	V	V
V	F	F
F	V	F
F	F	F

Operadores Lógicos - OR

Tabela 3.6 – Tabela verdade do operador or(ou)

V_1	V_2	$V_1 \text{ or } V_2$
V	V	V
V	F	V
F	V	V
F	F	F

Expressões Lógicas

- Os operadores lógicos podem ser combinados em expressões lógicas mais complexas;
- Quando um expressão possuir mais de um operador lógico, eles são avaliados na seguinte ordem:
 1. NOT
 2. AND
 3. OR

Expressões Lógicas

- Operadores relacionais podem ser utilizados em expressões com operadores lógicos;

```
>>> True or False and not True
True
>>> 2 > 1 and 2 < 3
True
>>> 2 >= 1 and 2 <= 2
True
>>> 1 == 1 or not 1 == 1
True
.
```

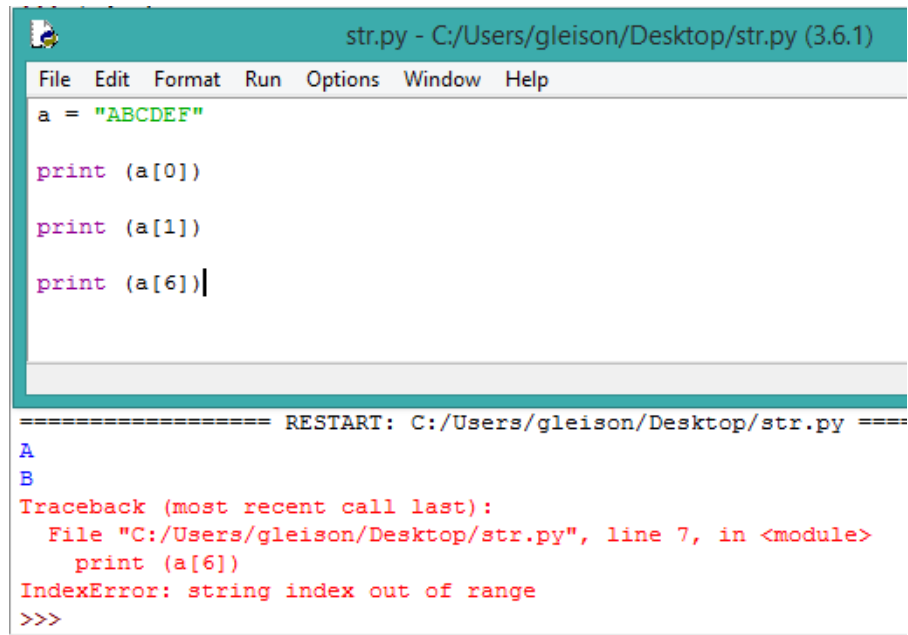
Variáveis String

- Armazenam uma cadeia de caracteres;



Figura 3.2 – Índices e conteúdo de uma variável string.

Exemplo



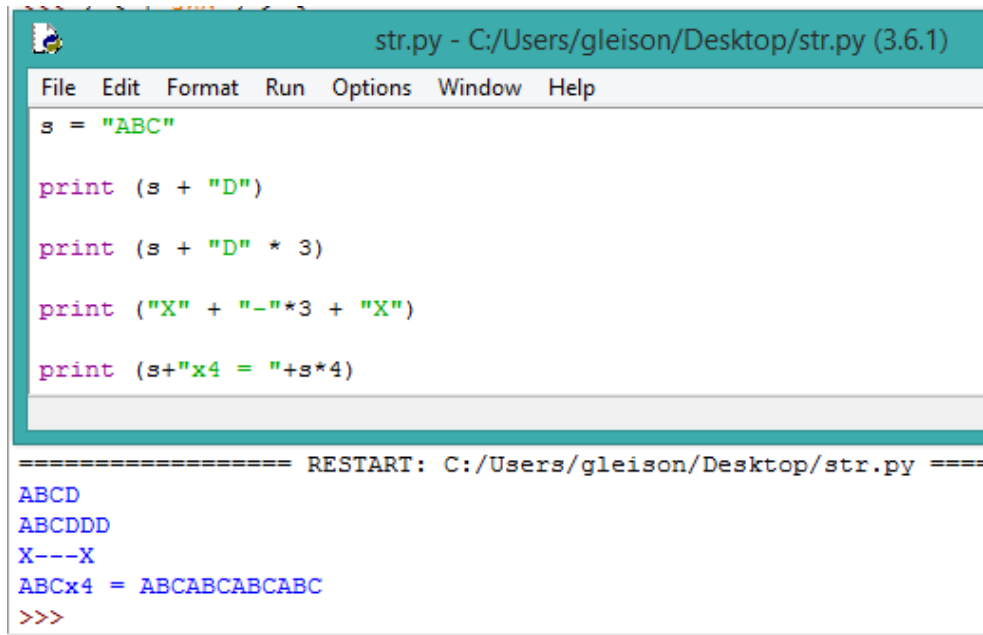
```
str.py - C:/Users/gleison/Desktop/str.py (3.6.1)
File Edit Format Run Options Window Help
a = "ABCDEF"
print (a[0])
print (a[1])
print (a[6])|

===== RESTART: C:/Users/gleison/Desktop/str.py =====
A
B
Traceback (most recent call last):
  File "C:/Users/gleison/Desktop/str.py", line 7, in <module>
    print (a[6])
IndexError: string index out of range
>>>
```

