

# Python

**Programação Orientada a Objetos**

**Lindemberg Silva  
Joir Ramalho  
Frederico Pranto**

# **Python - Principais características**

**Linguagem interpretada**

**Dinamicamente tipada**

**Estrutura o código pela indentação**

**Permite sobrecarga de operadores**

**As classes devem ser importadas uma-a-uma**

# Java - Principais características

Linguagem interpretada

Fortemente tipada

Estrutura o código em blocos {}

Não permite sobrecarga de operadores

As classes podem ser importadas de uma vez só

# **C++ - Principais características**

**Linguagem compilada**

**Fortemente tipada**

**Estrutura o código em blocos {}**

**Permite sobrecarga de operadores**

**Um único programa precisa ser compilado em cada sistema operacional no qual o código deve ser executado**

# Python - Código

```
count = 0
for i in range(1000):
    count += i

print(count)
```

# Criando Classes

```
1 class Carro(object):  
2     pass
```

# Criando Classes

```
1 class Carro():  
2     pass|
```

# Instanciando Classes

```
1 ▼ class Carro():  
2     pass  
3  
4     Fusca = Carro()
```



# Atributos de Instância

```
1 ▼ class Carro():  
2     estado = "NOVO"  
3  
4     c = Carro()  
5     print(c.estado) # 'NOVO'
```

# Atributos de Classe

```
1 ▼ class Carro():  
2     estado = "NOVO"  
3  
4     print(Carro.estado) # 'NOVO'
```

# Métodos

```
1 ▼ class Carro(object):  
2     estado = "NOVO"  
3 ▼     def dirigir(self):  
4         self.estado = "USADO"  
5  
6     # Exemplo 1  
7     porsche = Carro()  
8     porsche.dirigir()  
9     print(porsche.estado) # USADO  
10  
11    # Exemplo 2  
12    ferrari = Carro()  
13    print(ferrari.estado) # NOVO
```

# Self Explícito

```
1 ▼ class Carro(object):  
2     estado = "NOVO"  
3 ▼     def dirigir(self):  
4         self.estado = "USADO"  
5  
6     # Exemplo 1  
7     porsche = Carro()  
8     porsche.dirigir()  
9     print(porsche.estado) # USADO  
10  
11    # Exemplo 2  
12    ferrari = Carro()  
13    print(ferrari.estado) # NOVO
```

# Construtor

```
1 ▼ class Carro(object):  
2 ▼     def __init__(self, estado):  
3         self.estado = estado  
4  
5     bmw = Carro('SEMINOVO')  
6     print(bmw.estado) # 'SEMINOVO'
```

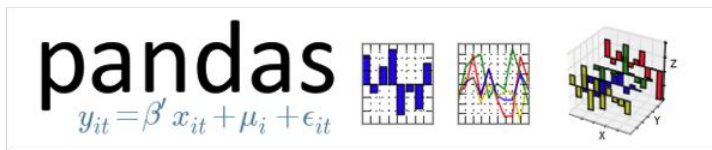
# Herança

```
1 ▼ class Veiculo(object):  
2     estado = "NOVO"  
3  
4 ▼ class Carro(Veiculo):  
5     pass  
6  
7     bmw = Carro()  
8     print(bmw.estado) # 'NOVO'|
```

# Bibliotecas



IP[y]: IPython  
Interactive Computing



# Comparação

// Python

```
count = 0
for i in
range(1000):
    count += i

print(count)
```

// Java

```
public class Java {
    public static void main(String[]
args) {
        int count = 0;
        for(int i=0; i<1000; i++){
            count += i;
        }
        System.out.println(count +
""");
    }
}
```

// C++

```
#include <iostream>

int main (){
    unsigned int i, count=0;
    for(i=0; i<1000; ++i){
        count += i;
    }
    std::cout << count <<
std::endl;
    return 0;
}
```