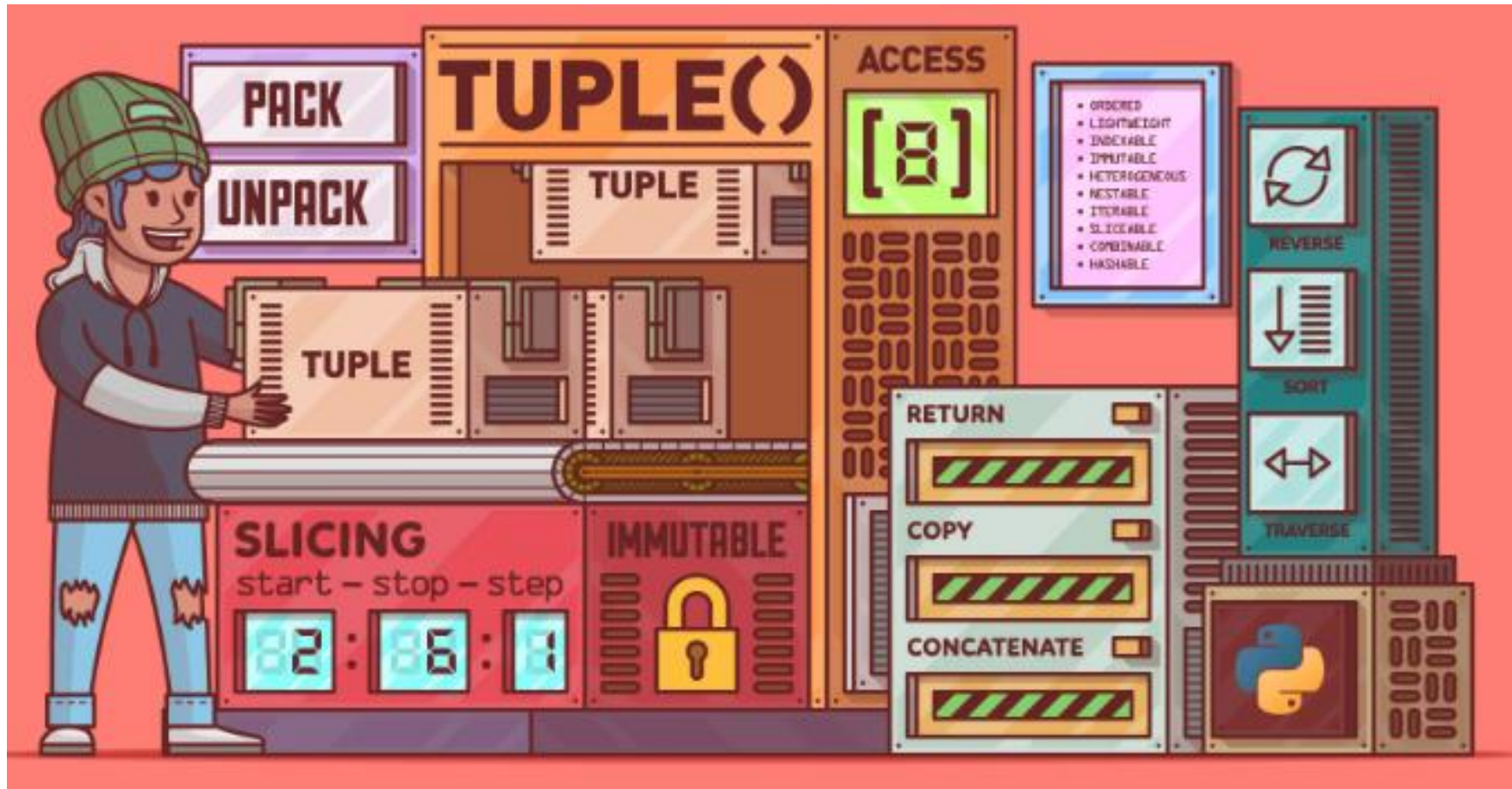






# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

As Tuplas são estruturas de dados em programação que permitem armazenar diversos elementos de diferentes tipos. As Tuplas compartilham semelhanças com listas, mas apresentam a característica fundamental de serem imutáveis.

