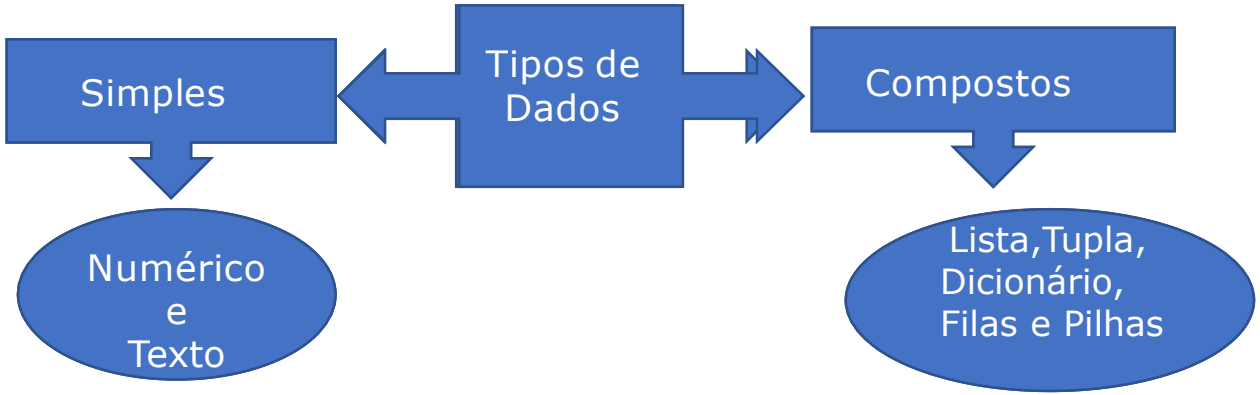
TIPOS DE DADOS

ESCOPO DE UMA VARIÁVEL

O escopo de uma variável é a faixa de sentenças nas quais ela é visível. Uma variável é visível em uma sentença se ela pode ser referenciada ou atribuída nessa sentença.

O escopo de uma variável indica sua visibilidade – ou seja, a partir de onde, no código, a variável é acessível.

- Tipos de Dados:
  - Os tipos de dados podemos definir em dois tipos os simples e compostos.



Diferenças entre Lista, Tupla e Dicionário

- Lista:
  - E uma seqüencia de valores de qualquer tipo,são mutáveis podendo ser alternadas a qualquer momento.
- Tupla:
  - É um tipo de lista com objetivo de armazenar um conjunto de elementos acessíveis pelo índice inteiro.
  - Diferentes da lista ,são imutáveis.
- Dicionário:
  - São conhecidos também como "vetores de associação", tendo como diferença das listas na sua indexação, pois ela é feita através de chaves.
- Classe:
  - Estrutura fundamental para definir objetos.

Listas

- Não existe vetores em Python, nós usamos **listas**
- Uma lista guarda uma seqüência de itens São
- identificadas por colchetes “[ ]”

```
>>> lista = [1,2,3,4,5]
>>> print lista  [1, 2, 3, 4, 5]
```

- Os itens podem ser acessados como um vetor

```
>>> lista[0]  1
>>> lista[4]  5
```

Listas

Inserindo elementos

- **append**(obj)
  - Insere o objeto no fim da lista

```
>>> lista = [1,2,3]
>>> lista.append(0)
>>> lista  [1, 2, 3, 0]
```

- **insert**(posição, obj)
  - Insere o objeto na posição especificada

```
>>> lista = [1,2,3]
>>> lista.insert(1, 'objeto')
>>> lista
[1, 'objeto', 2, 3]
```

Listas

Removendo elementos

- **remove**(obj)
  - Remove o objeto da lista

```
>>> lista = ['a','b',3]
>>> lista.remove('b')
>>> lista  ['a', 3]
```

- **pop**(posição)
  - Remove e retorna o objeto da posição especificada. O pop() sem parâmetros retira o último elemento

```
>>> lista = ['a', 2, 'b', 4, 'c']
>>> lista.pop(0)  'a'
>>> lista
[2, 'b', 4, 'c']
>>> lista.pop()  'c'
>>> lista  [2, 'b', 4]
```

Listas

Outros métodos

- **count**(obj)
  - Retorna a quantidade de elementos “obj” da lista
- **index**(obj)
  - Retorna a posição do “obj” na lista
- **reverse**()
  - Inverte a lista colocando os elementos ao contrário
- **sort**()
  - Ordena a lista

Tuplas

As tuplas são objetos iguais as listas com a diferença de que tuplas são imutáveis. Uma vez criadas não podem ser modificadas.

Sintaxe:

```
>>> t = (1,2,3)
>>> print t  (1,2,3)
>>> t[0] = 5
< erro >
>>> print t[0]  1
>>>
```

Tuplas

Com as tuplas podemos fazer uma manipulação diferente das variáveis. Essa técnica é chamada de "packing-unpacking".

```
>>> (a,b) = ("ola", 5.734)
>>> print a  ola
>>>print b
5.734
>>> a,b = "ola", 5.734
                                    #pode se usar sem parênteses
```

O código acima atribuiu "ola" para *a* e 5.734 para *b*.
Podemos facilmente trocar os valores entre duas variáveis:

```
>>> a = 5
>>> b = 1
>>> a,b = b,a
>>> print a  1
>>> print b  5
```

Dicionários

- O programador pode definir a chave do item
- Representados por chaves “{ }”

```
>>> dicionario = {'chave' : 'item', 2 : 4}
>>> dicionario['chave']  'item'
>>> dicionario[2]  4
```

Dicionários

- Inserir item

  –dicionario[chave] = obj Remover item

- –dicionario.**pop**(chave)
  - Se não encontrar a chave retorna erro

  –dicionario.**pop**(chave, retorno)
  - Se não encontrar a chave retorna o valor de retorno

  Remover todos itens

- –dicionario.**clear**()

Dicionários

- Acessar item
  - dicionario[chave]
    - Se a chave não existir retorna erro
  - dicionario.**get**(chave)
    - Se a chave não existir retorna None
  - dicionario.**get**(chave, retorno)
    - Se a chave não existir retorna o valor de retorno
- Verificar se existe item
  - dicionario.**has_key**(chave)