



INSTITUTO  
FEDERAL  
Rio Grande  
do Sul

Desenvolvi-  
mento de  
Sistemas II

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Módulos

Pacotes

Subpacotes

# Desenvolvimento de Sistemas II

Prof. Me. Cleber Schroeder Fonseca

Curso Técnico em Informática para Internet  
IFRS Rio Grande

2024

## ① Módulos

## ② Pacotes Subpacotes



INSTITUTO  
FEDERAL  
Rio Grande  
do Sul

## Desenvolvimento de Sistemas II

Prof. Me.  
Cleber  
Schroeder  
Fonseca

### Módulos

#### Pacotes

Subpacotes

## ① Módulos

## ② Pacotes Subpacotes

Em Python, um módulo é um arquivo contendo definições de funções, classes e variáveis que podem ser usadas em outros programas.

É uma forma de reutilizar código e organizar projetos em partes menores e mais gerenciáveis.

Então um módulo nada mais é que um arquivo `.py` do Python contendo uma série de funções que podem ser chamadas em outros arquivos para serem reutilizadas.

Por exemplo podemos criar o arquivo `fibonacci.py` com o seguinte código:

```
1 # Fibonacci numbers module
2
3 def fib(n):    # write Fibonacci series up to n
4     a, b = 0, 1
5     while a < n:
6         print(a, end=' ')
7         a, b = b, a+b
8     print()
9
10 def fib2(n):  # return Fibonacci series up to n
11     result = []
12     a, b = 0, 1
13     while a < n:
14         result.append(a)
15         a, b = b, a+b
16     return result
```

Em seguida onde se deseja utilizar alguma das funções do nosso módulo devemos utilizar o comando `import` que permite a importação de documentos externos no nosso programa.

E passamos logo em seguida do comando o nome do arquivo Python que queremos importar sem a extensão `.py`, então se formos importar o módulo `fibo.py` que criamos no slide anterior usariamos o comando:

```
import fibo
```

Ao fazer isso as funções Python que foram criadas no arquivo `fibo.py` ficam disponíveis para utilização no arquivo onde o comando foi utilizado.

Por exemplo:

```
1 import fibo
2
3 fibo.fib(120)
```



Nesse caso ao chamar a função ao invés de simplesmente chamar o nome da função devemos informar o nome do módulo juntamente como o nome da função:

```
fibonacci.fib()
```

Assim para facilitar e permitir a chamada somente do nome da função podemos utilizar um comando que diz especificamente cada função que queremos importar para isso devemos utilizar o comando:

```
1 from fibonacci import fib, fib2
2
3 fibonacci.fib(100) # imprime 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
```

Uma alternativa quando queremos importar todas as funções de um módulo é utilizar o caractere “\*” após o import.

```
1 from fibo import *  
2  
3 fib(120)
```

Em alguns casos pode ser de interesse dar um alias (apelido) para um módulo ou função no momento da importação para tanto basta inserir a palavra “**as**” seguida do alias a ser utilizado.

```
1 import fibo as fb
2
3 fb.fib(120)
```

Essa técnica do alias também pode ser usada nas funções.

```
1 from fibo import fib as f
2
3 f(120)
```

A função embutida `dir()` é usada para apresentar quais funções são definidas por um módulo. Ela devolve uma lista ordenada de strings contendo as funções e demais informações sobre o pacote:

```
1 import fibo
2
3 dir(fibo) #imprime ['__builtins__', 'fib', 'fib2']
```



INSTITUTO  
FEDERAL  
Rio Grande  
do Sul

## Desenvolvimento de Sistemas II

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Módulos

**Pacotes**

Subpacotes

### ① Módulos

### ② Pacotes Subpacotes

Os pacotes são uma maneira de estruturar o “espaço de nomes” dos módulos Python, usando “nomes de módulo com pontos”. Por exemplo, o nome do módulo A.B designa um submódulo chamado B, em um pacote chamado A.

Assim como o uso de módulos evita que os autores de módulos diferentes tenham que se preocupar com nomes de variáveis globais, o uso de nomes de módulos com pontos evita que os autores de pacotes com muitos módulos, como NumPy ou Pillow, tenham que se preocupar com os nomes dos módulos uns dos outros.

Eles são basicamente diretórios contendo um arquivo especial chamado `__init__.py` e outros módulos Python.



Para criar um pacote em Python, basta criar um diretório com o nome desejado e dentro dele incluir o arquivo `__init__.py`.

Esse arquivo é executado quando o pacote é importado, permitindo que você defina variáveis, funções, classes ou outros módulos que podem ser usados pelo restante do pacote.

Por exemplo, suponha que você tenha um pacote chamado "meupacote" com os seguintes arquivos:

```
meupacote/  
├── __init__.py  
├── modulo1.py  
└── modulo2.py
```

Para chamar uma função soma no modulo1.py temos de chamar importar da seguinte forma:

```
1 from meupacote.modulo1 import soma
2
3 soma(1, 2) # imprime 3
```

Podemos fazer uma manobra para que não precisemos chamar o `modulo1` na importação da função, para isso basta que utilizemos o arquivo `__init__.py` para deixar essas funções visíveis no `meupacote` diretamente.

```
1 from .modulo1 import soma
2 from .modulo2 import classe_modulo2
```

Assim a utilização da função pode ser feita assim:

```
1 from meupacote import soma
2
3 soma(1, 2) # imprime 3
```

## ① Módulos

## ② Pacotes Subpacotes

Os pacotes também podem ter subpacotes, que são simplesmente diretórios dentro do pacote com seu próprio `__init__.py`.

A hierarquia de pacotes pode ser tão profunda quanto necessário para organizar da maneira desejada.

```
meupacote/  
├── __init__.py  
├── module1.py  
├── subpacote1/  
│   ├── __init__.py  
│   └── modulo2.py  
└── subpacote2/  
    ├── __init__.py  
    └── modulo3.py
```

Com isso, podemos importar qualquer módulo ou subpacote usando a sintaxe de pontos:

```
1 from meupacote import modulo1
2 from meupacote.subpacote1 import modulo2
3 from meupacote.subpacote2.modulo3 import minha_função
```

Assim podemos reaproveitar de maneira mais fácil partes de código desenvolvidas.





INSTITUTO  
FEDERAL  
Rio Grande  
do Sul

Desenvolvimento de  
Sistemas II

Prof. Me.  
Cleber  
Schroeder  
Fonseca

Módulos

Pacotes

Subpacotes

# MUITO OBRIGADO!

Cleber Schroeder Fonseca

<http://ifrs.edu.br/riogrande>

[profcleberfonseca@gmail.com](mailto:profcleberfonseca@gmail.com)

[cleber.fonseca@riogrande.ifrs.edu.br](mailto:cleber.fonseca@riogrande.ifrs.edu.br)