Algumas Operações com String

- Concatenação;
- Composição;
- Fatiamento;

Concatenação



```
str.py - C:/Users/gleison/Desktop/str.py (3.6.1)
File Edit Format Run Options Window Help
s = "ABC"

print (s + "D")

print (s + "D" * 3)

print ("X" + "-"*3 + "X")

print (s+"x4 = "+s*4)

===== RESTART: C:/Users/gleison/Desktop/str.py =====
ABCD
ABCD
ABCD
ABCD
X---X
ABCDx4 = ABCABCABCABC
>>>
```


Composição

- Usando composições em Python podemos escrever de forma simples e clara;

```
x = 18  
print ("O Gleison tem %d anos." % x)
```

```
O Gleison tem 18 anos.  
>>>
```

Composição

- Python suporta diversos tipos de marcadores, abaixo são mostrados os principais.

Tabela 3.7 – Marcadores

Marcador	Tipo
%d	Números inteiros
%s	Strings
%f	Números decimais

Exemplo 2

- Python suporta diversos tipos de marcadores, abaixo são mostrados os principais.

► Listagem 3.10 – Exemplo de composição com marcadores

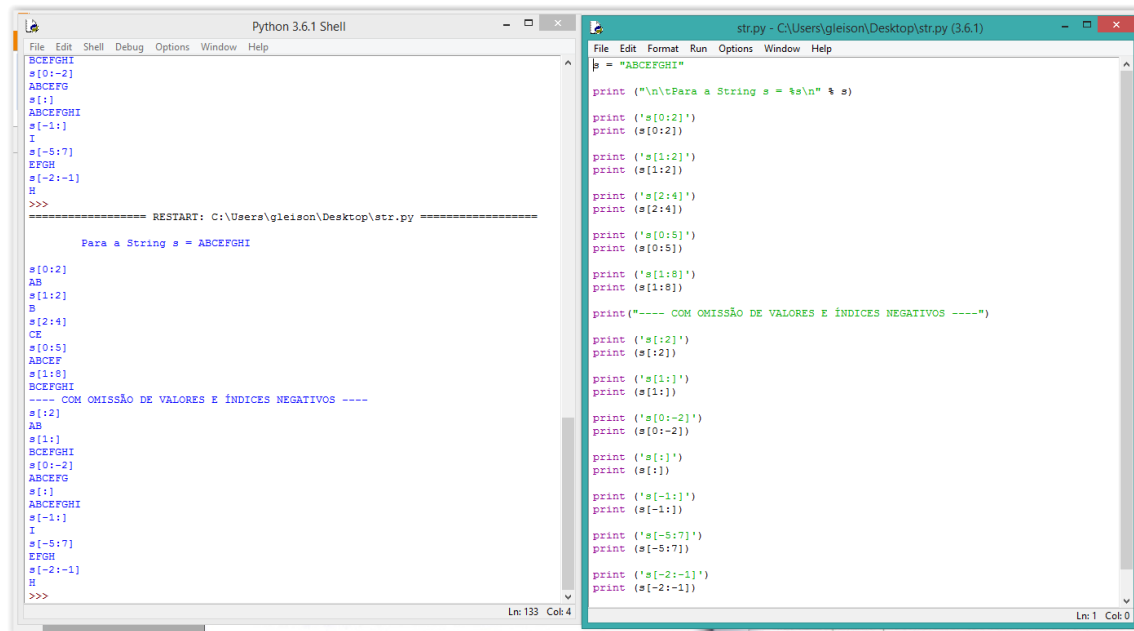
```
>>> idade = 22
>>> print("[%d]" % idade)
[22]
>>> print("[%03d]" % idade)
[022]
>>> print("[%3d]" % idade)
[ 22]
>>> print("[% -3d]" % idade)
[22 ]
```

