

## Material de apoio

`__add__(obj, x)`: Chamado quando `a + x`

```
def __add__(self, n):  
    if not isinstance(n, int):  
        n = int(n)  
    return self.n1 + n
```

`__sub__(obj, x)`: Chamado quando `a - x`

```
def __sub__(self, n):  
    if not isinstance(n, int):  
        n = int(n)  
    return self.n1 - n
```

`__mul__(obj, x)`: Chamado quando `a * x`

```
def __mul__(self, n):  
    if not isinstance(n, int):  
        n = int(n)  
    return self.n1 * n
```

`__mod__(obj, x)`: Chamado quando `a % x`

```
def __mod__(self, n):  
    if not isinstance(n, int):  
        n = int(n)  
    return self.n1 % n
```

`__neg__(obj)`: Chamado quando `-a`

```
def __neg__(self):  
    return list(map(lambda x: -x if x > 0 else x, self.n1))
```

`__pos__(obj)`: Chamado quando `+a`

```
def __pos__(self):  
    return list(map(lambda x: +x if x < 0 else x, self.n1))
```

`__getitem__(obj, key)`: Chamado quando `a['key']`

```
def __getitem__(self, key):  
    if not key in self.pessoa:  
        self.pessoa[key] = None  
    return self.pessoa[key]
```

`__setitem__(obj, key, value)`: chamado quando `a[key] = value`

```
def __setitem__(self, key, value):  
    self.pessoa[key] = value
```

`__delitem__(obj, key)`: chamado quando `del a[key]`

```
def __delitem__(self, key):  
    self.pessoa[key] = None
```

`__call__(obj, *args, **kwargs)`: chamado quando `a(args)`

```
def __call__(self, *args, **kwargs):  
    print(args)
```