

# PROGRAMAÇÃO WEB II

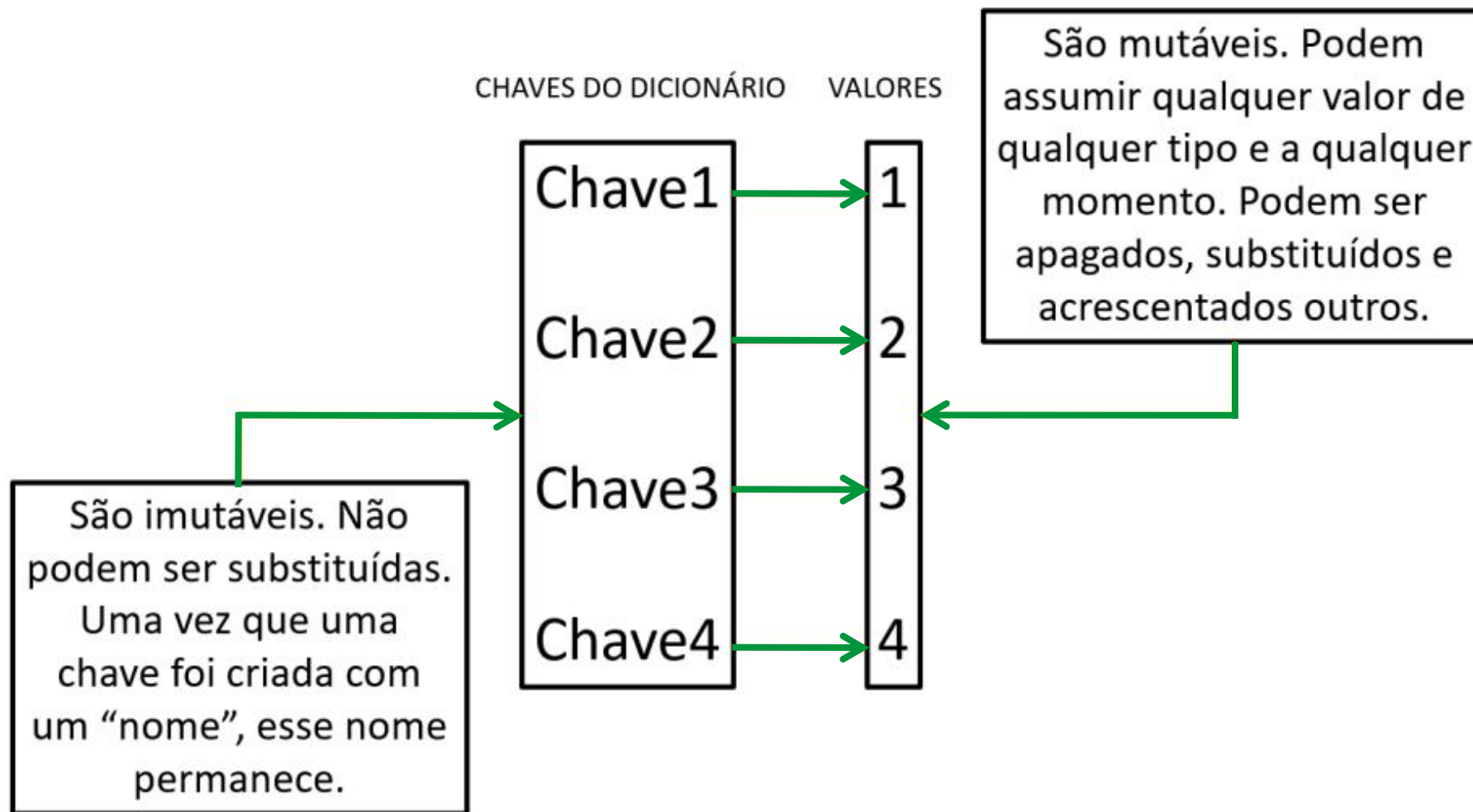
Curso Técnico Integrado em Informática  
Lucas Sampaio Leite



# Dicionários

- Em Python, um dicionário é uma estrutura de dados que mapeia chaves a valores.
- Ele é uma coleção mutável e não ordenada, usada para armazenar pares no formato chave: valor.
- As chaves devem ser de tipos imutáveis (como strings, números ou tuplas imutáveis), enquanto os valores podem ser de qualquer tipo, inclusive mutáveis.
- A estrutura é considerada mutável porque seus valores associados às chaves podem ser alterados, adicionados ou removidos dinamicamente.

