

# Estrutura de Dados Dicionários

Prof<sup>a</sup>. Dr<sup>a</sup>. Aline Marques  
prof1658@iesp.edu.br

# Dicionários

- Dicionários é um exemplo de tabela *hash*:

# Dicionários

- Dicionários é um exemplo de tabela *hash*:

É uma estrutura de dados utilizada para implementar um array associativo.

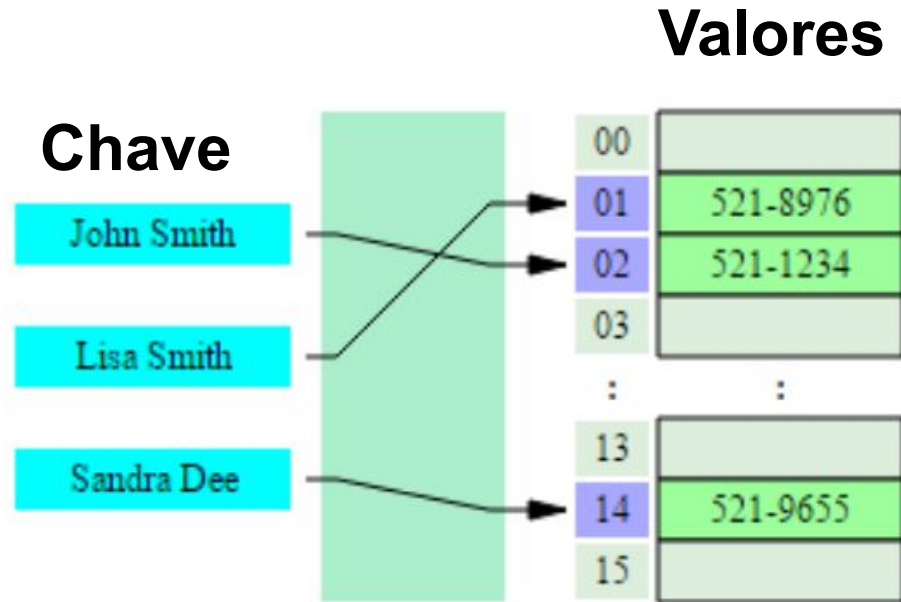
# Dicionários

- Dicionários é um exemplo de tabela *hash*:

É uma estrutura de dados utilizada para implementar um array associativo.

- Cada valor tem uma **chave** associada a ele.
- Utiliza uma função *hash* para computar um índice no array de *slots*.

# Dicionário



# Dicionário - Operações

- Criação do dicionário
- Adicionar nova chave no dicionário
- Adicionar valor no dicionário
- Varrer valores do dicionário
- Varrer chaves do dicionário
- Busca de valores no dicionário
- Busca de chaves no dicionário
- Remoção de chaves no dicionário
- Remoção de valores no dicionário

# Dicionário - Operações

- **Criação do dicionário**
- Adicionar nova chave no dicionário
- Adicionar valor no dicionário
- Varrer valores do dicionário
- Varrer chaves do dicionário
- Busca de valores no dicionário
- Busca de chaves no dicionário
- Remoção de chaves no dicionário
- Remoção de valores no dicionário

# Criação do dicionário

```
def main():  
    dicionario = { }  
  
main()
```

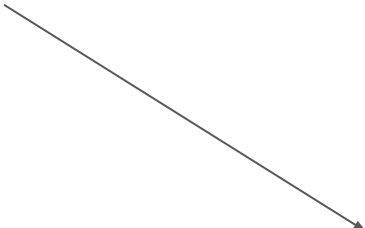


# Criação do dicionário

```
def main():
```

```
    dicionario = { }
```

```
main()
```



{ }

# Dicionário - Operações

- Criação do dicionário
- **Adicionar nova chave no dicionário**
- Adicionar valor no dicionário
- Varrer valores do dicionário
- Varrer chaves do dicionário
- Busca de valores no dicionário
- Busca de chaves no dicionário
- Remoção de chaves no dicionário
- Remoção de valores no dicionário

# Adicionar nova chave no dicionário

```
def main():  
    dicionario = { }  
  
    dicionario['chave1'] = None  
    print(dicionario)  
  
main()
```

# Dicionário - Operações

- Criação do dicionário
- Adicionar nova chave no dicionário
- **Adicionar valor no dicionário**
- Varrer valores do dicionário
- Varrer chaves do dicionário
- Busca de valores no dicionário
- Busca de chaves no dicionário
- Remoção de chaves no dicionário
- Remoção de valores no dicionário

# Adicionar valores no dicionário

```
def main():  
    dicionario = { }  
  
    dicionario['jr'] = '2.123.123'  
    dicionario['mario'] = '3.345.345'  
    print(dicionario)  
  
main()
```

# Adicionar valores no dicionário

```
def main():  
    dicionario = { }  
  
    dicionario['jr'] = ['2.123.123', 1990]  
    dicionario['mario'] = ['3.345.345', 1997]  
    print(dicionario)  
  
main()
```

# Adicionar valores no dicionário

```
def main():  
    dicionario = {'jr': '2.123.123', 'mario': '3.345.345'}  
    print(dicionario)  
  
main()
```

# Exercício A

Crie um dicionário que possua a data de nascimento e endereço completo de 5 usuários.

