

Design de Software

Aula: Dicionários em Python

Problema

Analise o código:

```
lista = ["Fulano", "99999-1111"], ["Sicrano", "99999-2222"],  
        ["Beltrano", "99999-3333"]  
  
nome = input("Digite um nome? ")  
  
for i in range(len(lista)):  
    if lista[i][0] == nome:  
        print("Telefone: {}".format(lista[i][1]))
```

Problema

Não seria bom se ao invés de uma lista indexada por números, houvesse uma “lista” indexada por palavras?

Poderia por exemplo, achar rapidamente um telefone de um amigo sem precisar percorrer a lista...

Solução:

dicionários!

```
lista = [ ["Fulano", "99999-1111"], ["Sicrano", "99999-2222"],  
          ["Beltrano", "99999-3333"] ]
```

```
nome = input("Digite um nome? ")
```

```
for i in range(len(lista)):  
    if lista[i][0] == nome:  
        print("Telefone: {}".format(lista[i][1]))
```



```
dicionario = {"Fulano": "99999-1111", "Sicrano": "99999-2222",  
              "Beltrano": "99999-3333"}
```

```
nome = input("Digite um nome? ")
```

```
if nome in dicionario:  
    print("Telefone: {}".format(dicionario[nome]))
```

Solução: dicionários!

| 0 | 1 | 2 |
|--------------------------|---------------------------|----------------------------|
| ["Fulano", "99999-1111"] | ["Sicrano", "99999-2222"] | ["Beltrano", "99999-3333"] |



| "Fulano" | "Sicrano" | "Beltrano" |
|--------------|--------------|--------------|
| "99999-1111" | "99999-2222" | "99999-3333" |

Dicionários

São como listas, mas o índice pode ser um valor de qualquer tipo

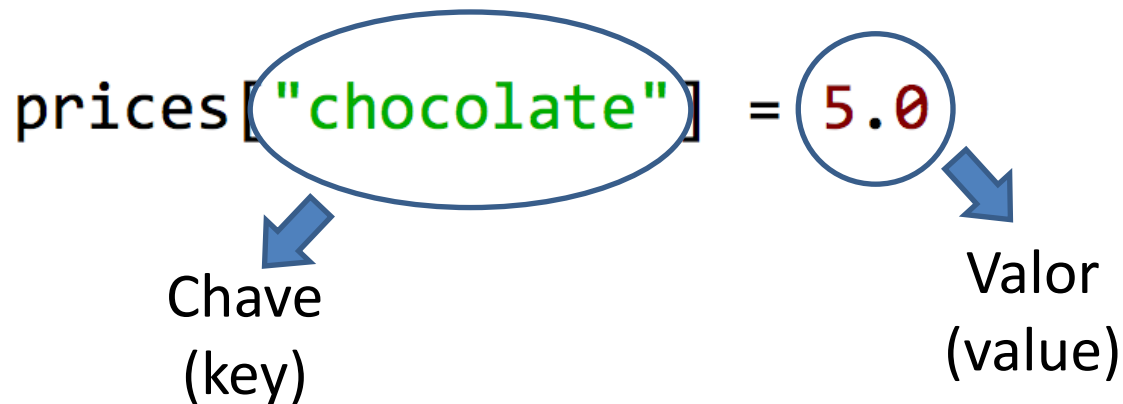
```
prices = dict()    ou    prices = {}
```

```
prices["chocolate"] = 5.0
```

```
prices["juice"]      = 4.
```

```
{'chocolate': 5.0, 'juice': 4.0}
```

Chaves, valores e itens



Item = Chave + Valor

Inicialização de dicionários

Podemos declarar o conteúdo quando criamos o dicionário

```
prices = {'choc': 5.0, 'juice': 4.0}
```


Dados armazenados

Os valores armazenados podem ser de qualquer tipo, listas por exemplo:

```
coleção = dict()
```

```
coleção["Felipe"] = ["brontossauro", "T-Rex"]
```

```
coleção["Julia"] = ["Euro", "Dólar", "Bitcoin"]
```

```
print(coleção)
```

```
{'Felipe': ['brontossauro', 'T-Rex'], 'Julia':  
['Euro', 'Dólar', 'Bitcoin']}
```

Tem *append*?

- Em uma **lista** temos que usar o *método* `.append()` para colocar um elemento extra na lista (criando uma posição nova)

```
minha_lista.append(42)
```

- Em um **dicionário** **isso não é necessário**: basta colocar o novo elemento na nova posição

```
meu_dicionário['abóbora'] = 42
```

Dicionário - exemplo

```
port2eng = dict()
```

```
port2eng["couve"] = "kale"
```

```
port2eng["repolho"] = "cabbage"
```

```
port2eng["brócolis"] = "broccoli"
```

Como seria outro jeito
de criar esse dicionário?