Fatiamento

- Para string `s = "ABCDEFGHI"`

S	A	B	C	D	E	F	G	H	I
índice	0	1	2	3	4	5	6	7	8
	-9	-8	-7	-6	-5	-4	-3	-2	-1

Fatiamento



The image displays two side-by-side screenshots of a Python 3.6.1 environment. The left window, titled 'Python 3.6.1 Shell', shows a series of string slicing operations on the string 'BCEFGHI'. The right window, titled 'str.py - C:\Users\gleison\Desktop\str.py (3.6.1)', shows a Python script that prints the results of these slicing operations, including examples with negative indices and omissions.

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
s="BCEFGHI"
s[0:-2]
ABCEFG
s[1:]
ABCEFGHI
s[-1:]
I
s[-5:7]
EFGH
s[-2:-1]
H
>>>
===== RESTART: C:\Users\gleison\Desktop\str.py =====
Para a String s = ABCEFGHI

s[0:2]
AB
s[1:2]
B
s[2:4]
CE
s[0:5]
ABCEF
s[1:8]
BCEFGHI
----- COM OMISSÃO DE VALORES E ÍNDICES NEGATIVOS -----
s[1:2]
AB
s[1:]
BCEFGHI
s[0:-2]
ABCEFG
s[1]
ABCEFGHI
s[-1:]
I
s[-5:7]
EFGH
s[-2:-1]
H
>>>
```

```
str.py - C:\Users\gleison\Desktop\str.py (3.6.1)
File Edit Format Run Options Window Help
s = "ABCEFGHI"
print ("\n\tPara a String s = %s\n" % s)
print ('s[0:2]')
print (s[0:2])
print ('s[1:2]')
print (s[1:2])
print ('s[2:4]')
print (s[2:4])
print ('s[0:5]')
print (s[0:5])
print ('s[1:8]')
print (s[1:8])
print ("----- COM OMISSÃO DE VALORES E ÍNDICES NEGATIVOS -----")
print ('s[:2]')
print (s[:2])
print ('s[1:]')
print (s[1:])
print ('s[0:-2]')
print (s[0:-2])
print ('s[1:]')
print (s[1:])
print ('s[-1:]')
print (s[-1:])
print ('s[-5:7]')
print (s[-5:7])
print ('s[-2:-1]')
print (s[-2:-1])
```

Entrada de dados

- A função **input** é utilizada para solicitar dados do usuário.

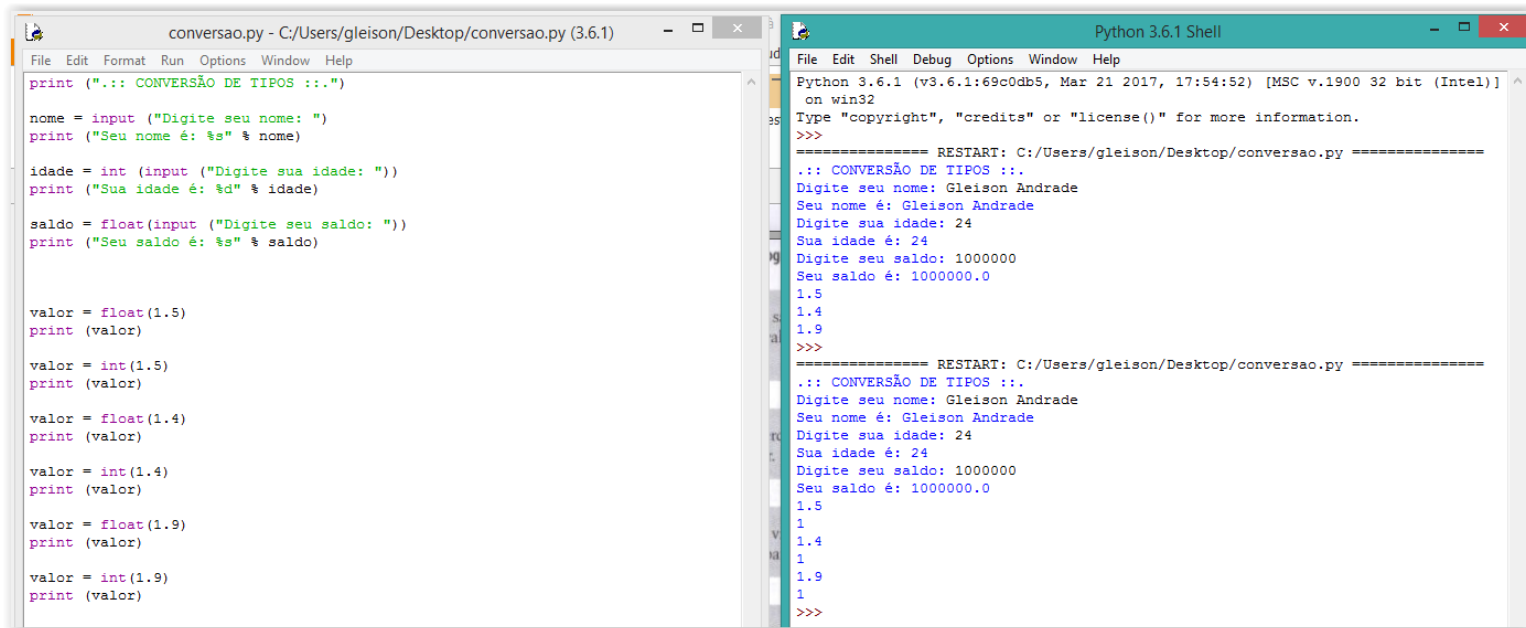
► Listagem 3.16 – Entrada de dados

