
Métodos de Desenvolvimento de Software - Python

O conteúdo da prova é identificar o que ocorre no código e explicar o que está ocorrendo no código.

❖ *Funções*

Funções são blocos de código identificados por um nome, que podem receber parâmetros pré-determinados.

Cada função tem um ponto de entrada, e sua execução termina em um ponto de saída, contendo no seu fluxo de execução: Sequências de Ações, Seleções e Iterações.

Tem o objetivo de facilitar a compreensão, o desenvolvimento, o teste e a manutenção dos sistemas.

Se utiliza a palavra-chave **def** para definir a função.

Ex:

```
def my_function():  
    print("Hello from a function")
```

❖ **Self**

A variável representa a instância do próprio objeto, pode acessar os atributos e métodos de uma classe, responsável por vincular os atributos com os argumentos enviados para uma função ou método.

❖ **`__init__()`**

É um Método Construtor (Atribui valores às propriedades do objeto ou outras operações necessárias quando o objeto está sendo criado).

❖ Herança

A Herança nos permite definir uma classe que herda todos os métodos e propriedades de outra classe.

A Classe Pai é a classe da qual está sendo herdada, também chamada de classe Base.

Classe Filha é a classe que herda de outra classe, também chamada de classe Derivada.

❖ super()

A função é usada para dar acesso a métodos e propriedades de uma classe pai ou irmã, e Retorna um Objeto que Representa a Classe Pai.

Ex:

```
class Parent:
    def __init__(self, txt):
        self.message = txt

    def printmessage(self):
        print(self.message)

class Child(Parent):
    def __init__(self, txt):
        super().__init__(txt)

x = Child("Hello, and welcome!")

x.printmessage()
```

❖ Return

Se utiliza a palavra-chave **return** para devolver algum valor.

Ex:

```
def myfunction():
    return 3+3

print(myfunction())
```

❖ *Variáveis Locais e Globais*

Variáveis Globais: São todas as que são definidas no exterior de uma função (na “Raiz” do nosso código) e têm um alcance global, ou seja, podem ser acessadas de qualquer função.

Variáveis Locais: São todas as que são definidas no interior de uma função, ou seja, só pode ser usada dentro da sua função.

Resumindo, nem todas as variáveis são acessíveis a partir de todos os locais do nosso código.

Ex:

```
x = 0 # x é uma variável global

def setup():
    size(256, 256)
    background(0, 0, 200)

def draw():
    # Podemos usar/ler o valor da variável global x.
    ellipse(x, height / 2, 100, 100)

def keyPressed():
    # Queremos alterar o valor da variável global x.
    global x
    x = x + 1
```

❖ *Palavra-Chave Global*

Normalmente, quando você cria uma variável dentro de uma função, essa variável é Local e só pode ser usada Dentro dessa Função.

Para criar uma variável Global Dentro de uma Função, você pode usar a palavra-chave **global**, também pode ser usada para alterar uma variável global dentro de uma função.

Ex:

```
def myfunc():  
    global x  
    x = "fantastic"  
  
myfunc()  
  
print("Python is " + x)
```

Para alterar o valor de uma variável global dentro de uma função, consulte a variável usando a palavra-chave **global**:

```
x = "awesome"  
  
def myfunc():  
    global x  
    x = "fantastic"  
  
myfunc()  
  
print("Python is " + x)
```

Links Úteis Para Estudar

https://www.w3schools.com/python/python_functions.asp

https://www.w3schools.com/python/python_scope.asp

https://www.w3schools.com/python/python_variables_global.asp

https://www.w3schools.com/python/gloss_python_global_scope.asp

https://www.w3schools.com/python/gloss_python_class_init.asp

https://www.w3schools.com/python/ref_func_super.asp

https://www.w3schools.com/python/python_inheritance.asp

https://www.w3schools.com/python/ref_keyword_return.asp