

# PROGRAMAÇÃO I

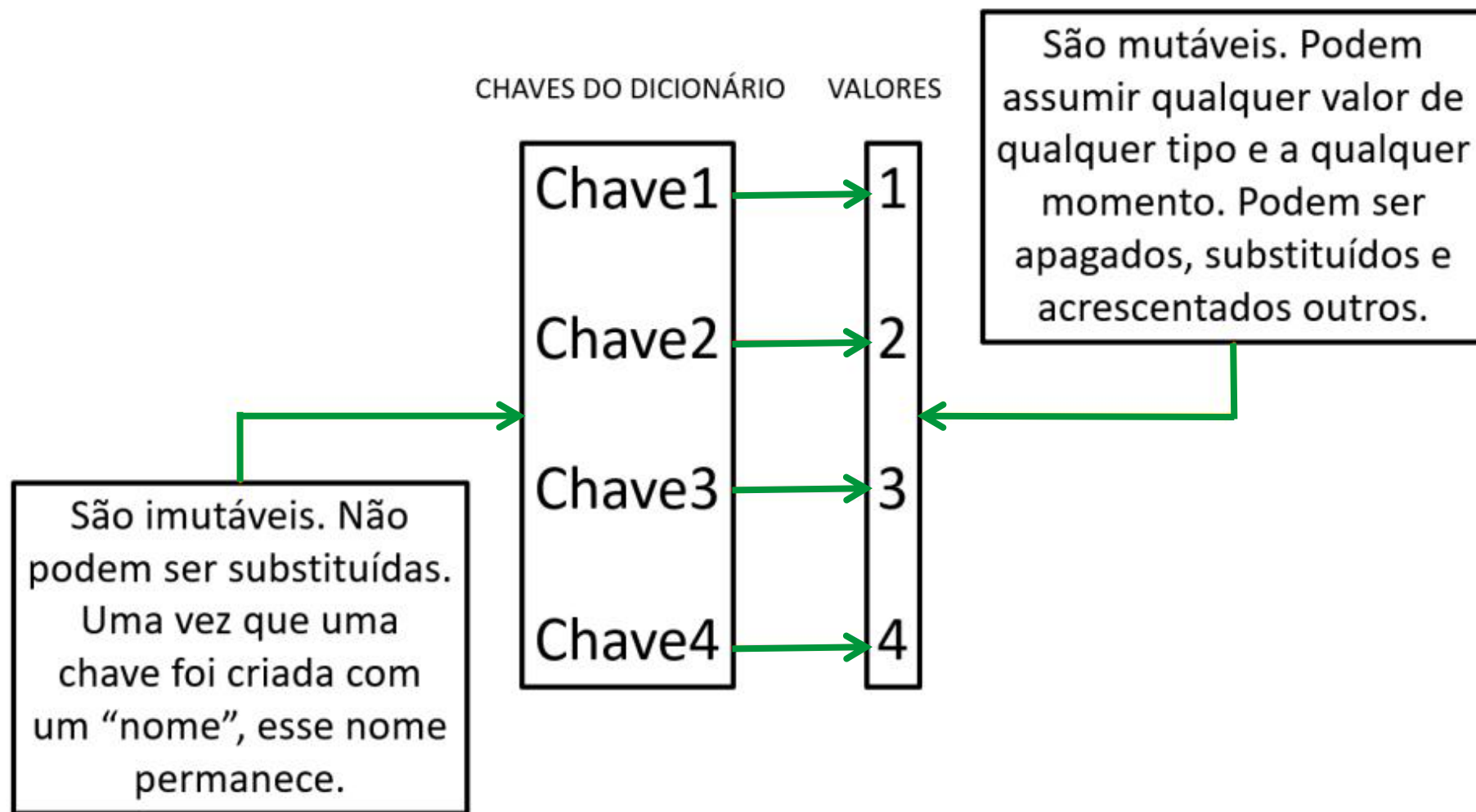
Curso Técnico Subsequente em Informática  
Lucas Sampaio Leite



# Dicionários

- Em Python, um dicionário é uma estrutura de dados que armazena pares de chave e valor.
- Trata-se de uma coleção mutável e não indexada por posição, mas sim pelas chaves.
- Cada chave deve ser de um tipo imutável (como str, int ou tuple), enquanto os valores podem ser de qualquer tipo (mutáveis ou imutáveis).
- O dicionário é considerado mutável porque seus pares chave-valor podem ser alterados, adicionados ou removidos durante a execução do programa.

