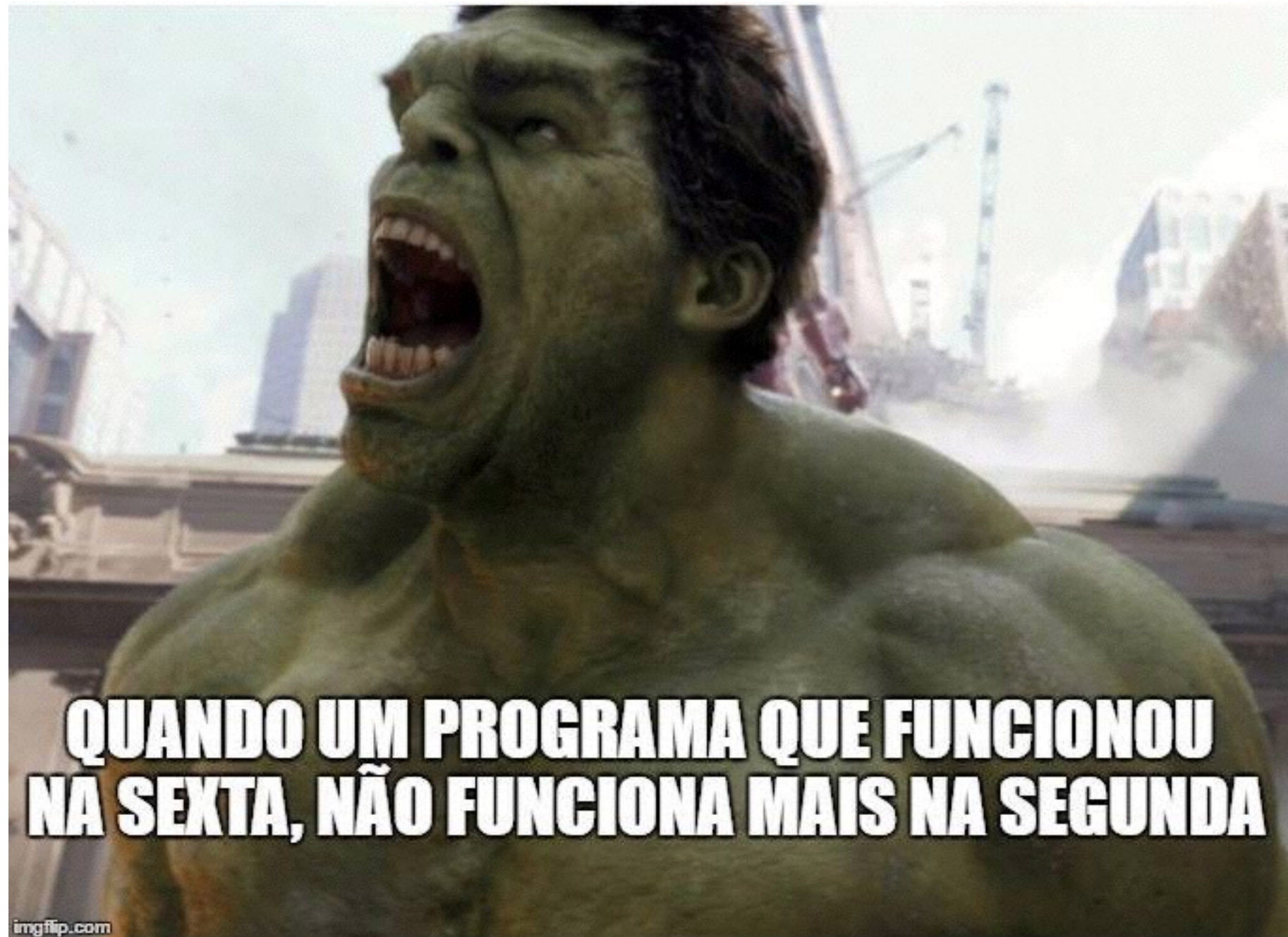


Variáveis e Entrada de Dados

Módulo IV



Nome de variáveis

- Toda variável é obrigatório iniciar com LETRA ou _.
- Por gentileza, não utilizar variáveis com acento (Pode dar problemas com Encoding)
- PALAVRAS RESERVADAS: 'False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield'