# Dicionários



# Dicionários

- Um dicionário pode ser criado vazio de forma semelhante às listas e tuplas, utilizando: `nome_da_variavel = { }`.
- Para adicionar um novo par chave-valor, basta atribuir o valor desejado a uma chave usando a sintaxe: `nome_da_variavel['nome_da_chave'] = valor`.
- Embora as chaves dos dicionários precisem ser imutáveis, elas não precisam ser necessariamente strings.
  - É comum o uso de tuplas imutáveis como chave, entre outros tipos imutáveis como inteiros e booleanos.

# Dicionários

```
diccionario = {"chave1": 1, "chave2": 2, "chave3": 3}  
print(diccionario)
```



```
{'chave1': 1, 'chave2': 2, 'chave3': 3}
```



```
diccionario = dict()  
diccionario["chave1"] = 1  
diccionario["chave2"] = 2  
diccionario["chave3"] = 3  
print(diccionario)
```

# Dicionários

```
telefones = {}  
print(type(telefones))  
telefones['Lucas'] = '(81)9999.9999'  
telefones['João'] = '(81)9999.8888'  
telefones['Maria'] = '(81)9999.7777'  
print(telefones)
```



```
<class 'dict'>  
{'Lucas': '(81)9999.9999', 'João': '(81)9999.8888', 'Maria': '(81)9999.7777'}
```

# Dicionários

- Buscando um valor pela chave:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
| 'Maria': '(81)9999.7777'}  
print(telefones['Lucas'])  
resultado = telefones['Lucas']  
print(resultado)  
print(type(resultado))
```



```
(81)9999.9999  
(81)9999.9999  
<class 'str'>
```



# Dicionários

- Buscando um valor pela chave: se a chave não existir no dicionário, uma exceção `KeyError` será lançada. Para evitar isso, pode-se usar o método `get()` para acessar o valor de uma chave, retornando um valor padrão caso a chave não exista:

```
telefones.get('Zezinho', 'default value')
```



# Dicionários

- Adicionando um novo par chave-valor:

```
telephones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
| 'Maria': '(81)9999.7777'}  
print('Dicionário antes da inserção: ')  
print(telephones)  
print('\nDicionário após a inserção: ')  
telephones['Antônio']='(81)9999.6666'  
print(telephones)
```



```
Dicionário antes da inserção:  
{'Lucas': '(81)9999.9999', 'João': '(81)9999.8888', 'Maria': '(81)  
9999.7777'}  
  
Dicionário após a inserção:  
{'Lucas': '(81)9999.9999', 'João': '(81)9999.8888', 'Maria': '(81)  
9999.7777', 'Antônio': '(81)9999.6666'}
```

# Dicionários

- Unindo dicionários:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
             'Maria': '(81)9999.7777'}  
telefones_aux = {'Carlos': ['(87)9999.6666', '(87)9999.6669'],  
                 'Joana': ['(87)9999.5555', '(87)9999.5559']}  
print(telefones)  
print(telefones_aux)  
telefones.update(telefones_aux)  
print(telefones)  
print(telefones['Joana'][0])  
print(telefones['Joana'][1])
```

# Dicionários

- Convertendo dicionários em listas e tuplas:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
             'Maria': '(81)9999.7777'}  
lista = list(telefones)  
tupla = tuple(telefones)  
print(lista)  
print(type(lista))  
print(tupla)  
print(type(tupla))
```



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

# Dicionários

- Convertendo dicionários em listas e tuplas:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
             'Maria': '(81)9999.7777'}  
lista = list(telefones)  
tupla = tuple(telefones)  
print(lista)  
print(type(lista))  
print(tupla)  
print(type(tupla))
```



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

As chaves de um dicionário podem ser convertidas em uma lista ou tupla utilizando as funções `list()` ou `tuple()`, respectivamente.



