DICIONÁRIOS

OBS: Em algumas linguagens de programação, os dicionários são conhecidos por mapas.

Dicionários são coleções do tipo chave/valor, representados por chaves {}

print(type({}))

>>> <class 'dict'>

OBS: Sobre dicionários:

- Chave e valor são separados por 2 pontos 'chave:valor';

- Chave pode ser qualquer tipo imutavél, valor pode ser qualquer tipo de dado;

Ex: (int, float, string, bool)

- Podemos misturar tipos de dados(type de key != value)

Criando um dicionário:

Ex1:

>>> d = { } # Foi criado um dicionário vazio

>>> d

{ }

--------------------------------------------------------------------------------------------------------------------

Um dicionário com elementos:

Ex1:

>>> d = {'nome': 'Rubem', 'sobrenome': 'Álecks'}

>>> d

{'nome': 'Rubem', 'sobrenome': 'Álecks'}

CRIANDO NOVAS CHAVES E VALORES

>>> dic\_test = {‘chave0’: ‘test0’}

>>> dic\_test

{'chave0': 'test0'}

Ok....

Agora para adicionar uma new key and values:

>>> dic\_test['chave1'] = 'test1'

>>> dic\_test

{'chave0': 'test0', 'chave1': 'test1'}

ACESSANDO DADOS EM DICIONÁRIOS

FORMA 1

Podemos também solicitar uma informação direta ao dicionário criado, usando uma chave especifica. Exemplo:

>>> d = {'nome': 'Rubem', 'sobrenome': 'Álecks'}

>>> d['sobrenome']

'Álecks'

---------------------

# Neste caso precisamos indicar a chave através de colchetes [ ] assim como nas listas. Note que a chave fica entre aspas ‘ ’. A chave pode ser qualquer tipo de imutável, como é o exemplo das tuplas, mas não pode ser uma lista por outro lado. Já o valor pode ser qualquer objeto de valor existente no python.

Ex1:

>>> dic = {('uma\_tupla'): 'qualquer valor'}

>>> dic['uma\_tupla']:

'qualquer valor'

# note que mesmo sendo uma tupla a chave definida é usada da mesma forma, como se fosse uma string, mas caso a chave fosse um numeral (inteiro/float) não usamos as aspas ‘ ’. Caso tente usar aspas para uma chave que é um numeral será retornado um KeyError.

Ex2:

paises = {'br': 'Brasil', 'eua': 'Estados Unidos', 'py': 'Paraguai'}

print(paises['eua'])

Estados Unidos

# Caso tentemos fazer um acesso de uma chave inexistente teremos um 'KeyError'

FORMA 2:

.GET

Acessando via get - FORMA RECOMENDADA

print(paises.get('br'))

print(paises.get('ru')) # caso não encontre a chave retorna None, sem gerar KeyError

# Recomenda-se usar sempre o .get

print(paises.get('ru', 'Não Encontrei'))

~ .get faz a solicitação e caso não ache a key especifica (‘key’) e caso ele não ache retorna o segundo valor (‘Não Encontrei’)

~ caso não determine o 2º elemento por padrão get retorna None, assim evitando um KeyError.

CONSULTANDO CHAVES EM DICIONÁRIOS

>>> print('br' in paises)

True

>>> print('ru' in paises)

False

>>> print('Estados Unidos' in paises) # False, ele busca key e não valor. Estados Unidos é um value e não uma key.

False