# Dicionários



# Dicionários

- Um dicionário vazio pode ser criado de forma semelhante às listas e tuplas, utilizando chaves vazias: `nome_da_variavel = {}`
- Para adicionar novos pares chave-valor, utiliza-se a seguinte sintaxe:  
`nome_da_variavel['nome_da_chave'] = elemento_a_ser_adicionado`
- Embora as chaves de um dicionário devam ser imutáveis, elas não precisam ser, necessariamente, strings. Em muitos casos, podem ser tuplas, entre outros tipos imutáveis.

# Dicionários

- Criando e inserindo elementos em um dicionário:

```
dicionario = {"chave1": 1, "chave2": "IF Baiano", "chave3": [1,2,3]}  
print(dicionario)
```



```
{'chave1': 1, 'chave2': 'IF Baiano', 'chave3': [1, 2, 3]}
```



```
dicionario = dict()  
dicionario["chave1"] = 1  
dicionario["chave2"] = "IF Baiano"  
dicionario["chave3"] = [1,2,3]  
  
print(dicionario)
```



```
dicionario = {}  
dicionario["chave1"] = 1  
dicionario["chave2"] = "IF Baiano"  
dicionario["chave3"] = [1,2,3]  
  
print(dicionario)
```

# Dicionários

```
telefones = {}  
print(type(telefones))  
telefones["Lucas"] = "(99)9999.9999"  
telefones["João"] = "(88)8888.8888"  
telefones["Maria"] = "(77)7777.7777"  
print(telefones)
```



```
<class 'dict'>  
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777'}
```



# Dicionários

- Acessando um valor a partir de sua chave:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
print(telefones["Lucas"])  
resultado = telefones["Lucas"]  
print(resultado)  
print(type(resultado))
```



```
(99)9999.9999  
(99)9999.9999  
<class 'str'>
```

# Dicionários

- Acessando um valor a partir de sua chave:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
print(telefones["Zezinho"])
```



```
print(telefones["Zezinho"])  
~~~~~^~~~~~  
KeyError: 'Zezinho'
```

Ao acessar um valor diretamente pela chave, se ela não existir no dicionário, será lançada a exceção `KeyError`.



## Dicionários

- Para evitar uma `KeyError` ao acessar elementos em um dicionário, uma prática comum é verificar se a chave existe antes de utilizá-la. Isso pode ser feito com o operador `in`, que retorna `True` se a chave estiver presente no dicionário e `False` caso contrário:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
  
if "Zezinho" in telefones:  
    print(telefones["Zezinho"])  
else:  
    print("Chave não encontrada")
```

## Dicionários

- Para evitar uma `KeyError` ao acessar elementos em um dicionário, pode-se também utilizar o método `.get()`, que retorna o valor associado à chave ou, caso a chave não exista, um valor padrão definido pelo programador.

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
  
print(telefones.get("Lucas", "Chave não encontrada"))  
print(telefones.get("Zezinho", "Chave não encontrada"))
```



```
(99)9999.9999  
Chave não encontrada
```

# Dicionários

- A inserção de elementos em um dicionário em Python é feita associando uma nova chave a um valor:

```
telephones = {"Lucas": "(99)9999.9999",  
              "João": "(88)8888.8888",  
              "Maria": "(77)7777.7777"}  
  
print("Dicionário antes da inserção:")  
print(telephones)  
  
print("\nDicionário após a inserção:")  
telephones["Antônio"] = "(66)6666.6666"  
print(telephones)
```

Dicionário antes da inserção:

```
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777'}
```

Dicionário após a inserção:

```
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777', 'Antônio': '(66)6666.6666'}
```

# Dicionários

- A inserção de elementos em um dicionário em Python é feita associando uma nova chave a um valor:

```
telephones = {"Lucas": "(99)9999.9999",  
              "João": "(88)8888.8888",  
              "Maria": "(77)7777.7777"}  
  
print("Dicionário antes da inserção:")  
print(telephones)  
  
print("\nDicionário após a inserção:")  
telephones["Antônio"] = "(66)6666.6666"  
print(telephones)
```

Caso a chave já exista,  
o valor anterior será  
sobrescrito pelo novo.

