Tipos de instruções

O interpretador Python entende intruções que estão divididas em 3 categorias

Expressões (expressions)

Toda instrução que gera um processamento e espera um valor de retorno, exprimimos qual operação desejamos que seja computada usando funções ou operadores matemáticos e sempre esperamos obter um resultado para armazenar em variáveis ou efetuar operações de comparação.

exemplo:

```
# Expressão literal, retorno o próprio número
>>> 1
1

# Expressão, retorna o próprio texto
>>> "Bruno"
'Bruno'

# Expressão de chamada de método ou função
>>> "Bruno".upper()
BRUNO

# Expressão com operador, retorna um resultado booleano
>>> 1 > 2
False

>>> 89 >= 89
True

>>> "Bruno" != "Joao"
True

# Chamada de função
>>> sum([1, 2, 3])
6
```

A precedencia de operadores no Python.

Além da precedencia de operadores aritméticos PEMDAS também existe a tabela de precedência de operadores da própria linguagem:

Nível	Categoria	Operadores
7(alto)	exponenciação	**
6	multiplicação	*,/,//,%
5	adição	+,-
4	relacional	==,!=,<=,>=,>,<
3	lógico	not
2	lógico	and
1(baixo)	lógico	or

Declarações (statements)

São formadas por uma ou mais palavras chave e servem para preparar o interpretador para efetuar alguma operação, são comandos que alteram estado ou declaram fluxo lógico.

Algumas palavras chave que são statementes `if, else, elif, for, while, pass, def`

Exemplos:

```
# Condicional com comparação de
if 1 > 2:
    print("Eita, um é maior que 2?")

# Comparação com comparação de igualdade
if x == y:
   # faça isso
else:
   # façá aquilo

# Repetição com comparação de valor
while numero < 10:
    numero += 1
    print(numero)

# Instrução nula
if nao_faca_nada:
    pass
```

Nos exemplo `if 1 > 2:` temos um statemente `if` seguido de uma expressão `1 > 2`

Atribuição (Assignment)

É o nome dado a expressão que pega o resultado de uma expressão e salva em uma variável atribuindo um `nome/identificador` ao resultado que pode ser usado como referencia para acesso.

```
preco = 10
quantidade = 5
```

```
total = preco * quantidade
print(f"O total da sua compra é {total}")
```

A atribução é sempre feita com o sinal de `=` e do lado esquerdo definimos um identificador e do lado direito a expressão a ser atribuida.

No exemplo `total = preco * quantidade` primeiro o Python resolve a expressão `10 * 5` resultando emm `50` e então a apartir do sinal de `=` armazena `50` como valor da variável `total`