Introdução a Python

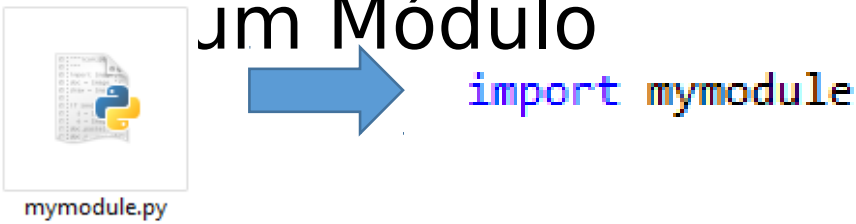
Cristian Enrique Munoz Villalobos

Porque Python?(lembrando ...)

- **Qualidade**: Focado na legibilidade, coerência e qualidade de software.
- **Portabilidade do Programa**: Programas de python na maioria de plataformas: Windows, Linux, MAC OS, etc.
- **Bibliotecas de apoio**: Python vem com uma grande coleção de funcionalidade pré-compiladas, conhecidas como Standard Library.
- **Componentes de Integração**: Scripts de Python podem facilmente comunicar-se com outras partes de uma aplicação, usando uma variedade de mecanismos de integração.
- **Prazer** ...

Módulos

- Um módulo é um **arquivo de python** que (geralmente) tem só definições de **variáveis, funções** e **classes.**
- Alguns módulos de Python são escritos em linguagem diferentes de Python, comumente C ou C++. Estes módulos são chamados de **Módulos de extensão.**
- Para imp[illegible]um Módulo



Módulos

- Para acessar funções e outros items dentro do módulo:

```
mymodule.myfunction(x)
```

- Outras formas de acessar aos atributos do modulo são:

```
from mymodule import myfunction # Um item especifico
from mymodule import *          # Todas as funções do Módulo
```

- Para Importar um módulo usando um nome diferente:

```
import mymodule as mymod
```

Módulos

- Alguns Módulos Básicos
  - sys
  - os
  - math
  - random
  - Date Time, subprocess ...
- Alguns Módulos Avançados
  - Módulos para suporte a decisão

Módulo **“sys”**

- O módulo **sys** é para controlar e interatuar com o interpretador Python. Ele inclui toda a informação sobre o **sistema operativo** e a **versão de python** que está sendo utilizada.
- Para utilizar o módulo escreve-

```
import sys
```

```
sys.path # Path de Python
sys.version # Versão do interpretador Python
sys.platform # Identifica Plataforma (win32, Linux)
sys.argv # Linha de argumentos
sys.exit() # Sair do Interpretador
```

- Para listar todas as funções presentes no módulo escrever:

```
help(sys)
```

sys.argv dentro de python(x,y)

Módulo **“os”**

- O módulo **os** permitem a python trabalhar com seu sistema operativo.
  - Trabalhar com path e permissão
  - Trabalhar com arquivos

- Alaumas funcões usadas no Modulo **os**

```
os.getcwd() # Get Current Work Directory
os.chdir('D:\Samples') # Change the current working directory
os.mkdir() # Create a directory
os.remove() # Remove file
os.rmdir() # Remove directory
```

Módulo **“os”**

- O **“os.path”** é um **submódulo de “os”** que fornece uma variedade de métodos para manipular arquivos e diretórios. Alguns métodos usados:

```
os.path.join('D:\Samples','arquivo1.txt') # 'D:\\Samples\\arquivo1.txt'

# Work Directory 'D:\\Samples\\ImportData'
os.path.abspath('arquivo1.txt') # 'D:\\Samples\\ImportData\\arquivo1.txt'
```