Isso implica que não é possível adicionar, remover ou atribuir valores aos elementos de uma tupla após sua criação.

Para criar Tuplas, utilizamos parênteses, sendo importante mencionar que os parênteses não são obrigatórios.



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

Tuplas.

```
3 linguagens = ('Assembly' , 'Java' , 'Python', 'C#')
4 print(linguagens)
5 #Resultado: ('Assembly' , 'Java' , 'Python', 'C#')
6
7
8 linguagens = 'Assembly' , 'Java' , 'Python', 'C#'
9 print(linguagens)
10 #Resultado: ('Assembly' , 'Java' , 'Python', 'C#')
11
```



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

O acesso a elementos em tuplas é feito por meio de indexação, semelhante às listas em Python. Os elementos em uma tupla são ordenados e cada elemento possui um índice que indica a sua posição na tupla. Vamos supor que temos a seguinte Tupla:

```
2 lanches = ('Hambúrguer', 'Cerveja', 'Pizza', 'Pudim')
```

```
4 #Acessando o segundo item da tupla
5 segundoItem = lanches[1]
6 #Resultado: 'Cerveja'
```

```
9 #Acessando os itens da tupla de trás para frente
10 ultimoItem = lanches[-1]
11 #Resultado: 'Pudim'
```



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

```
2 lanches = ('Hambúrguer', 'Cerveja', 'Pizza', 'Pudim')
```

```
13 #Acessando mais de um item de uma tupla
14 doisItens = lanches[1:3]
15 #Resultado: ('Cerveja', 'Pizza')
```

```
19 #Acesando até o ultimo item da tupla
20 ateUltimoItem = lanches[1:]
21 #Resultado: ('Cerveja', 'Pizza', 'Pudim')
22
```

```
24 #Acesando do primeiro item da tupla
25 desdePrimeiroItem = lanches[:3]
26 #Resultado: ('Hamburguer', 'Cerveja', 'Pizza')
```



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

As tuplas em Python são imutáveis, o que significa que, uma vez criadas, não podem ser alteradas. Se você tentar modificar um elemento específico de uma tupla ou tentar atribuir um novo valor a um índice, isso resultará no seguinte erro.

```
2 lanches = ('Hambúrguer', 'Cerveja', 'Pizza', 'Pudim')
3
4 lanches[1] = 'Batata Frita'
5 #Resultado: TypeError: 'tuple' object does not support item assignment
```



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

O loop for é frequentemente utilizado com tuplas em Python para iterar sobre os elementos da tupla, acessando cada item sequencialmente. Aqui estão alguns cenários comuns nos quais o loop for é usado com tuplas:

```
1  lanche = ('Hambúrguer', 'Cerveja', 'Pizza', 'Pudim')
2
3  #Verificando o tamanho de uma Tupla
4  tamanhoTupla = len(lanche)
5
6  #Acessando a Tupla de forma sequencial: Maneira Tradicional
7  for i in range(0, len(lanche)):
8      print(lanche[i])
9
```

```
2  lanche = ('Hambúrguer', 'Cerveja', 'Pizza', 'Pudim')
3
4  #Acessando a Tupla de forma sequencial
5  for comida in lanche:
6      print(comida)
7
```

```
2  lanche = ('Hambúrguer', 'Cerveja', 'Pizza', 'Pudim')
3
4  #Listando e enumerando uma Tupla do FOR
5  for pos, comida in enumerate(lanche):
6      print(f"Eu vou comer {comida} na posição {pos}")
7
```





# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

A conversão de uma lista para uma tupla em Python é feita usando a função **tuple()**. Essa função cria uma nova tupla a partir dos elementos da lista fornecida.

