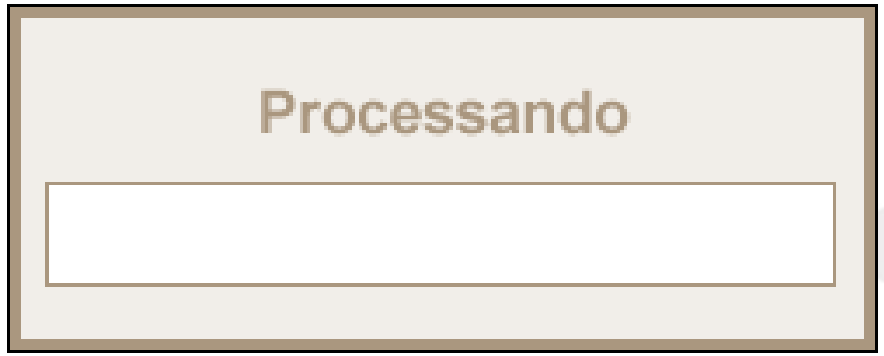


Usando o Python com CODESKULPTOR

Fonte: http://viladosilicio.com.br/wp-content/uploads/2016/08/Post1_pt2.png

Ms. Jacqueline Felix da Silva – jacfel@gmail.com

Identificadores...

De caracteres

Ex:
Corretos => minha_classe, classe1, _nome
Errados => minha classe, 1classe, !val

- Maiúsculas ≠ Minúsculas
- Começar com letras(a..z) ou sublinhado (_)
- Não pode haver espaços em branco, nem caracteres especiais. Pode conter números (0..9)

Obs.: De preferência criar variáveis com letras **minúsculas**, isso inclui nomes compostos.

Nome das classes

- é recomendado iniciar com letras maiúsculas
- nomes compostos iniciando com letras maiúsculas

Ex.: **L**etra**M**aiúscula
Nome**D**a**C**lasse

Exercício:
Escreva um algoritmo para ler dois números (A e B) e trocar os seus valores. Exibir os valores de A e B após a troca.

De Constantes

- Letras **Maiúsculas**
- Utilizar “_ “ em nomes compostos.

Ex.: PI
VALOR_MAXIMO
JUROS
PRESTACAO

Identificadores...

De caracteres

Ex:
Corretos => minha_classe, classe1, _nome
Errados => minha classe, 1classe, !val

- Maiúsculas ≠ Minúsculas
- Começar com letras(a..z) ou sublinhado (_)
- Não pode haver espaços em branco, nem caracteres especiais. Pode conter números (0..9)

Obs.: De preferência criar variáveis com letras **minúsculas**, isso inclui nomes compostos.

Nome das classes

- é recomendado iniciar com letras maiúsculas
- nomes compostos iniciando com letras maiúsculas

Ex.: Letra**M**aiúscula
Nome**D**a**C**lasse

Exercício:
Escreva um algoritmo para ler dois números (A e B) e trocar os seus valores. Exibir os valores de A e B após a troca.

```
>>> #Trocar valores utilizando uma variável
...
>>> a=20
>>> b=35
>>> aux=a
>>> a=b
>>> b=aux
>>> print('a=',a,'','b=',b)
```

Variável String

Variável do tipo String armazenam cadeias de caracteres como nomes e textos em geral. Chamamos **cadeia de caracteres** uma sequência de símbolos como letras, números, sinais de pontuação, etc. Exemplo: João come pão. Nesse caso, João é uma sequência com as letras, J, o, ã, o. Para simplificar o texto, utilizaremos o nome **string** para mencionar cadeias de caracteres. Podemos imaginar uma string como uma sequência de blocos, onde cada letra, número ou espaço em branco ocupa uma posição como mostra a figura abaixo.

STRING												
J	o	ã	o		c	o	m	e		p	ã	o

Função len

Uma string em Python tem um tamanho associado, assim como um conteúdo que pode ser acessado caractere a caractere. O tamanho de uma string pode ser obtido utilizando-se a função **len**.

A função **len** retorna um valor do tipo inteiro, representando a quantidade de caracteres contidos na string.

Obs.: Se a string é vazia (representada por “”, ou seja, duas aspas sem nada entre elas, nem mesmo espaço em branco), seu tamanho é igual a zero.

