

# Objetivo Geral

Aprender como o interpretador Python utiliza a indentação do código para delimitar os blocos de comandos.

# A estética

Indentar código é uma forma de manter o código fonte mais legível e manutenível. Mas em Python ela exerce um segundo papel, através da indentação o interpretador consegue determinar onde um bloco de comando inicia e onde ele termina.

# Bloco de comando

As linguagens de programação costumam utilizar caracteres ou palavras reservadas para terminar o início e fim do bloco. Em Java e C por exemplo, utilizamos chaves:

# Bloco em Java

```
void sacar(double valor) { // início do bloco do método

    if (this.saldo >= valor) { // início do bloco do if

        this.saldo -= valor;

    } // fim do bloco do if

} // fim do bloco do método
```

# Bloco em Java sem formatar

```
void sacar(double valor) { // início do bloco do método
if (this.saldo >= valor) { // início do bloco do if
this.saldo -= valor;
} // fim do bloco do if
} // fim do bloco do método
```

# Utilizando espaços

Existe uma convenção em Python, que define as boas práticas para escrita de código na linguagem. Nesse documento é indicado utilizar 4 espaços em branco por nível de indentação, ou seja, a cada novo bloco adicionamos 4 novos espaços em branco.

# Bloco em Python

```
def sacar(self, valor: float) -> None: # início do bloco do método

    if self.saldo >= valor: # início do bloco do if

        self.saldo -= valor

    # fim do bloco do if

# fim do bloco do método
```

# Isso não funciona em Python!

```
def sacar(self, valor: float) -> None: # início do bloco do método
if self.saldo >= valor: # início do bloco do if
self.saldo -= valor
# fim do bloco do if
# fim do bloco do método
```

# Qual versão é mais fácil de ler?

```
void sacar(double valor) {  
  if (this.saldo >= valor) {  
    this.saldo -= valor;}}
```

```
def sacar(self, valor: float) -> None:  
    if self.saldo >= valor:  
        self.saldo -= valor
```