**FACULDADE DE COMPUTAÇÃO E INFORMÁTICA**

**CIÊNCIA DA COMPUTAÇÃO E SISTEMAS DE INFORMAÇÃO**

**Linguagem de Programação I**

**AULA 10: DICIONARIOS**

|  |  |
| --- | --- |
| http://images.flatworldknowledge.com/ketchen/ketchen-fig05_x001.jpg | Nossos **objetivos** nesta semana são:   * Conhecer o conceito de dicionários. * Trabalhar com dicionários utilizando listas. |
|  | O **Capítulo 4** do livro:  DIERBACH, C. *Introduction to Computer Science Using Python: A Computational Problem Solving Focus.* 1st Edition, New York: Wiley, 2012. |
|  |  |

**DICIONÁRIOS**

* Dicionários consistem em uma estrutura de dados similar às listas, mas com propriedades de acesso diferentes. Um dicionário é composto por um conjunto de chaves e valores. Um dicionário consiste em relacionar uma chave a um valor específico.

**Criação de Dicionários**

* Em Python, criamos dicionários utilizando chaves { }. Cada elemento do dicionário é uma combinação de chaves e valor. Vejamos um exemplo onde os preços de mercadorias sejam como o da Tabela 1.

|  |  |
| --- | --- |
| **Produto** | **Preço** |
| Alface | R$ 0,45 |
| Batata | R$ 1,20 |
| Tomate | R$ 2,30 |
| Feijão | R$ 1,50 |

* A Tabela 1 pode ser vista como um dicionário onde a chave seria o produto; e valor o seu preço. Agora vejamos como criar este dicionário em Python:

tabela = { “Alface”: 0.45,

“Batata”: 1.20,

“Tomate”: 2.30,

“Feijão”: 1.50 }

* Pode-se escrever o comando em uma única linha:

tabela = { “Alface”: 0.45, “Batata”: 1.20, “Tomate”: 2.30, “Feijão”: 1.50 }

* Um dicionário é acessado por meio de suas chaves. Para obter o preço da alface, digite no interpretador, depois de ter criado a tabela, tabela[“Alface”], onde *tabela* é o nome da variável do tipo dicionário e “Alface” é a nossa chave. O valor retornado é o mesmo que associamos na tabela, ou seja, 0,45.
* Diferentemente de listas, onde o índice é um número, dicionários utilizam suas chaves como índice. Quando atribuímos um valor a uma chave, duas coisas podem ocorrer:
  + **Se a chave já existe:** o valor associado é alterado para o novo valor.
  + **Se a chave não existe:** a nova chave será adicionada ao dicionário.

tabela = { “Alface”: 0.45,

“Batata”: 1.20,

“Tomate”: 2.30,

“Feijão”: 1.50 }

print( tabela[“Tomate”] )

>> 2.3

tabela[“Tomate”] = 2.50

print( tabela[“Tomate”] )

>> 2.5

tabela[“Cebola”] = 1.20

print(tabela[“Cebola”] )

>> 1.2

Print (tabela)

>> {‘Batata’: 1.2, ‘Alface’: 0.45, ‘Tomate’: 2.5, ‘Cebola’: 1.2, ‘Feijão’: 1.5}

* Observe que ao utilizar dicionários perde-se a noção de ordem. Durante a manipulação do código acima, a ordem das chaves foi alterada.

**Acesso a uma chave inexistente**

Quanto ao acesso aos dados, é preciso verificar se uma chave existe, antes de acessá-la:

tabela = { “Alface”: 0.45,

“Batata”: 1.20,

“Tomate”: 2.30,

“Feijão”: 1.50 }

print( tabela[“Manga”] )

>> Traceback (most recente call last ):

Fila “<stdin>”, line 1, in <module>

KeyError: ‘Manga’

* Se a chave não existir, uma exceção do tipo KeyError será ativada. Para verificar se uma chave pertence a um dicionário, pode-se usar o operador in:

tabela = { “Alface”: 0.45,

“Batata”: 1.20,

“Tomate”: 2.30,

“Feijão”: 1.50 }

print( “Manga” **in** tabela )

>> **False**

print( “Batata” **in** tabela )

>> **True**

**Obtenção de uma lista de chave de valores**

* É possível obter uma lista com as chaves do dicionário ou mesmo uma lista dos valores associados:

tabela = { “Alface”: 0.45,

“Batata”: 1.20,

“Tomate”: 2.30,

“Feijão”: 1.50 }

print( tabela.keys( ) )

>> dict\_keys( [‘Batata’, ‘Alface’, ‘Tomate’, ‘Cebola’, ‘Feijão’] )

print( tabela.keys( ) )

>> dict\_values( [1.2, 0.45, 2.5, 1.2, 1.5] )

print(tabela.items( ) )

>> dict\_items([('Batata', 2.2), ('Feijão', 1.5), ('Alface', 0.4), ('Tomate', 2.3), ('Cebola', 1.4)])

* Observe que os métodos keys( ) e values( ) retornam geradores. É possível utilizar esses geradores diretamente dentro de um for ou transformá-los e listas usando a função list.

**Obtenção do preço com dicionário**

* Vejamos um programa que utiliza um dicionário para exibir o preço de um produto:

tabela = { “Alface:” 0.45,

“Batata:” 1.20,

“Tomate:” 2.30,

“Feijão:” 1.50 }

while True:

produto = input(“Digite o nome do produto ou a palavra ‘fim’ para terminar: ”)

**if** produto == ‘fim’:

**break**

**if** produto in tabela:

print(“Preço %5.2f” % tabela[produto])

**else:**

print(“Produto não encontrado!”)

* O exemplo acima verifica se o dicionário contém a chave procurada. Em caso afirmativo o preço associado à chave ou haverá uma mensagem de erro.

**Exclusão de uma associação do dicionário**

* Para apagar uma chave, utilizaremos a instrução **del**:

tabela = { “Alface:” 0.45,

“Batata:” 1.20,

“Tomate:” 2.30,

“Feijão:” 1.50 }

del( tabela[“Tomate”] )

print (tabela)

>> {‘Batata’: 1.2, ‘Alface’: 0.45, ‘Cebola’: 1.2, ‘Feijão’: 1.5}

* Você deve estar se perguntando quando utilizar listas e quando utilizar dicionários. Tudo depende do que você deseja realizar. Se seus dados são facilmente acessados por suas chaves, ou seja, quase nunca de uma só vez, então um dicionário é mais interessante. Além disso você pode acessar os valores associados a uma chave rapidamente sem pesquisar.
* Se seus dados precisam preservar a ordem de inserção (como filas ou pilhas), então continue usando listas, pois nestes casos, dicionários não são boas opções.

**EXERCÍCIOS**

1. Crie um dicionário chamado “dicionario”:

* + Coloque nele os dados fornecidos pelo usuário:
    - nome, idade, telefone e endereço.
  + Imprima todos os itens do dicionário no formato chave: valor.

1. Altere o script 02 de form a a solicitar ao usuário o produto e a
2. quantidade vendida. Verifique se o nome do produto digitado existe
3. no dicionário e só então efetue a baixa em estoque.
4. Altere o script 02 de form a a solicitar ao usuário o produto e a
5. quantidade vendida. Verifique se o nome do produto digitado existe
6. no dicionário e só então efetue a baixa em estoque.
7. Crie um dicionário chamado “produtos”:
   1. Adicione seis produtos (chave) e quantidade em estoque (valores).
   2. Solicite ao usuário o produto e a quantidade vendida.
   3. Verifique se o nome do produto digitado existe no dicionário e só então efetue a baixa no estoque.