

Python - Pacotes

Em Python, módulo é um script Python com extensão .py e contém objetos como classes, funções, etc. Pacotes em Python ampliam ainda mais o conceito de abordagem modular. Pacote é uma pasta que contém um ou mais arquivos de módulo; além disso, um arquivo especial "__init__.py" que pode estar vazio, mas pode conter a lista de pacotes.

Vamos criar um pacote Python com o nome mypackage. Siga as etapas abaixo -

- Crie uma pasta externa para armazenar o conteúdo de mypackage. Deixe seu nome ser packagedemo.
- Dentro dela, crie outra pasta mypackage. Este será o pacote Python que iremos construir. Dois módulos Python areafunctions.py e mathfunctions.py serão criados dentro de mypackage.
- Crie um arquivo "__init__.py" vazio dentro da pasta mypackage.
- Dentro da pasta externa, armazenaremos posteriormente um script Python example.py para testar nosso pacote.

A **estrutura do arquivo/pasta** deve ser conforme mostrado abaixo -

PackageDemo

- example.py
- testpackage.py
- setup.py
- mypackage
 - __init__.py
 - areafunctions.py
 - mathfunctions.py

Usando seu editor de código favorito, salve os dois módulos Python a seguir na pasta mypackage.

```
# mathfunctions.py
def sum(x,y):
    val = x+y
    return val
```



```
def average(x,y):
    val = (x+y)/2
    return val

def power(x,y):
    val = x**y
    return val
```

Crie outro script Python -

```
# areafunctions.py
def rectangle(w,h):
    area = w*h
    return area

def circle(r):
    import math
    area = math.pi*math.pow(r,2)
    return area
```

Vamos agora testar o pacote **myexample** com a ajuda de um script Python acima desta pasta do pacote. Consulte a estrutura de pastas acima.

```
#example.py
from mypackage.areafunctions import rectangle
print ("Area :", rectangle(10,20))

from mypackage.mathsfunctions import average
print ("average:", average(10,20))
```

Este programa importa funções de **mypackage** . Se o script acima for executado, você deverá obter o seguinte **resultado** -

```
Area : 200
average: 15.0
```

Definir lista de pacotes

Você pode colocar funções selecionadas ou quaisquer outros recursos do pacote no arquivo "__init__.py". Vamos colocar o seguinte código nele.

```
from .areafunctions import circle
from .mathsfunctions import sum, power
```



Para importar as funções disponíveis deste pacote, salve o seguinte script como `testpackage.py`, acima da pasta do pacote como antes.

```
#testpackage.py
from mypackage import power, circle

print ("Area of circle:", circle(5))
print ("10 raised to 2:", power(10,2))
```

Ele produzirá a seguinte **saída** -

```
Area of circle: 78.53981633974483
10 raised to 2: 100
```

Instalação do pacote

No momento, podemos acessar os recursos do pacote a partir de um script logo acima da pasta do pacote. Para poder usar o pacote em qualquer lugar do sistema de arquivos, você precisa instalá-lo usando o utilitário PIP.

Primeiro de tudo, salve o seguinte script na pasta pai, no nível da pasta do pacote.

```
#setup.py
from setuptools import setup
setup(name='mypackage',
      version='0.1',
      description='Package setup script',
      url='#',
      author='anonymous',
      author_email='test@gmail.com',
      license='MIT',
      packages=['mypackage'],
      zip_safe=False)
```

Execute o utilitário PIP no prompt de comando, permanecendo na pasta pai.

```
C:\Users\user\packagedemo>pip3 install .
Processing c:\users\user\packagedemo
Preparing metadata (setup.py) ... done
Installing collected packages: mypackage
Running setup.py install for mypackage ... done
Successfully installed mypackage-0.1
```



Agora você poderá importar o conteúdo do pacote em qualquer ambiente.

```
C:\Users>python
```

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import mypackage
```

```
>>> mypackage.circle(5)
```

```
78.53981633974483
```