Ex.:

```
print (len(“A”))
print (len(“AB”))
print(len(“”))
print(len(“O rato roeu a roupa”))
```

Outra característica de string

É possível acessar caractere a caractere utilizando um número inteiro para representar sua posição. Esse número é chamado de índice, e começamos a contar do zero. Isso quer dizer que o 1º caracter de uma string é de posição ou índice 0.

índice

Ex:

	0	1	2	3	4	5	6	7	8	9
a=	A	B	C	D	E	F	G	H	I	J

a= (“ABCDEFGHIJ”)

print(a[0])

print(a[5])

print(a[10])

print(len(a)])

Outra característica de string

É possível acessar caractere a caractere utilizando um número inteiro para representar sua posição. Esse número é chamado de índice, e começamos a contar do zero. Isso quer dizer que o 1º caracter de uma string é de posição ou índice 0.

Ex:

índice

0	1	2	3	4	5	6	7	8	9
Ovos	Banana	Abacate	Arroz	Feijão	Abóbora	Café	Manteiga	Leite	Iogurte

lista=[“ovos”,”banana”, ”abacate”, “arroz”, “feijão”, “abóbora”, “café”, “manteiga”, “leite”, “iogurte”]

```
print(len(lista))
print(lista[0])
print(lista[10])
print(lista[9])
```

Operações com Strings

Em Python, as variáveis do tipo string suportam operações como **concatenação, composição e fatiamento**. Veremos como funcionam algumas dessas operações.

Concatenação

É possível concatenar o conteúdo de variáveis **string** utilizando o operador de adição +.

```
s = "ABC"
print (s + "D")
```

Uma outra possibilidade para casos de concatenação de uma **string** várias vezes é usar o operador de multiplicação

```
print (s + "E" * 4)
```

Composição

Algumas vezes é necessário compor mensagens com várias informações, por exemplo, se tivermos que exibir uma mensagem que mostre as vezes que um determinado usuário acessou determinada seção do site.

Então teremos as seguintes variáveis **usuario**, **quantidade_de_acesso**, **secao**, e as informações guardadas nelas serão usadas para compor a seguinte mensagem: “O usuário **Ivan Costa** acessou **10** vezes a seção de **Esporte**.

Bem, se formos fazer isso através da concatenação, não será muito prático. Uma opção é usar a composição de strings do Python, uma forma bem mais simples e clara:

```
nome=(“Ivan Costa")
quantidade_de_acesso=10
seção=("Esportes")

print("O usuario %s acessou %i vezes a seção de %s“  %(nome, quantidade_de_acesso, secao))
```

Recapitulando...

O Python é uma linguagem de **tipagem dinâmica**, ou seja, o tipo de cada variável será definido de forma dinâmica sem precisar declarar explicitamente qual é o tipo de variável.

Além disso, o Python suporta diversas operações com **marcadores**.

Marcador	Tipo
%i	inteiro
%d	decimal
%s	string
%f	float

Obs.: No lugar do marcador deve ser colocado um determinado valor.

Ex.:
```
nome, idade = (“Rafael", 26)
print ("Nome: %s\nIdade:%i" % (nome, idade))     #Vai pular uma linha  \n
```

Mais exemplos...

```
nome=input("Digite um nome: ")
quantidade_de_acesso=int(input("Quantidade de acessos: "))
secao=input("Informe qual secao lida no jornal: ")

print("O(a) usuario(a) %s acessou %i vezes a secao de %s" %(nome, quantidade_de_acesso, secao))
```

DESAFIO

Refaça o programa acima, utilizando marcadores na estrutura de repetição WHILE. Defina a condição de saída.

```
cont=1
num=int(input("Digite um número: "))
while cont<=num:
    nome=input("Digite um nome: ")
    quantidade_de_acesso=int(input("Quantidade de acessos: "))
    secao=input("Informe qual seção lida no jornal: ")
    print("O(a) usuario(a) %s acessou %i vezes a seção de %s“ %(nome, quantidade_de_acesso, secao))
    cont+=1
```

