

- `#` – Comentário de linha única.
- `"""` – Comentário de múltiplas linhas.
- `%` – Operador de módulo (resto da divisão).
- `%=` – Atribuição com módulo.
- `*=` – Atribuição com multiplicação.
- `****` – Operador de exponenciação.
- `*=` – Atribuição com exponenciação.
- `+` – Operador de adição.
- `+=` – Atribuição com adição.
- `-` – Operador de subtração.
- `-=` – Atribuição com subtração.
- `/` – Operador de divisão.
- `//` – Operador de divisão inteira.
- `//=` – Atribuição com divisão inteira.
- `/=` – Atribuição com divisão.
- `=` – Operador de atribuição.
- `==` – Comparação de igualdade.
- `!=` – Comparação de desigualdade.
- `>` – Comparação de maior que.
- `<` – Comparação de menor que.
- `>=` – Comparação de maior ou igual que.
- `<=` – Comparação de menor ou igual que.
- `[]` – Acesso a elementos de uma lista ou string.
- `{}` – Criação de dicionários ou f-strings.
- `()` – Criação de tuplas ou chamada de funções.

## A

- **abs**: Retorna o valor absoluto de um número.
- **all**: Retorna True se todos os elementos de uma sequência forem verdadeiros.
- **any**: Retorna True se pelo menos um elemento de uma sequência for verdadeiro.
- **assert**: Verifica se uma condição é verdadeira, lançando um erro se não for.

## B

- **break**: Sai de um loop.

## C

- **class**: Define uma classe.
- **continue**: Pula para a próxima iteração de um loop.
- **def**: Define uma função.
- **del**: Remove um objeto ou elemento de uma coleção.

- **dict**: Cria um dicionário.
- **dir**: Retorna uma lista de atributos e métodos de um objeto.

## E

- **elif**: Verifica outra condição se a primeira não for verdadeira.
- **else**: Executa algo se nenhuma das condições anteriores for verdadeira.
- **enumerate**: Retorna um objeto enumerado, que é um iterador de tuplas contendo índices e valores.
- **eval**: Avalia uma string como código Python.
- **exec**: Executa uma string como código Python.
- **except**: Captura e trata exceções (erros) que ocorrem no bloco try.

## F

- **False**: Valor booleano que representa falso.
- **filter**: Filtra elementos de uma sequência com base em uma função.
- **finally**: Executa um bloco de código independentemente de ocorrer uma exceção.
- **float**: Converte um valor em um número de ponto flutuante.
- **for**: Cria um loop que percorre uma sequência de elementos.
- **from**: Importa módulos ou funções específicas de um módulo.
- **frozenset**: Cria um conjunto imutável.

## G

- **getattr**: Retorna o valor de um atributo de um objeto.
- **global**: Declara uma variável como global.

## H

- **help**: Exibe a documentação de um objeto.

## I

- **id**: Retorna o identificador único de um objeto.
- **if**: Verifica se uma condição é verdadeira.
- **import**: Importa módulos.
- **in**: Verifica se um valor está presente em uma sequência.
- **input**: Permite que o usuário insira algo.
- **int**: Converte uma entrada em um número inteiro.
- **is**: Verifica se dois objetos são o mesmo objeto.
- **isinstance**: Verifica se um objeto é uma instância de uma classe ou tipo.
- **issubclass**: Verifica se uma classe é subclasse de outra.

## L

- **lambda**: Cria uma função anônima (lambda).

- **len**: Retorna o comprimento de uma sequência.
- **list**: Cria uma lista.
- **locals**: Retorna um dicionário com as variáveis locais.

## M

- **map**: Aplica uma função a cada item de uma sequência.
- **max**: Retorna o maior valor entre os argumentos.
- **memoryview**: Retorna um objeto de visualização de memória.
- **min**: Retorna o menor valor entre os argumentos.

## N

- **None**: Representa a ausência de valor.
- **nonlocal**: Declara uma variável como não local.
- **not**: Inverte o valor de uma condição.
- **not in**: Verifica se um valor não está presente em uma sequência.

## O

- **open**: Abre um arquivo.
- **or**: Combina condições e retorna verdadeiro se pelo menos uma for verdadeira.
- **ord**: Retorna o valor Unicode de um caractere.

## P

- **pass**: Declara um bloco de código vazio.
- **pow**: Retorna o resultado de um número elevado a outro.
- **print**: Exibe algo na tela.
- **property**: Define uma propriedade de uma classe.

## R

- **range**: Cria uma sequência de números.
- **repr**: Retorna uma string que representa um objeto.
- **return**: Retorna um valor de uma função.
- **reversed**: Retorna um iterador invertido.
- **round**: Arredonda um número.

## S

- **set**: Cria um conjunto.
- **setattr**: Define o valor de um atributo de um objeto.
- **slice**: Cria um objeto de fatiamento.
- **sorted**: Retorna uma lista ordenada.
- **str**: Converte um valor em uma string.
- **sum**: Soma os elementos de uma sequência.

- **super**: Permite acessar métodos da classe pai.

## T

- **True**: Valor booleano que representa verdadeiro.
- **tuple**: Cria uma tupla.
- **type**: Retorna o tipo de um objeto.

## V

- **vars**: Retorna o dicionário de atributos de um objeto.

## W

- **while**: Cria um loop que continua enquanto uma condição for verdadeira.
- **with**: Usado para gerenciar contextos (como abrir e fechar arquivos).
- **yield**: Retorna um valor de uma função geradora.

## Z

- **zip**: Combina elementos de múltiplas sequências em tuplas.