```
Dicionário antes da inserção:  
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777'}  
Dicionário após a inserção:  
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777', 'Antônio': '(66)6666.6666'}
```



# Dicionários

- É possível inserir múltiplos elementos de uma vez utilizando o método

update():

```
telefones = {"Lucas": "(99)9999.9999",
             "João": "(88)8888.8888",
             "Maria": "(77)7777.7777"}

telefones_aux = {"Carlos": ["(55)5555.5555", "(44)4444.4444"],
                 "Joana": ["(33)3333.3333", "(22)2222.2222"]}

telefones.update(telefones_aux)
telefones.update({"Laura": "(11)1111.1111"})

print(telefones)
print(telefones["Carlos"][1])
print(telefones["Joana"][0])
```

```
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777', 'Carlos': ['(55)5555.5555', '(44)4444.4444'], 'Joana': ['(33)3333.3333', '(22)2222.2222'], 'Laura': '(11)1111.1111'}
```

# Dicionários

- Listar todas as chaves:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}
```

```
for chave in telefones:  
    print(chave)
```

```
print(telefones.keys())
```

```
for chave in telefones.keys():  
    print(chave)
```




```
Lucas  
João  
Maria  
dict_keys(['Lucas', 'João', 'Maria'])  
Lucas  
João  
Maria
```




# Dicionários

- Listar todos os valores:

```
telephones = {"Lucas": "(99)9999.9999",  
              "João": "(88)8888.8888",  
              "Maria": "(77)7777.7777"}  
  
for chave in telephones:  
    print(telephones[chave])  
  
print(telephones.values())  
  
for valor in telephones.values():  
    print(valor)
```



```
(99)9999.9999  
(88)8888.8888  
(77)7777.7777  
dict_values(['(99)9999.9999', '(88)8888.8888', '(77)7777.7777'])  
(99)9999.9999  
(88)8888.8888  
(77)7777.7777
```



## Dicionários

- Listar todos os pares chave-valor:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
  
for chave in telefones:  
    print(f"{chave} -> {telefones[chave]}")  
  
print(telefones.items())  
  
for chave, valor in telefones.items():  
    print(f"{chave} -> {valor}")
```



```
Lucas -> (99)9999.9999  
João -> (88)8888.8888  
Maria -> (77)7777.7777  
dict_items([('Lucas', '(99)9999.9999'), ('João', '(88)8888.8888'), ('Maria', '(77)7777.7777')])  
Lucas -> (99)9999.9999  
João -> (88)8888.8888  
Maria -> (77)7777.7777
```

# Dicionários

- Removendo elementos:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
  
print(telefones)  
  
del telefones["João"]  
print(telefones)  
  
valor_removido = telefones.pop("Lucas")  
print(telefones)  
print(valor_removido)
```



```
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777'}  
{'Lucas': '(99)9999.9999', 'Maria': '(77)7777.7777'}  
{'Maria': '(77)7777.7777'}  
(99)9999.9999
```

# Dicionários

- Ao tentar remover uma chave que não existe em um dicionário, o Python lança uma `KeyError`:

```
telephones = {"Lucas": "(99)9999.9999",  
              "João": "(88)8888.8888",  
              "Maria": "(77)7777.7777"}  
  
del telephones["Zezinho"]  
print(telephones)
```



```
del telephones["Zezinho"]  
~~~~~^  
KeyError: 'Zezinho'
```



# Dicionários

- Verificando se a chave existe antes da remoção:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
  
if "Zezinho" in telefones:  
    del telefones["Zezinho"]  
else:  
    print("Chave não encontrada")
```

# Dicionários

- O método `clear()` esvazia completamente o dicionário:

```
telephones = {"Lucas": "(99)9999.9999",  
              "João": "(88)8888.8888",  
              "Maria": "(77)7777.7777"}  
  
print(telephones)  
  
telephones.clear()  
  
print(telephones)
```



```
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777'}  
{}
```



# Dicionários

- Verificando o tamanho do dicionário:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
  
print(len(telefones))  
  
print(telefones)
```



```
3  
{'Lucas': '(99)9999.9999', 'João': '(88)8888.8888', 'Maria': '(77)7777.7777'}
```

# Dicionários

- Em Python, os dicionários podem ser convertidos em listas ou tuplas de diferentes formas:
  - Considerando as chaves:

```
['Lucas', 'João', 'Maria']  
<class 'list'>  
('Lucas', 'João', 'Maria')  
<class 'tuple'>
```