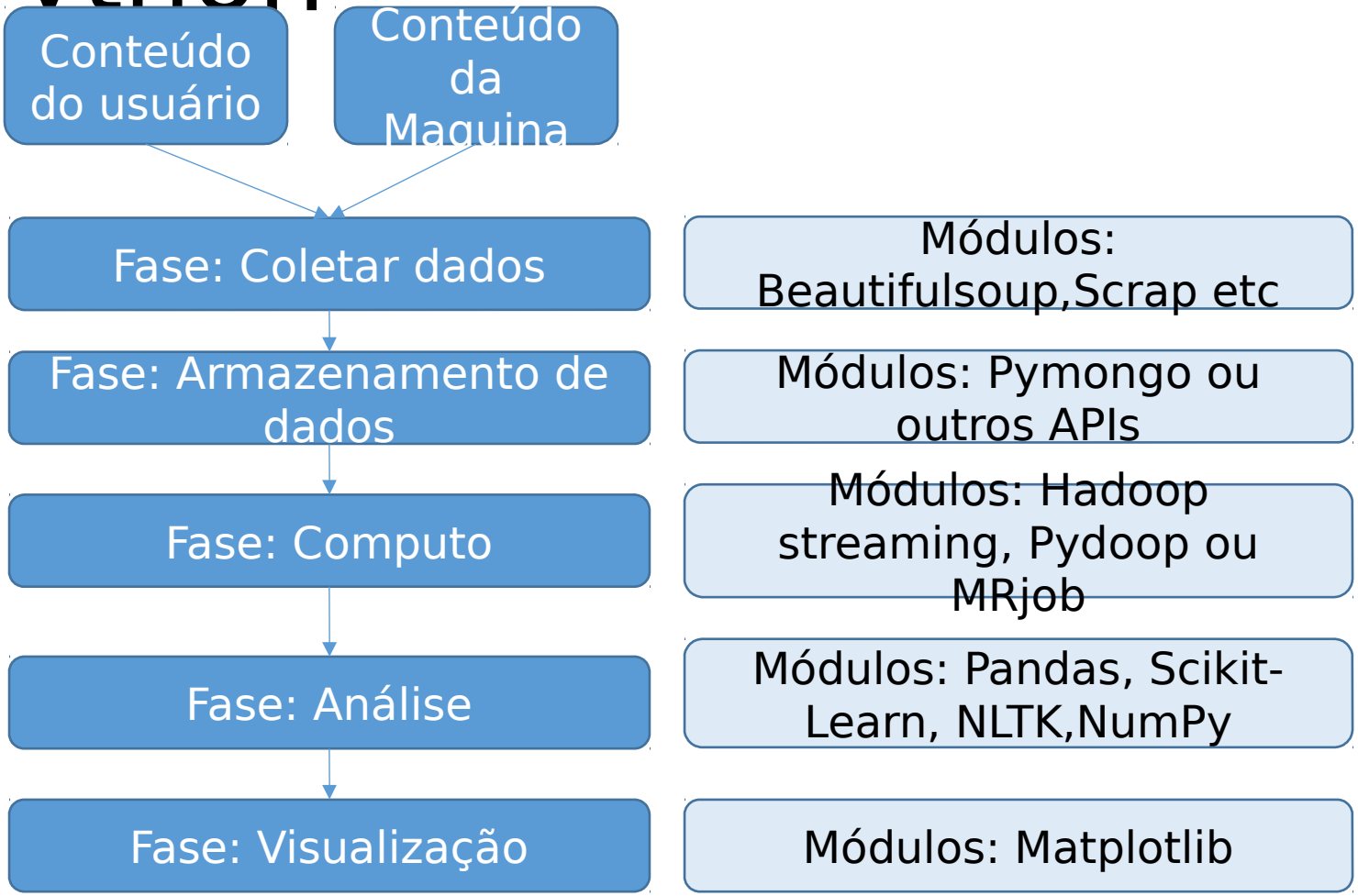
Módulo **“math”**

- O módulo math contem varias funções matemáticas predefinidas.
  - Teoria de números:
    - **ceil, fabs**
  - Potencia e funções logarítmicas:
    - **exp, log**
  - Funções trigonométricas:
    - **acos, asin, atan, cos**
  - Conversão Angular e Constantes:
    - **degrees, radians, pi, e**

Módulo **“Random”**

- Este módulo implementa um gerador de números pseudoaleatórios para varias distribuições.
- Funções para inteiros
  - Retorna um inteiro aleatório dentro do rango dado
    - **Random.randrange(stop)**
  - Retorna um inteiro aleatório dentro do rango(start,stop,setp)
    - **Random.randrange(start,stop[,step])**
  - Retorna um inteiro N: a<N<b
    - **Random.randint(a,b)**

Suporte a decisão com Python



Módulos Avançados

- Armazenamento de dados :
  - Pymongo, outros DB APIs
- Computo:
  - Hadoop Streaming, Pydoop ou Mrjob
- Analise de dados:
  - Pandas, Scikit-Learn, NLTK, Numpy
- Visualização:
  - Matplotlib