

Métodos e funções

Wednesday, September 16, 2020 11:11 PM

Métodos

Métodos podem executar uma ação específica no objeto, pode ou não ter argumentos. Possuem a seguinte estrutura:

objeto.método(arg, arg2, ...)

No jupyter é possível ver os métodos disponíveis utilizando a tecla TAB.

help() - A função help explica como utilizar um método

```
lista = [1, 2, 3]
help(lista.count)

Help on built-in function count:

count(value, /) method of builtins.list instance
    Return number of occurrences of value.
```

dir () - A função dir lista os métodos e atributos disponíveis para aquele objeto passado como parâmetro.

```
dir(lista)

['_add_',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__']
```

Funções

def nome_funcao (parametro) :

```
def funcao(nome) :
    print('%s, te amo!' %(nome))
```

```
funcao('Angelo')
```

```
Angelo, te amo!
```

Built-in Functions				
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

Função Lambda

Nome da função = **lambda** parametro de entrada: saída

```
def media1(nota1, nota2) :
    media = (nota1 + nota2)/2
    return media
```

```
media2 = lambda nota1, nota2 : (nota1 + nota2)/2
```

```
media1(2,10)
```

6.0

```
media2(2,10)
```

6.0