```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}
```

```
lista = list(telefones)  
tupla = tuple(telefones)
```

```
print(lista)  
print(type(lista))
```

```
print(tupla)  
print(type(tupla))
```

# Dicionários

- Em Python, os dicionários podem ser convertidos em listas ou tuplas de diferentes formas:
  - Considerando as chaves:

```
dict_keys(['Lucas', 'João', 'Maria'])  
<class 'dict_keys'>  
['Lucas', 'João', 'Maria']  
<class 'list'>  
('Lucas', 'João', 'Maria')  
<class 'tuple'>
```



```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}  
  
lista = list(telefones.keys())  
tupla = tuple(telefones.keys())  
  
print(telefones.keys())  
print(type(telefones.keys()))  
  
print(lista)  
print(type(lista))  
  
print(tupla)  
print(type(tupla))
```

# Dicionários

- Em Python, os dicionários podem ser convertidos em listas ou tuplas de diferentes formas:
  - Considerando os valores:


```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}
```

```
lista = list(telefones.values())  
tupla = tuple(telefones.values())
```

```
print(telefones.values())  
print(type(telefones.values()))
```

```
print(lista)  
print(type(lista))
```

```
print(tupla)  
print(type(tupla))
```



```
dict_values(['(99)9999.9999', '(88)8888.8888', '(77)7777.7777'])  
<class 'dict_values'>  
['(99)9999.9999', '(88)8888.8888', '(77)7777.7777']  
<class 'list'>  
('(99)9999.9999', '(88)8888.8888', '(77)7777.7777')  
<class 'tuple'>
```



# Dicionários

- Em Python, os dicionários podem ser convertidos em listas ou tuplas de diferentes formas:
  - Considerando os pares chave-valor:

```
telefones = {"Lucas": "(99)9999.9999",  
             "João": "(88)8888.8888",  
             "Maria": "(77)7777.7777"}
```

```
lista = list(telefones.items())  
tupla = tuple(telefones.items())
```

```
print(telefones.items())  
print(type(telefones.items()))
```

```
print(lista)  
print(type(lista))
```

```
print(tupla)  
print(type(tupla))
```



```
dict_items([('Lucas', '(99)9999.9999'), ('João', '(88)8888.8888'), ('Maria', '(77)7777.7777')])  
<class 'dict_items'>  
[('Lucas', '(99)9999.9999'), ('João', '(88)8888.8888'), ('Maria', '(77)7777.7777')]  
<class 'list'>  
(('Lucas', '(99)9999.9999'), ('João', '(88)8888.8888'), ('Maria', '(77)7777.7777'))  
<class 'tuple'>
```

# Exercícios

1. Crie um dicionário contendo as cinco pessoas mais próximas de você, utilizando o nome de cada pessoa como chave e a cor da camisa que ela está usando como valor.
2. Crie um dicionário vazio chamado semana e complete-o com uma chave para cada dia da semana. O valor de cada chave deve ser uma lista contendo as aulas que você tem nesse dia. Para sábado e domingo, a lista pode estar vazia caso você não tenha aulas nesses dias.
3. Crie um dicionário vazio chamado filmes. Para cada filme, utilize o nome como chave e, como valor, outro dicionário que contenha o vilão do filme e o ano em que ele foi lançado. Preencha o dicionário com cinco filmes e, em seguida, acesse os dados armazenados para cada um deles.



# Exercícios

4. Escreva um programa para armazenar uma agenda de telefones em um dicionário. Cada pessoa pode ter um ou mais telefones e a chave do dicionário é o nome da pessoa. Seu programa deve conter um menu onde dependendo da entrada do usuário, será possível:
- `incluirNovoNome` - acrescenta um novo nome na agenda, com um ou mais telefones.
  - `incluirTelefone` - acrescenta um telefone em um nome existente na agenda. Caso o nome não exista na agenda, você deve perguntar se a pessoa deseja incluí-lo.
  - `excluirTelefone` - exclui um telefone de uma pessoa que já está na agenda. Se a pessoa tiver apenas um telefone, ela deve ser excluída da agenda.
  - `excluirNome` - exclui uma pessoa da agenda.
  - `consultarTelefone` - retorna os telefones de uma pessoa na agenda.

# Dúvidas



# PROGRAMAÇÃO I

Curso Técnico Subsequente em Informática  
Lucas Sampaio Leite