Palavras Reservadas

- Vá no terminal e digite:
 - `import keyword`
 - `print (keyword.kwlist)`



Vamos praticar no modo interativo



Variáveis Numéricas

- Quando se armazena um numero inteiro ou ponto flutuante, dizemos que é uma variável numérica.
 - Variável inteiro: 66
 - Variável Float: 70.534
 - Variável Complexa: 1j

Dica do _

- Sempre que quiser utilizar o resultado anterior imprimido no terminal, use o _.
- Ou seja, neste “_” está armazenado o ultimo valor exibido.

Built-ins Functions (Funções Embutidas) do Python

- <https://docs.python.org/3.1/library/functions.html>
- <http://www.programiz.com/python-programming/built-in-function>

Round, Int, Float

- Round (Arredondar):

```
>>> a = 1.5
>>> a
1.5
>>> round(a)
2
```
- Transforma o numero em inteiro:

```
>>> a = 1.5
>>> type (a)
<class 'float'>
>>> a_novo = int(a)
>>> type (a_novo)
<class 'int'>
>>> a_novo
1
```
- Transforma o numero de float para inteiro:

```
>>> a = 1
>>> type (a)
<class 'int'>
>>> float (a)
1.0
```

Operadores Aritméticos

Operador	Ação
-	Subtração, também menos unário
+	Adição
*	Multiplicação
/	Divisão
%	Módulo da divisão (resto)
--	Decremento
++	Incremento

** Potenciação

Variáveis Lógicas

- Verdadeiro (True)
- Falso (False)
- Sempre escreva a primeira letra em maiúsculo.

Operadores Lógicos

Símbolo	Nome do Operador	Exemplo	Significado
>	Maior que	$x > y$	x é maior que y?
>=	Maior ou igual	$x >= y$	x é maior ou igual a y ?
<	Menor que	$x < y$	x é menor que y?
<=	Menor ou igual	$x <= y$	x é menor ou igual a y ?
==	Igualdade	$x == y$	x é igual a y?
!=	Diferente de	$x != y$	x é diferente de y?

Operadores Lógicos

- not (não)
- and (e)
- or (ou)

Tô bugando!



Variáveis String

- Cadeia de caracteres (sequencia de símbolos como letras, numero, sinais e etc)
- Imagine que uma variável string seja uma sequencia de bloco como abaixo:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
U	N	A		B	O	M		D	E	S	P	A	C	H	O

Variáveis String

- Essa tabelinha equivale a String: “Una Bom Despacho”
- Os espaços dentro das aspas são contados também como um caracter.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
U	N	A		B	O	M		D	E	S	P	A	C	H	O

Operações com String

- Sabe o que é legal? Que podemos manipular os blocos da string, utilizando colchetes [] e colocando a posição que queremos acessar ou delimitar.
- Built-in function **len()**, mostra a quantidade de caracteres existe na string, ou em uma lista.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
U	N	A		B	O	M		D	E	S	P	A	C	H	O

Operações com String

- **Concatenação**
- **Composição:**
 - **Números inteiros: %i**
 - **Strings: %s**
 - **Números decimais: %f**

```
print ("O %s tem %d anos e possui APENAS R$%0.2f reais na carteira" %(nome, idade, dinheiro))
```

Concatenação

- **Concatenação**

```
>>> texto = ("LEO" + " GENTE" + " BOA" + "\n") * 10
>>> print (texto)
LEO GENTE BOA
LEO GENTE BOA
LEO GENTE BOA
LEO GENTE BOA
LEO GENTE BOA
LEO GENTE BOA
LEO GENTE BOA
LEO GENTE BOA
LEO GENTE BOA
LEO GENTE BOA
```

Composição

- Utilizando o format:

```
print ("O {} tem {} anos e possui APENAS R${:0.2f}  
reais na carteira".format(nome, idade, dinheiro))
```


Fatiamento

- Utilize os colchetes [posicaoini:posicaoofim]
- **Importante:** O indice da posicao do fim, não está incluído, ou seja, a string “Hello”, se fizer o fatiamento de [0:2] vai aparecer apenas **He** e não **Hel**
- **Terceira posição:** pular de quanto em quanto.