# Dicionários

- Listando todas as chaves de um dicionário:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
             'Maria': '(81)9999.7777'}  
for telefone in telefones:  
    print(telefone)  
print(telefones.keys())
```



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

# Dicionários

- Listando todos os valores de um dicionário:

```
telephones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
| 'Maria': '(81)9999.7777'}  
for telefone in telephones:  
|     print(telephones[telefone])  
print(telephones.values())
```



```
(81)9999.9999  
(81)9999.8888  
(81)9999.7777  
dict_values(['(81)9999.9999', '(81)9999.8888', '(81)9999.7777'])
```

# Dicionários

- Convertendo dicionários para listas e tuplas:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
             'Maria': '(81)9999.7777'}  
lista = list(telefones.values())  
tupla = tuple(telefones.values())  
print(lista)  
print(type(lista))  
print(tupla)  
print(type(tupla))
```



```
['(81)9999.9999', '(81)9999.8888', '(81)9999.7777']  
<class 'list'>  
('(81)9999.9999', '(81)9999.8888', '(81)9999.7777')  
<class 'tuple'>
```



# Dicionários

- Percorrendo chaves e valores de um dicionário:

```
telefones = {  
    'Lucas': '(81)9999.9999',  
    'João': '(81)9999.8888',  
    'Maria': '(81)9999.7777',  
    'Carlos': ['(87)9999.5555', '(87)9999.5559']  
}  
  
for chave, valor in telefones.items():  
    print(f'chave -> {chave}, valor -> {valor}')
```



```
chave -> Lucas, valor -> (81)9999.9999  
chave -> João, valor -> (81)9999.8888  
chave -> Maria, valor -> (81)9999.7777  
chave -> Carlos, valor -> ['(87)9999.5555', '(87)9999.5559']
```

# Dicionários

- Removendo um elemento de um dicionário:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
| 'Maria': '(81)9999.7777'}  
print(telefones)  
telefones.pop('Lucas')  
print(telefones)  
del telefones['João']  
print(telefones)
```



```
{'Lucas': '(81)9999.9999', 'João': '(81)9999.8888', 'Maria': '(81)  
9999.7777'}  
{'João': '(81)9999.8888', 'Maria': '(81)9999.7777'}  
{'Maria': '(81)9999.7777'}
```

# Dicionários

- Removendo um elemento de um dicionário:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
| 'Maria': '(81)9999.7777'}  
print(telefones)  
telefones.pop('Carlos')
```



```
{'Lucas': '(81)9999.9999', 'João': '(81)9999.8888', 'Maria': '(81)  
9999.7777'}  
Traceback (most recent call last):  
  File "/home/lucas/Documents/vscode-projects/teste.py", line 4, in  
n <module>  
    telefones.pop('Carlos')  
KeyError: 'Carlos'
```

# Dicionários

- Removendo um elemento de um dicionário:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
| 'Maria': '(81)9999.7777'}  
print(telefones)  
if 'Carlos' in telefones:  
|     telefones.pop('Carlos')  
else:  
|     print('A chave não existe!')
```



```
{'Lucas': '(81)9999.9999', 'João': '(81)9999.8888', 'Maria': '(81)  
9999.7777'}  
A chave não existe!
```



# Dicionários

- Verificando se um elemento existe no dicionário:

```
telefones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
'Maria': '(81)9999.7777'}  
print(telefones.get('Lucas', "Não existe no dicionário!"))  
print(telefones.get('Sérgio', "Não existe no dicionário!"))
```



```
(81)9999.9999  
Não existe no dicionário!
```

# Dicionários

- Verificando o tamanho de um dicionário:

```
telephones = {'Lucas': '(81)9999.9999', 'João': '(81)9999.8888',  
              'Maria': '(81)9999.7777'}  
print(telephones)  
print(len(telephones))
```



```
{'Lucas': '(81)9999.9999', 'João': '(81)9999.8888', 'Maria': '(81)  
9999.7777'}  
3
```

# Exercícios

1. Faça um dicionário com as 5 pessoas mais perto de você, tendo o nome como chave e a cor da camisa que está usando como valor.
2. Crie um dicionário vazio semana = {} e o complete com uma chave para cada dia da semana, tendo como seu valor uma lista com as aulas que você tem nesse dia (sábado e domingo recebem listas vazias, ou você tem aula?).
3. Crie um dicionário vazio filmes = {}. Utilize o nome de um filme como chave. E, como valor, outro dicionário contendo o vilão e o ano em que o filme foi lançado. Preencha 5 filmes e acesse os dados do dicionários.



# 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 WEB II

Curso Técnico Integrado em Informática  
Lucas Sampaio Leite

