



Linguagem de Programação Python

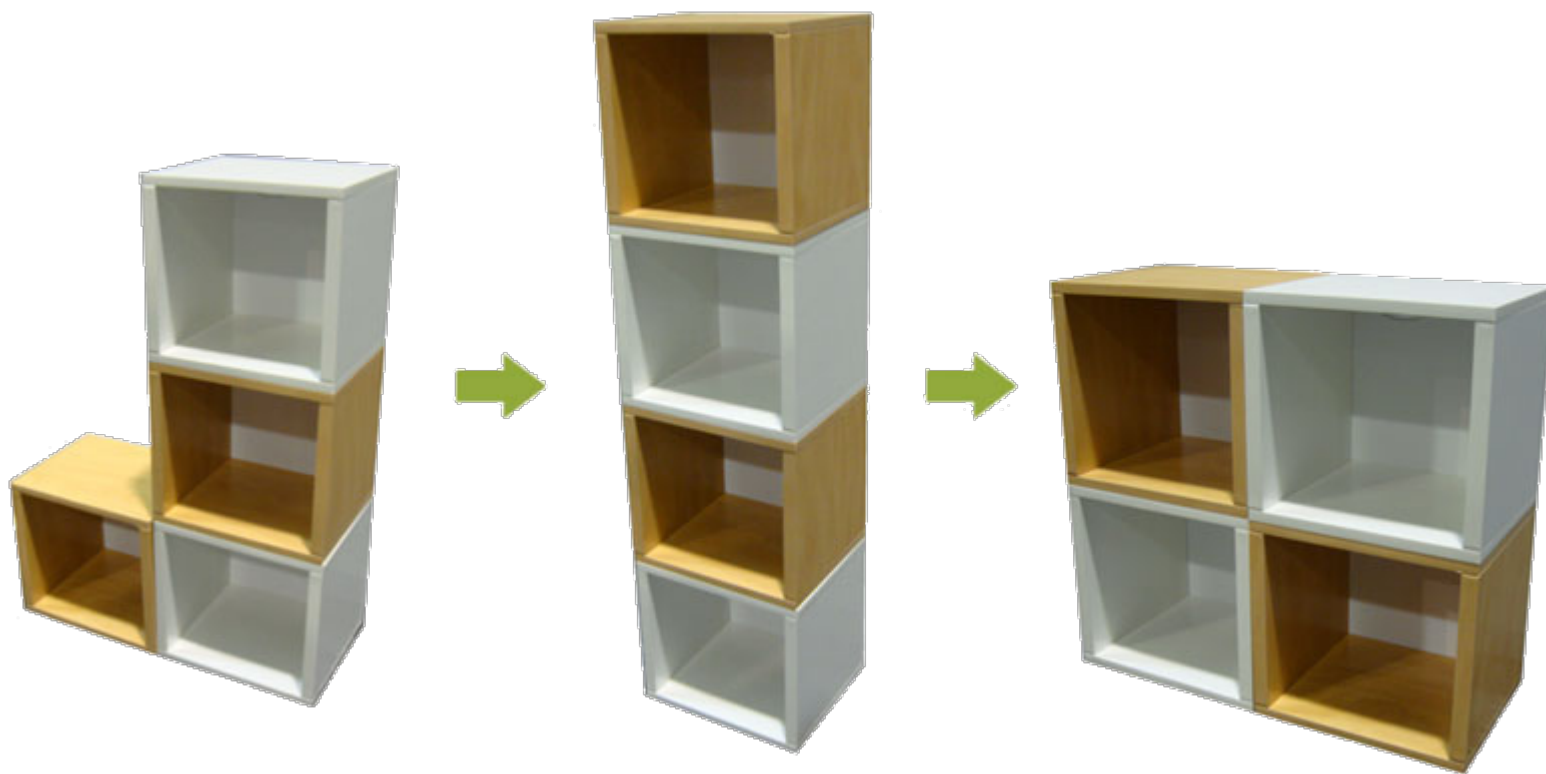
Modularização – parte I

Professor: Ritomar Torquato

04 Modularização

Objetivos: Compreender a necessidade da criação de sub-rotinas para a organização do código fonte em módulos.

Modularização



Modularização

- Facilitar a construção de grandes programas, dividindo em **módulos**.
 - Programas são grandes!
- Também chamados de:
 - Subprograma;
 - Sub-rotina;
 - Função;
 - Procedimento; ou
 - Método

Modularização

- Em termos Gerais: Módulo ou Sub-Rotina

```
tipo nomeDoModuloAuxiliar ([parâmetros]) {  
  
    [retorna valor]; //Opcional  
}
```

```
tipo moduloPrincipal ([parâmetros]) {  
    nomeDoModuloAuxiliar ([valores]);  
    [retorna valor];  
}
```

Modularização

- Em termos de Python

```
def funcaoAuxiliar ([parâmetros]):  
    #Não faz nada
```

```
funcaoAuxiliar([valores])
```


Modularização

```
def AloMundo() :  
    print("Alo Mundo!")
```

```
AloMundo()
```

Modularização

- Parâmetros: Entrada de dados para funções



```
def AloNome (nome) :  
    print ("Alo " + nome + " !")
```

```
AloNome ("Maria")
```


Modularização

- Funções com retorno: Calcula e retorna;
 - Normalmente não ler nada do usuário e não mostra nada na tela;
 - Recebe os dados necessários por parâmetros;
 - Retorna usando **return**

Modularização

- Funções com retorno: Calcula e retorna;

```
def dobro(x):  
    return x * 2
```

```
a = int(input("Digite um numero inteiro: "))  
a = dobro(a)  
print("O dobro é: ", a)
```

Variáveis Locais e Globais

- Todas as variáveis em Python são local por padrão;
 - Para acessar uma variável global, usa-se a palavra reservada global
 - Não é recomendável uso de variáveis globais

Variáveis Locais e Globais

```
x = 42
```

```
def func():  
    x = 1  
    print ("Dentro da função:", x) # Imprime 1
```

```
func()  
print ("Fora da função:", x) #Imprime 42
```

Variáveis Locais e Globais

```
x = 42
```

```
def func():  
    global x  
    x = 1  
    print ("Dentro da função:", x) # Imprime 1
```

```
func()  
print ("Fora da função:", x) #Imprime 1
```

Prática

- Crie uma função que mostra na tela os passos para trocar uma lâmpada. Teste sua execução.
- Crie uma função que mostra na tela os passos para se trocar o pneu de um carro. Teste sua execução.

Prática

- Crie uma função para calcular o resultado de uma equação do primeiro grau ($y = ax + b$). Teste sua execução
- Crie uma função para calcular área de um quadrado ($S = \text{lado}^2$). Teste sua execução