H	e	l	l	o
0	1	2	3	4
-5	-4	-3	-2	-1

Corte

- Built-in function **STRIP()**

```
>>> mensagem = "                ???leopardocouy@hotmail.com?????"
>>> mensagem
'                ???leopardocouy@hotmail.com?????                \t\t\t'
>>> len(mensagem)
69
>>> mensagem.strip()
'????leopardocouy@hotmail.com?????'
>>> mensagem
'                ???leopardocouy@hotmail.com?????                \t\t\t'
>>> mensagem = mensagem.strip() # remove os espacos e as tabulacoes
>>> mensagem
'????leopardocouy@hotmail.com?????'
>>> mensagem = mensagem.strip('?')
>>> mensagem
'leopardocouy@hotmail.com'
```

Dividir

- Built-in function **Split()**

```
>>> mensagem = mensagem.strip('?')
>>> mensagem
'leonardocouy@hotmail.com'
>>> mensagem.split('@')
['leonardocouy', 'hotmail.com']
>>> mensagem = mensagem.split('@')
>>> print(mensagem)
['leonardocouy', 'hotmail.com']
>>> print(mensagem[0])
leonardocouy
>>>
```

Substituição

- Built-in function **replace()**

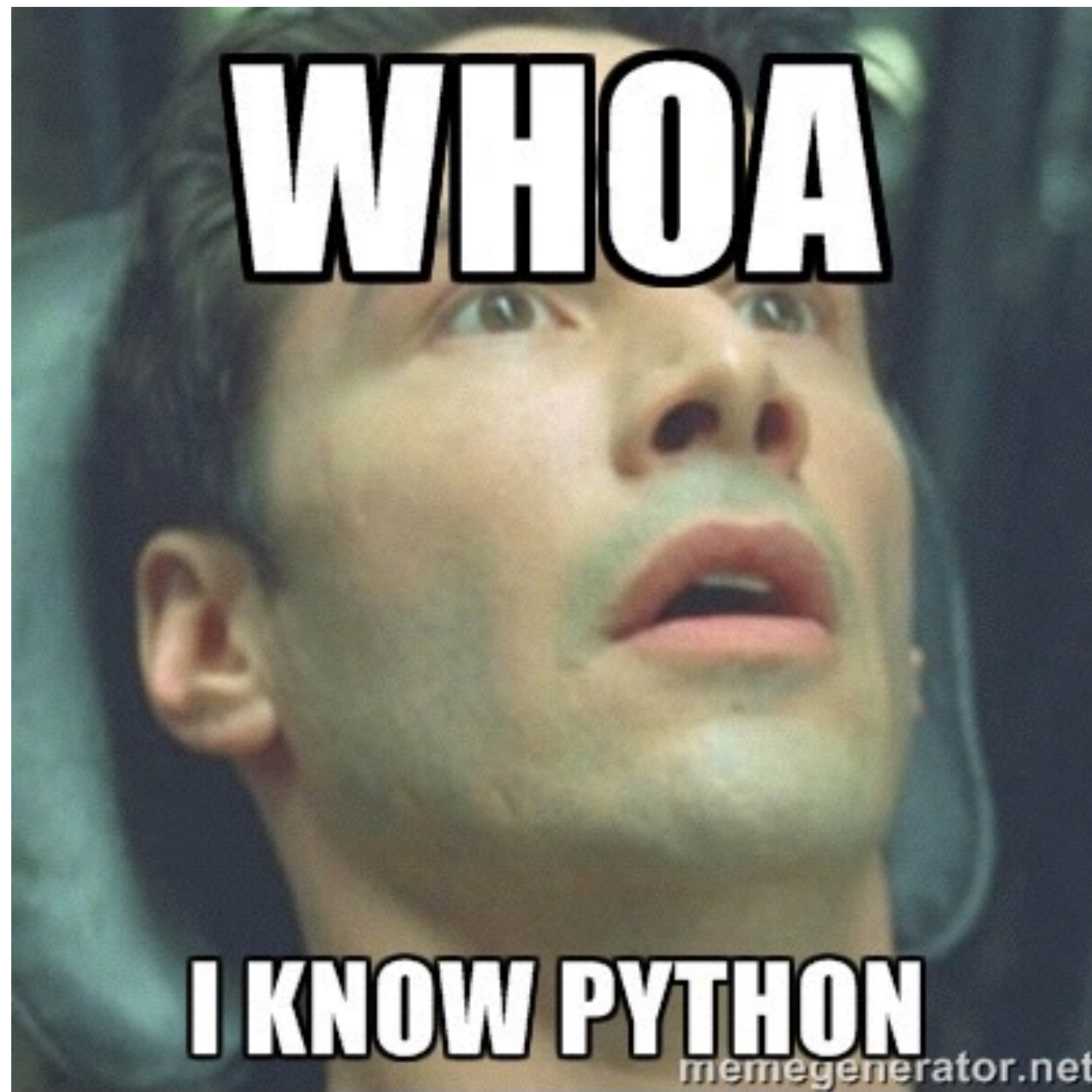
```
>>> frase = "leo é muito gente boa"
>>> frase.replace("é", "não é")
'leo não é muito gente boa'
```