Operador in

in – procura nas chaves

```
c = "couve"  
if c in port2eng:  
    print("{0} é {1}".format(c, port2eng[c]))
```



couve é kale

KeyError

Caso você busque uma chave que não existe no dicionário, o Python emite um erro.

Por isso, teste suas chaves com o operador in

```
x = "alface"  
print(port2eng[x])
```



Traceback (most recent call last):

<várias linhas aqui...>

File "C:/temp/temp.py", line 21, in <module>
 print(port2eng[x])

KeyError: 'alface'

Exercícios

Faça um programa, onde você pergunte o nome da disciplina e o programa imprime o nome do respectivo professor. **Dica: use dicionários.**

Soluções

```
dict_prof_disciplina = {'GDE': 'Prof de GDE', 'ND': 'Prof de ND',  
                        'DS': 'Prof de DS', 'IM': 'Prof de IM', 'MS': 'Prof de MS'}  
  
disciplina = input('Qual a disciplina? ')  
  
if disciplina in dict_prof_disciplina:  
    professor = dict_prof_disciplina[disciplina]  
    print('Professor de {0} é {1}'.format(disciplina, professor))  
else:  
    print('A disciplina {0} não existe!'.format(disciplina))
```

Exercícios

Faça uma função que recebe uma *string* com o nome de um dia da semana e retorna o seu número (“domingo” = 1, “segunda” = 2, etc). Se o dia for inválido, retorne -1.

Use dicionários.

Soluções

```
def dia_semana(dia):  
    semana = {"domingo": 1, "segunda": 2, "terça": 3, "quarta": 4,  
              "quinta": 5, "sexta": 6, "sábado": 7}  
    if dia in semana:  
        return semana[dia]  
    else:  
        return -1
```

Dicionário – voltando ao exemplo

```
port2eng = dict()
```

```
port2eng["couve"] = "kale"
```

```
port2eng["repolho"] = "cabbage"
```

```
port2eng["brócolis"] = "broccoli"
```

Percorrendo um dicionário – modo 1

For percorre as chaves automaticamente

```
for k in port2eng:  
    print("{0}: {1}".format(k, port2eng[k]))
```



```
couve: kale  
repolho: cabbage  
brócolis: broccoli
```

Percorrendo um dicionário – modo 2

A função `items()` retorna os itens

```
for k, v in port2eng.items():  
    print("{0}: {1}".format(k, v))
```



```
couve: kale  
repolho: cabbage  
brócolis: broccoli
```

Operações – keys()

.keys() – retorna as chaves

```
chaves = port2eng.keys()
```

```
for e in chaves:  
    print(e)
```



repolho
couve
brócolis

Operações – values()

.values() – retorna os valores

```
valores = port2eng.values()  
for v in valores:  
    print(v)
```



```
kale  
cabbage  
broccoli
```

Operações - del

Remove um elemento do dicionário:

```
del port2eng["couve"]
```

Exercícios

Faça uma função que recebe uma *string* e retorna um dicionário onde cada chave é uma letra da *string*, e cada valor é o número de ocorrências desta letra. Por exemplo, se passamos a *string* "banana nanica", a função devolve o dicionário:

```
{'b': 1, 'a': 5, 'n': 4, ' ': 1, 'i': 1, 'c': 1}
```

Nota importante: em geral
as chaves do dicionário não
estão ordenadas!

Soluções

```
def conta_letra(texto):  
    contagem = {}  
    for letra in texto:  
        if not letra in contagem:  
            contagem[letra] = 1  
        else:  
            contagem[letra] += 1  
    return contagem
```

Exercícios

Faça um programa para o cálculo da conta da lanchonete:

- a) Represente os preços dos produtos da lanchonete em um dicionário;
- b) Leia uma lista de compras e armazene numa lista; (vai ser uma lista de strings)
- c) Crie uma função que recebe o dicionário de preços e a lista de compras e retorna o valor total da conta;
- d) Imprima o valor total da compra.

Soluções

```
def total_compra(precos, lista):
    valor_total = 0.0
    for item in lista:
        if item in precos:
            valor_total += precos[item]
    return valor_total

menu = { "coxinha": 6.5, "pão de queijo": 4.0, "água": 2.0}

compras = []

produto = input("O que deseja comprar? (fim para terminar): ")
while produto != "fim":
    if produto in menu:
        compras.append(produto)
    else:
        print("Este produto não existe!")

    produto = input("O que deseja comprar? (fim para terminar): ")

total = total_compra(menu, compras)
print("O total das compras é {0}".format(total))
```

Insper

www.insper.edu.br