Fatiamento

O fatiamento é extremamente útil e prático, além disso, o nome é altamente sugestivo. O fatiamento funciona com a utilização de dois índices da string, um que vai indicar quando a fatia que você quer da string inicia e o outro quando a fatia acaba. Mas é **importante** notar uma coisa, quando você define os índices, **o final da fatia não está incluído**. Por exemplo:

```
nome = "Fernanda"
print (nome[0:2])
print (nome[1:2])
print (nome[0:4])
print (nome[0:8])
```

Você **também pode omitir um dos índices**, e assim estará indicando que a fatia ou vai até o fim ou começa desde o início. Caso ambos os índices sejam omitidos, estaremos fazendo uma cópia das strings.

```
nome = "Fernanda"
print (nome[:4])
print (nome[1:])
print (nome[:])
```

Fatiamento

Valores negativos também podem ser usados, nesse caso, eles indicarão posições a partir da direita. Como nos exemplos:

```
nome = "Fernanda"
print (nome[-1:])
print (nome[-4:7])
print (nome[-2:-1])
print (nome[-4:8])
print (nome[-3:-1])
```

O que acontecerá?

```
for _ in range(1,4):
    print("Pagina %i" %_)
```

```
for v in range(10):
    print(v)
```

```
nome="Jacqueline"
profissao="radialista e professora"
empresa="IZP e Uninassau"
idade=45
print ('Nome: '+nome + '\n'  + 'Profissão: '+profissao + '\n' + 'Empresa: ' +empresa+ '\n' + 'Idade: '+str(idade)+ ' anos')
```

Exercícios

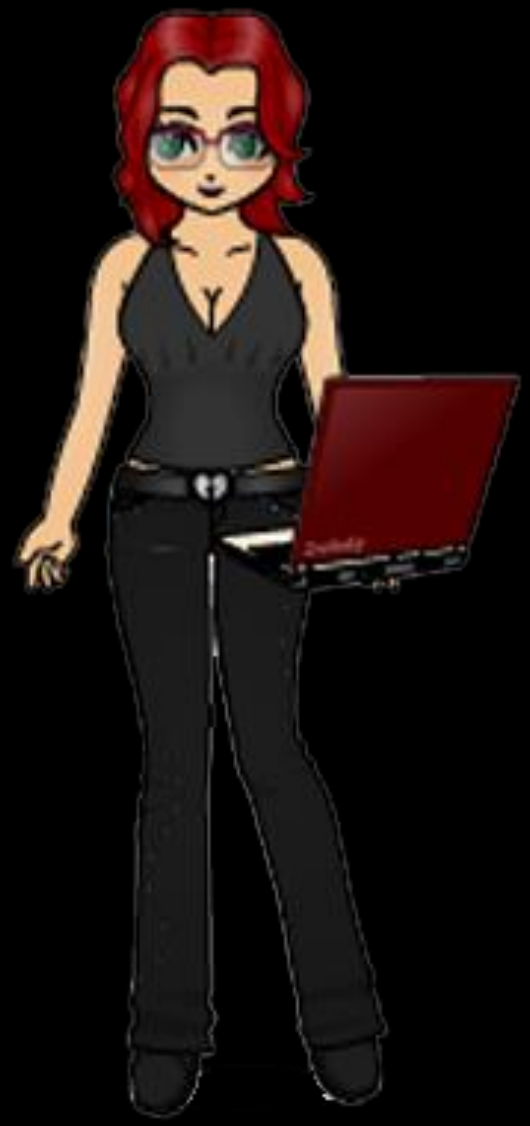
a) Refaça essa questão, solicitando para que o usuário entre com os dados (nome, profissão, empresa e idade).

b) Refaça a letra a) utilizando a estrutura de repetição **For** 3 vezes e colocando um espaço em branco entre os blocos de informação.

Referência complementar...

http://www3.ifrn.edu.br/~jurandy/fdp/doc/aprenda-python/capitulo_07.html
https://panda.ime.usp.br/pensepy/static/pensepy/08-Strings/strings.html
http://www2.ic.uff.br/~vanessa/material/prog-python/10-Strings.pdf

https://youtu.be/BnLF7BJtWRE
https://youtu.be/iO8ugV2uVvo



Contato:
jacfel@gmail.com

Bons estudos!

Fonte: http://academia.unse.edu.ar/nic3b1o-con-pc-gif1.gif