

# Fundamentos de Python

## Estruturas de Dados

Agora que já aprendemos a usar variáveis, fazer contas, tomar decisões e repetir comandos, chegou a hora de aprender como guardar vários dados ao mesmo tempo.

**Lista(list):** A lista é como uma caixa onde colocamos vários itens. Ela é feita com colchetes [].

```
frutas = ["maçã", "banana", "laranja"]
print(frutas[0]) # maçã
frutas.append("uva")
frutas[1] = "melancia"
frutas.remove("laranja")
```

- **Tuplas (tuple)** : As tuplas são parecidas com as listas, mas não podem ser alteradas depois de criadas. São feitas com parênteses ().

```
coordenadas = (10, 20)
print(coordenadas[1]) # 20
```

**Conjuntos (set)** O set é usado quando queremos guardar itens únicos, sem repetir. E a ordem não importa!

```
numeros = {1, 2, 3, 3}
print(numeros) # {1, 2, 3}
```

### Dicionários (dict):

O dict funciona como um formulário. Cada informação tem um nome (a "chave") e um valor.

```
pessoa = {"nome": "Lucas", "idade": 25}

print(pessoa["nome"]) # Lucas
pessoa["altura"] = 1.75
pessoa["idade"] = 26
```

## Métodos úteis das listas

```
numeros = [3, 1, 4, 1, 5, 9]
numeros.sort()
numeros.count(1)
numeros.pop()
```

## Percorrendo coleções

```
for fruta in frutas:
    print(fruta)
for i, fruta in enumerate(frutas):
    print(i, fruta)
```

**Compreensão de listas.** Um jeito mais rápido de montar uma lista:

```
quadrados = [x * x for x in range(6)]
print(quadrados) # [0, 1, 4, 9, 16, 25]
```

## Desafios práticos

**1. Lista de tarefas** Crie uma lista com suas tarefas do dia. Adicione, remova e organize com sort().

**2. Contador de palavras** Peça pro usuário digitar uma frase. Conte quantas vezes cada palavra apareceu. Dica: use split() e dict.