

# Python Package

---

Advanced Institute for Artificial Intelligence

<https://advancedinstitute.ai>

- ☐ Introdução
- ☐ Estrutura de um pacote
- ☐ Procedimento para criação de um pacote
- ☐ ferramenta `setuptools.py`

## ☐ Module

- A unidade básica de reutilização de código em Python.

## ☐ (Import) Package

- Um módulo Python que pode conter outros módulos ou recursivamente outros pacotes.

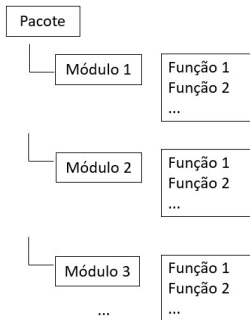
## ☐ (Distribution) Package

- Um arquivo versionado que contém pacotes, módulos e arquivos de recurso Python. Usado para distribuir um Release. O que os usuários baixam da Internet e instalam.

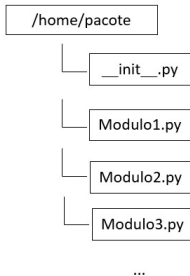
- ❑ Um diretório contendo diversos *scripts* python, que implementam um conjunto de funcionalidades para o mesmo fim, podem ser considerados um módulo
- ❑ Por exemplo, um conjunto de *scripts* python para manipular acesso a arquivos.
- ❑ Módulos pode ser ditribuídos livremente utilizando setuptools

# Package

## Estrutura de Pacotes e Módulos



## Estrutura de Diretórios e Arquivos



## Exemplo de Uso

```
Import pacote.modulo  
  
Funcao1()  
Funcao2()
```

- ❑ Para criar um pacote de distribuição é necessário organizar os módulos em um diretório
- ❑ No mesmo diretório é necessário incluir um arquivo vazio com o seguinte nome:  
    `--init--.py`
  - O nome do pacote é igual ao nome da pasta
  - O nome de cada módulo é igual ao nome de cada script python da pasta

Para reutilizar os pacotes é necessário utilizar o comando **import**

- A estrutura do **import** é o nome do **pacote** . **nome** do módulo
  - exemplo: `import package.math`
- A partir daí é possível utilizar quaisquer métodos do módulo

Instalando pacote no sistema

- ☐ O pacote pode ser instalado no lista de pacotes pip
- ☐ Os módulos do pacote ficam disponíveis a partir de qualquer pasta do sistema

Depende do pacote `setuptools.py`



Para instalar um pacote no ambiente conda deve seguir os seguintes passos:

- ☐ criar o instalador de pacote **setup.py**
- ☐ instalar a partir do fonte usando **pip install**

# Package

Exemplo de setup.py

```
from setuptools import setup
setup(name='package',
      version='0.1',
      description='Testing installation of Package',
      url='#',
      author='Silvio',
      author_email='silvio.stanzani@sprace.org.br',
      license='N/D',
      packages=['package'],
      zip_safe=False)
```

Definindo dependências de outros pacotes no setup.py

```
install_requires=[  
    'pandas',  
],
```

Em seguida instalar o pacote no ambiente usando pip

```
pip install .
```

# Package

Verificando o pacote instalado no ambiente

```
(package) silvio@top04:~/usingapp$ conda list
# packages in environment at /home/silvio/.conda/envs/package:
#
# Name                          Version                      Build      Channel
package                         0.1                         pypi_0     pypi
pandas                         0.25.2                      pypi_0     pypi
pip                             19.3.1                      py37_0
python                         3.7.4                       h265db76_1
```

Verificar informação a respeito do pacote instalado:

```
pip info package
```

- ❑ Para distribuir mais amplamente, é possível publicar no repositório pip
- ❑ Usuários podem pesquisar sua biblioteca via **pip search**
- ❑ Usuários podem instalar sem precisar obter o código fonte
- ❑ Para isso, seguir o guia de referência a seguir:
  - <https://packaging.python.org/tutorials/packaging-projects/>