

## Trabalhando com Classes no Python.

```
class Pessoa:
```

```
    def __init__(self, nome, idade):  
        self.nome = nome  
        self.idade = idade  
  
    def saudacao(self):  
        return f"Meu nome é {self.nome} e eu tenho {self.idade} anos."
```

Nesse exemplo, criamos uma classe chamada Pessoa. Ela possui dois atributos (nome e idade) e um método (saudacao()). O método `__init__()` é um construtor especial que é executado automaticamente quando um objeto da classe é criado. Ele é usado para inicializar os atributos do objeto com os valores passados como argumentos.

Considere nosso exemplo anterior: uma classe para representar pessoas em um sistema. Mesmo que duas pessoas tenham os mesmos atributos (como nome, idade, sexo, altura e peso), cada pessoa é única.

Para criar um objeto em Python, basta chamar o nome da classe seguido de parênteses vazios, como mostrado a seguir:

```
nome_objeto = nome_da_classe()
```

Isso cria uma instância da classe, que é um objeto único com suas próprias características e comportamentos, como mostra o exemplo a seguir.

Exemplo: definindo uma classe e objetos

```
class Pessoa:
```

```
    def __init__(self, nome, idade):  
        self.nome = nome  
        self.idade = idade  
  
    def saudacao(self):  
        return f"Meu nome é {self.nome} e eu tenho {self.idade} anos."
```

```
# Criando objetos da classe Pessoa
```

```
pessoa1 = Pessoa("João", 30)
```

```
pessoa2 = Pessoa("Maria", 25)
```

```
# Acessando os atributos e métodos dos objetos
```

```
print(pessoa1.nome)
```

```
print(pessoa2.idade)  
print(pessoa1.saudacao())  
print(pessoa2.saudacao())
```

### **Saída**

Meu nome é João e eu tenho 30 anos.

Meu nome é Maria e eu tenho 25 anos.