```
x = input("Digite um número: ")  
print(x)
```

► Listagem 3.17 – Saída na tela, tendo o 5 como exemplo de número digitado pelo usuário

```
Digite um número: 5  
5
```

Conversão da entrada de dados



The image shows a Python script named `conversao.py` and its execution in a Python 3.6.1 Shell. The script is designed to demonstrate data type conversion in Python.

Script Content (conversao.py):

```
print ("... CONVERSÃO DE TIPOS ...")

nome = input ("Digite seu nome: ")
print ("Seu nome é: %s" % nome)

idade = int (input ("Digite sua idade: "))
print ("Sua idade é: %d" % idade)

saldo = float(input ("Digite seu saldo: "))
print ("Seu saldo é: %s" % saldo)

valor = float(1.5)
print (valor)

valor = int(1.5)
print (valor)

valor = float(1.4)
print (valor)

valor = int(1.4)
print (valor)

valor = float(1.9)
print (valor)

valor = int(1.9)
print (valor)
```

Shell Execution Output:

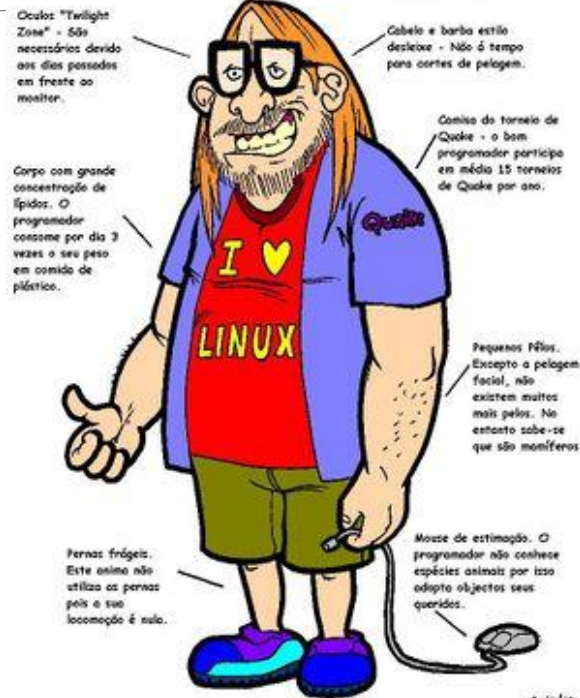
```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/gleison/Desktop/conversao.py =====
... CONVERSÃO DE TIPOS ...
Digite seu nome: Gleison Andrade
Seu nome é: Gleison Andrade
Digite sua idade: 24
Sua idade é: 24
Digite seu saldo: 1000000
Seu saldo é: 1000000.0
1.5
1.4
1.9
>>>
===== RESTART: C:/Users/gleison/Desktop/conversao.py =====
... CONVERSÃO DE TIPOS ...
Digite seu nome: Gleison Andrade
Seu nome é: Gleison Andrade
Digite sua idade: 24
Sua idade é: 24
Digite seu saldo: 1000000
Seu saldo é: 1000000.0
1.5
1
1.4
1
1.9
1
>>>
```

Agora falta pouco para você se tornar um Programador..



ANATOMIA DO PROGRAMADOR

Informaticus programatorum



Reinaldo 20/5/03