Obs: as chaves do dicionário será o primeiro nome dos usuários



# Dicionário - Operações

- Criação do dicionário
- Adicionar nova chave no dicionário
- Adicionar valor no dicionário
- **Varrer valores do dicionário**
- Varrer chaves do dicionário
- Busca de valores no dicionário
- Busca de chaves no dicionário
- Remoção de chaves no dicionário
- Remoção de valores no dicionário

# Varrendo valores no dicionário

```
def main():  
    dicionario = { }  
    dicionario['jr'] = ['2.123.123', 1990]  
    dicionario['mario'] = ['3.345.345', 1997]  
  
    for valor in dicionario.values():  
        print(valor)  
  
main()
```

# Dicionário - Operações

- Criação do dicionário
- Adicionar nova chave no dicionário
- Adicionar valor no dicionário
- Varrer valores do dicionário
- **Varrer chaves do dicionário**
- Busca de valores no dicionário
- Busca de chaves no dicionário
- Remoção de chaves no dicionário
- Remoção de valores no dicionário

# Varrendo chaves no dicionário

```
def main():  
    dicionario = { }  
    dicionario['jr'] = ['2.123.123', 1990]  
    dicionario['mario'] = ['3.345.345', 1997]  
  
    for chave in dicionario.keys():  
        print(chave)  
  
main()
```

# Exercício B

Crie um dicionário com endereços vinculados a diferentes CEPs (como chave).

Em seguida, mostre apenas os CEPs registrados.

# Dicionário - Operações

- Criação do dicionário
- Adicionar nova chave no dicionário
- Adicionar valor no dicionário
- Varrer valores do dicionário
- Varrer chaves do dicionário
- **Busca de valores no dicionário**
- Busca de chaves no dicionário
- Remoção de chaves no dicionário
- Remoção de valores no dicionário

# Buscando valores no dicionário

```
def main():  
    dicionario = { }  
    dicionario['jr'] = 1990  
    dicionario['mario'] = 1997  
  
    busca = 1997  
    if busca in dicionario.values():  
        print('esse valor existe no dicionário')  
    else:  
        print('esse valor não existe no dicionário')
```

main()

# Dicionário - Operações

- Criação do dicionário
- Adicionar nova chave no dicionário
- Adicionar valor no dicionário
- Varrer valores do dicionário
- Varrer chaves do dicionário
- Busca de valores no dicionário
- **Busca de chaves no dicionário**
- Remoção de chaves no dicionário
- Remoção de valores no dicionário



# Buscando chaves no dicionário

```
def main():  
    dicionario = { }  
    dicionario['jr'] = 1990  
    dicionario['mario'] = 1997  
  
    busca = 'jr'  
    if busca in dicionario.keys():  
        print('essa chave existe no dicionário')  
    else:  
        print('essa chave não existe no dicionário')
```

main()

# Exercício C

No seu algoritmo do Exercício B, busque informações do dicionário pelo CEP e pelo endereço completo.

# Dicionário - Operações

- Criação do dicionário
- Adicionar nova chave no dicionário
- Adicionar valor no dicionário
- Varrer valores do dicionário
- Varrer chaves do dicionário
- Busca de valores no dicionário
- Busca de chaves no dicionário
- **Remoção de chaves no dicionário**
- Remoção de valores no dicionário

# Removendo chave do dicionário

```
def main():  
    dicionario = { }  
    dicionario['jr'] = 1990  
    dicionario['mario'] = 1997  
    print(dicionario)  
    dicionario.pop('mario')  
    print(dicionario)  
  
main()
```

# Removendo chave do dicionário

```
def main():  
    dicionario = { }  
    dicionario['jr'] = 1990  
    dicionario['mario'] = 1997  
  
    print(dicionario)  
    del(dicionario['jr'])  
    print(dicionario)  
main()
```

# Dicionário - Operações

- Criação do dicionário
- Adicionar nova chave no dicionário
- Adicionar valor no dicionário
- Varrer valores do dicionário
- Varrer chaves do dicionário
- Busca de valores no dicionário
- Busca de chaves no dicionário
- Remoção de chaves no dicionário
- **Remoção de valores no dicionário**

# Removendo valor do dicionário

```
def main():  
    dicionario = { }  
    dicionario['jr'] = 1990  
    dicionario['mario'] = 1997  
  
    print(dicionario)  
    dicionario['jr'] = None  
    print(dicionario)  
main()
```