A conversão de uma tupla para uma lista é realizada usando a função **list()**

```
3  # Lista de exemplo
4  minha_lista = [1, 2, 3, 4, 5]
5
6  # Convertendo a lista em uma tupla
7  minha_tupla = tuple(minha_lista)
8
9  # Exibindo a tupla resultante
10 print(minha_tupla)
11
```

```
4  # Tupla de exemplo
5  minha_tupla = (1, 2, 3, 4, 5)
6
7  # Convertendo a tupla em uma lista
8  minha_lista = list(minha_tupla)
9
10 # Exibindo a lista resultante
11 print(minha_lista)
12
```



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

Tuplas em Python são objetos imutáveis, o que significa que, ao contrário das listas, não é possível modificar seus elementos depois de criadas. No entanto, mesmo sendo imutáveis, você pode usar alguns métodos e operações comuns em tuplas. Aqui estão algumas situações em que métodos podem ser úteis:



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

O método **sorted()** pode ser usado para criar uma nova lista ordenada temporariamente a partir dos elementos da tupla.

```
1  lanche = ('Hambúrguer', 'Cerveja', 'Pizza')
2  doces = ('Pudim', 'Sorvete', 'Chocolate')
```

```
4  #Metodo SORTED deixa uma tupla em ordem
5  organizandoTuplas = sorted(lanche)
6  #Resultado: ('Cerveja', 'Hambúrguer', 'Pizza')
7
```





# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

O **operador +** pode ser usado para concatenar duas tuplas, criando uma nova tupla resultante.

```
1  lanche = ('Hambúrguer', 'Cerveja', 'Pizza')
2  doces = ('Pudim', 'Sorvete', 'Chocolate')
```

```
8  #Concatenando duas Tuplas em uma nova Tupla
9  lancheDoces = lanche + doces
10 #Resultado: ('Hambúrguer', 'Cerveja', 'Pizza', 'Pudim', 'Sorvete', 'Chocolate')
11
```



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

Quando aplicada a uma tupla, o **len()** retorna o número de elementos (itens) contidos na tupla.

```
1  lanche = ('Hambúrguer', 'Cerveja', 'Pizza')
2  doces = ('Pudim', 'Sorvete', 'Chocolate')
```

```
12  #Verificando o tamanho da Tupla
13  tamanhoLanche = len(lanche)
14  #Resultado: 3
```



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

O método **count()** retorna o número de ocorrências de um determinado valor na tupla.

```
1  lanche = ('Hambúrguer', 'Cerveja', 'Pizza')
2  doces = ('Pudim', 'Sorvete', 'Chocolate')
```

```
16  #Verificando quantas vezes um item se repete na Tupla
17  quantidadeRepeticoes = lanche.count('Cerveja')
18  #Resultado: 1
```





# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

O método **index()** retorna o índice da primeira ocorrência do valor especificado.

```
1  lanche = ('Hambúrguer', 'Cerveja', 'Pizza')
2  doces = ('Pudim', 'Sorvete', 'Chocolate')
```

```
20  #Mostrando a posição de um item, pegando a sua primeira ocorrência
21  posicaoItem = lanche.index('Cerveja')
22  #Resultado: 1
```



# LÓGICA DE PROGRAMAÇÃO

## LINGUAGEM DE PROGRAMAÇÃO

### Tuplas.

Em tuplas, o **del** pode ser usado para excluir a tupla inteira, mas não para remover elementos individuais, pois as tuplas são imutáveis.

```
1  lance = ('Hambúrguer', 'Cerveja', 'Pizza')
2  doces = ('Pudim', 'Sorvete', 'Chocolate')
```

```
24  #Apagando a tupla por completo
25  del(lance)
26
```



# VAMOS PRATICAR.

EXEMPLOS E ATIVIDADES DE TUPLAS

colab

