

```
In [ ]: senai = "bom lugar"
```

```
In [ ]: type(senai)
```

```
Out[ ]: str
```

```
In [ ]: dir(senai)
```

```
Out[ ]: ['__add__',
         '__class__',
         '__contains__',
         '__delattr__',
         '__dir__',
         '__doc__',
         '__eq__',
         '__format__',
         '__ge__',
         '__getattribute__',
         '__getitem__',
         '__getnewargs__',
         '__getstate__',
         '__gt__',
         '__hash__',
         '__init__',
         '__init_subclass__',
         '__iter__',
         '__le__',
         '__len__',
         '__lt__',
         '__mod__',
         '__mul__',
         '__ne__',
         '__new__',
         '__reduce__',
         '__reduce_ex__',
         '__repr__',
         '__rmod__',
         '__rmul__',
         '__setattr__',
         '__sizeof__',
         '__str__',
         '__subclasshook__',
         'capitalize',
         'casefold',
         'center',
         'count',
         'encode',
         'endswith',
         'expandtabs',
         'find',
         'format',
         'format_map',
         'index',
         'isalnum',
         'isalpha',
         'isascii',
         'isdecimal',
         'isdigit',
         'isidentifier',
         'islower',
         'isnumeric',
         'isprintable',
         'isspace',
         'istitle',
         'isupper',
         'join',
         'ljust',
         'lower',
         'lstrip',
         'maketrans',
         'partition',
         'removeprefix',
```

```
'removesuffix',
'replace',
'rfind',
'rindex',
'rjust',
'rtpartition',
'rsplit',
'rstrip',
'split',
'splitlines',
'startswith',
'strip',
'swapcase',
'title',
'translate',
'upper',
'zfill']
```

```
In [ ]: senai.__class__ # nao deve os fazer
```

```
Out[ ]: str
```

```
In [ ]: id(senai)
```

```
Out[ ]: 3025184350864
```

## poo pratica

```
In [ ]: class Pessoa():
        def __init__(self, _nome, _idade ): #metodo construtor
            self.nome = _nome
            self.idade = _idade

        def apresentar(self):
            print(f"Olá, eu sou {self.nome}. Tenho {self.idade} anos. Prazer em te conh
```

```
In [ ]: joao = Pessoa("joao", 35)
```

```
In [ ]: joao.apresentar()
```

Olá, eu sou joao. Tenho 35 anos. Prazer em te conhecer

```
In [ ]: maria = Pessoa("Maria", 27)
```

```
In [ ]: maria.apresentar()
```

Olá, eu sou Maria. Tenho 27 anos. Prazer em te conhecer

```
In [ ]: dir (joao)
```

```
Out[ ]: ['__class__',
        '__delattr__',
        '__dict__',
        '__dir__',
        '__doc__',
        '__eq__',
        '__format__',
        '__ge__',
        '__getattr__',
        '__getstate__',
        '__gt__',
        '__hash__',
        '__init__',
        '__init_subclass__',
        '__le__',
        '__lt__',
        '__module__',
        '__ne__',
        '__new__',
        '__reduce__',
        '__reduce_ex__',
        '__repr__',
        '__setattr__',
        '__sizeof__',
        '__str__',
        '__subclasshook__',
        '__weakref__',
        'idade',
        'nome']
```

```
In [ ]: lista_de_pessoas = [
        Pessoa("Joao", 20),
        Pessoa("Jose", 40),
        Pessoa("Maria", 27)
    ]
```

```
In [ ]: for pessoa in lista_de_pessoas:
        pessoa.apresentar()
```

Olá, eu sou Joao. Tenho 20 anos. Prazer em te conhecer  
 Olá, eu sou Jose. Tenho 40 anos. Prazer em te conhecer  
 Olá, eu sou Maria. Tenho 27 anos. Prazer em te conhecer

```
In [ ]: class Pessoa():
        def __init__(self, _nome, _idade ): #método construtor
            self.nome = _nome
            self.idade = _idade

        def apresentar(self):
            print(f"Olá, eu sou {self.nome}. Tenho {self.idade} anos. Prazer em te conh
```

```
In [ ]: class Funcionario(Pessoa):
        def __init__(self, _nome, _idade, _cargo, _salario):
            super().__init__( _nome, _idade)
            self.cargo = _cargo
            self.salario = _salario

        def funcao(self):
            print(f"Meu cargo é {self.cargo}")
```

```
In [ ]: eliabel = Funcionario("Eliabel", 32, "Professor", 15000)
```

```
In [ ]: eliabel.apresentar()  
        eliabel.funcao()
```

Olá, eu sou Eliabel. Tenho 32 anos. Prazer em te conhecer  
Meu cargo é Professor

```
In [ ]: eliabel.cargo
```

```
Out[ ]: 'Professor'
```

```
In [ ]: eliabel.salario
```

```
Out[ ]: 15000
```