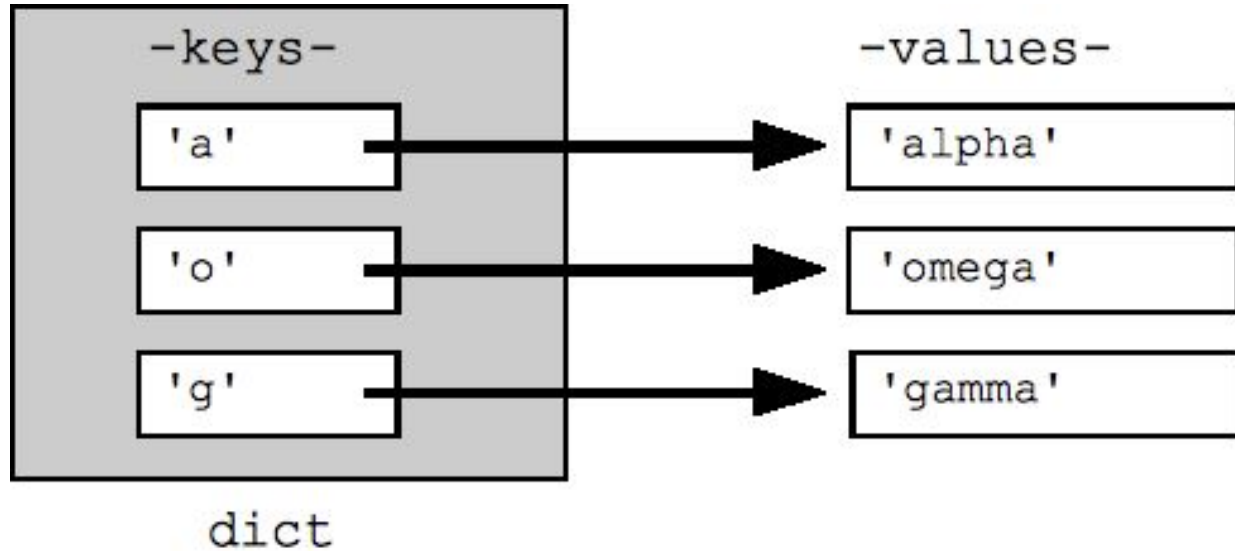
# Exercício D

Crie um dicionário com 6 códigos de barra e o seu respectivo produto e preço.

Em seguida, remova 2 registros do dicionário por chave e outros 2 registros do dicionário por valores.



# Nomenclatura das chaves (índices)



# Nomenclatura das chaves (índices)

1ª Estratégia:

- Chaves com nomes variados

# Nomenclatura das chaves (índices)

## 1ª Estratégia:

- Chaves com nomes variados

```
dicionario = { }  
chave = input('Chave: ')  
dicionario[chave] = None
```

O que pode acontecer se alguém digitar o mesmo nome de chave?



O que pode acontecer se alguém digitar o mesmo nome de chave?

Sobrescrever o valor

O que pode acontecer se alguém digitar o mesmo nome de chave?

Sobrescrever o valor

**Solução: criar uma estrutura de condição para não deixar que isso aconteça**

# Nomenclatura das chaves (índices)

2ª Estratégia:

- Chaves com nomes fixos

# Nomenclatura das chaves (índices)

## 2ª Estratégia:

- Chaves com nomes fixos:

```
dicionario = { }  
dicionario['nome'] = None  
dicionario['login'] = None  
dicionario['senha'] = None
```



O que pode acontecer se for necessário armazenar mais de um registro?



# O que pode acontecer se for necessário armazenar mais de um registro?

```
dicts = []  
dicionario = {}  
  
for i in range(0, 2):  
    dicionario['nome'] = input("nome: ")  
    dicionario['cpf'] = input("cpf: ")  
    dicts.append(dicionario.copy())  
  
print(dicts)
```

# O que pode acontecer se for necessário armazenar mais de um registro?

```
dicts = {}  
dicionario = {}  
  
for i in range(0, 2):  
    dicionario['nome'] = input("nome: ")  
    dicionario['cpf'] = input("cpf: ")  
    dicts[i] = dicionario.copy()  
  
print(dicts)
```



Ao utilizar dicionários dentro de outro, tenha cuidado nas operações, pois as varreduras serão mais complexas.

# Exercício E

Crie um dicionário que armazene informações sobre nome, endereço e cpf de 3 usuários. Em seguida, mostre na tela.

OBS: as chaves dos usuários devem ter os mesmos nomes entre si (nome, endereço e cpf).

# Atividade

Crie um dicionário que será a agenda de clientes que compraram fiado na bodega de dona Chica. Essa agenda contém três informações principais: nome, o quanto deve em reais, telefone e endereço. Crie interações que façam:

- a - O cadastro dos clientes
- b - A atualização dos valores em dívida
- c - A remoção dos clientes que não devem mais
- d - Uma busca por nome

# Dúvidas?



[prof1658@iesp.edu.br](mailto:prof1658@iesp.edu.br)