Entrada de dados

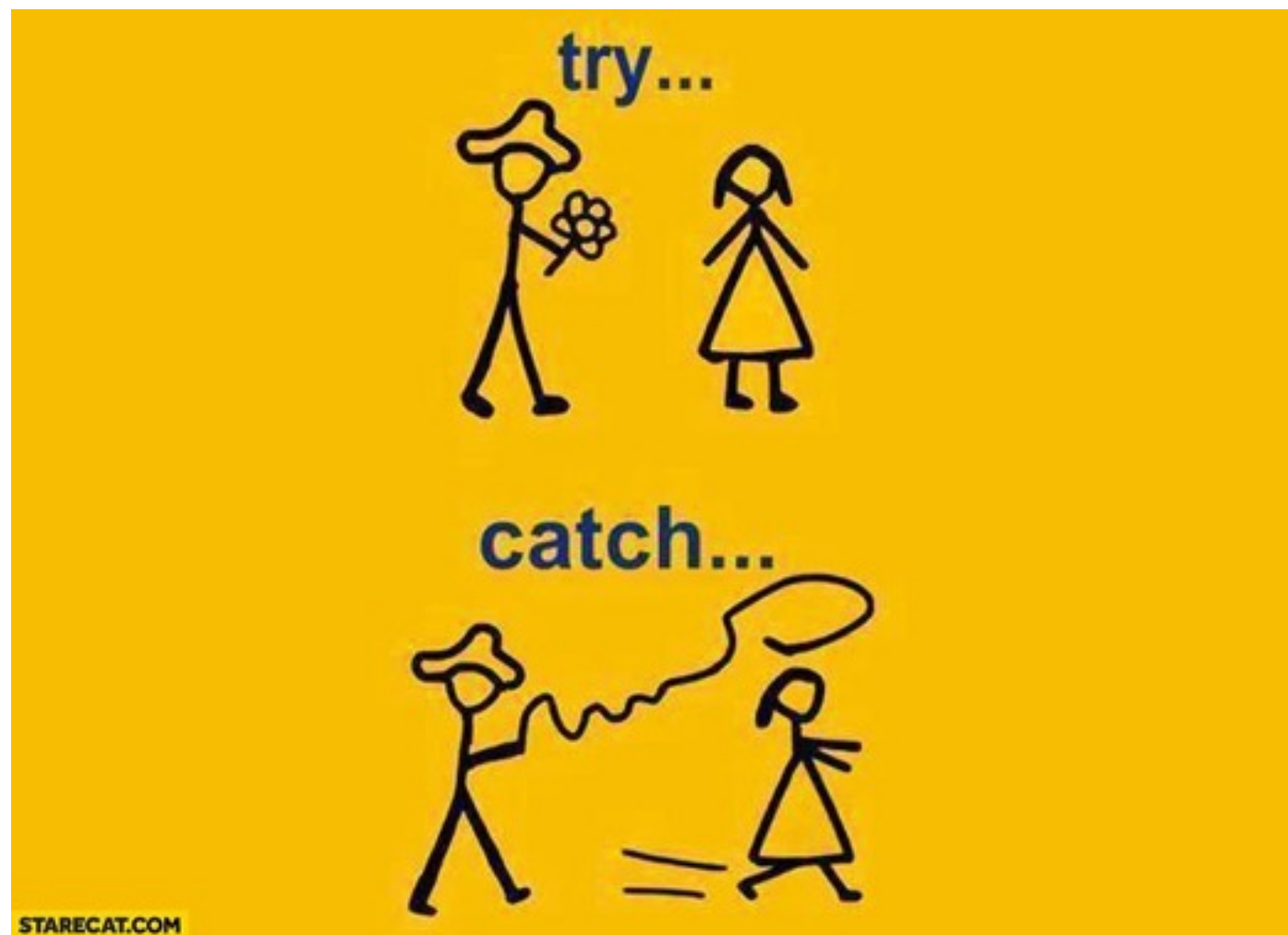
- Built-in function **input()**

- **Exemplo:**

```
>>> idade = int(input("Digite sua idade: "))
Digite sua idade: 21
>>> print ("Sua idade é: {}".format(idade))
Sua idade é: 21
```



O famoso try, catch



O famoso try, catch

```
>>> idade = int(input("Digite sua idade: "))
Digite sua idade: a
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'a'
>>>
```

Tratando erros

```
>>> try:
...     idade = int (input("Digite sua idade: "))
...     print ("Sua idade é: {}".format(idade))
... except ValueError:
...     print ("Ops, verifique se digitou um número válido")
...
Digite sua idade: 21
Sua idade é: 21
```

```
>>> try:
...     idade = int (input("Digite sua idade: "))
...     print ("Sua idade é: {}".format(idade))
... except ValueError:
...     print ("Ops, verifique se digitou um número válido")
...
Digite sua idade: a
Ops, verifique se digitou um número válido
```

Provocar erro

```
>>> try:
...     idade = int (input("Digite sua idade: "))
... except ValueError:
...     raise ValueError("Ops, verifique se digitou um número válido")
...
```

...

Digite sua idade: a

Traceback (most recent call last):

File "<stdin>", line 2, in <module>

ValueError: invalid literal **for** int() with base 10: 'a'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

File "<stdin>", line 4, in <module>

ValueError: Ops, verifique se digitou um número válido

Homework

1. Leia o nome, sobrenome e depois concatene o nome e o sobrenome, e mostre o nome completo.
2. Leia uma temperatura em graus **Celsius** e converta para graus **Fahrenheit**. *Formula: $F = (9 * C + 160) / 5$*
3. Retorne apenas a palavra "leo" do texto: "se tem um cara gente boa, eh esse tal do leo"
4. Leia um nome e imprima ele invertido
5. Divida a string de lista de emails em uma lista de emails em python: lista_de_emails =
"leonardocouy@hotmail.com,carlos@hotmail.com,kennedy@hotmail.com,carinhaquemoralogoali@hotmail.com,genteboa@hotmail.com"

Homework

6. Agora é com você, imagine algo que você possa fazer com que aprendeu e tente!
7. MANDA TUDO PRO GITHUB!