

```
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:13:53)
Type "copyright", "credits" or "license()" for more information

IPython 0.10.1 -- An enhanced Interactive Python.
?      -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help    -> Python's own help system.
object? -> Details about 'object'. ?object also works, ?? prints more.
```

```
In [1]: str?
Type:      type
Base Class: <type 'type'>
String Form: <type 'str'>
Namespace: Python builtin
Docstring:
str(object) -> string

Return a nice string representation of the object.
If the argument is a string, the return value is the same object.
```

```
In [2]: ls
test.tif
```

```
In [3]:
```

```
In [3]:
```

```
In [5]:
```

ASTROINFORMÁTICA I

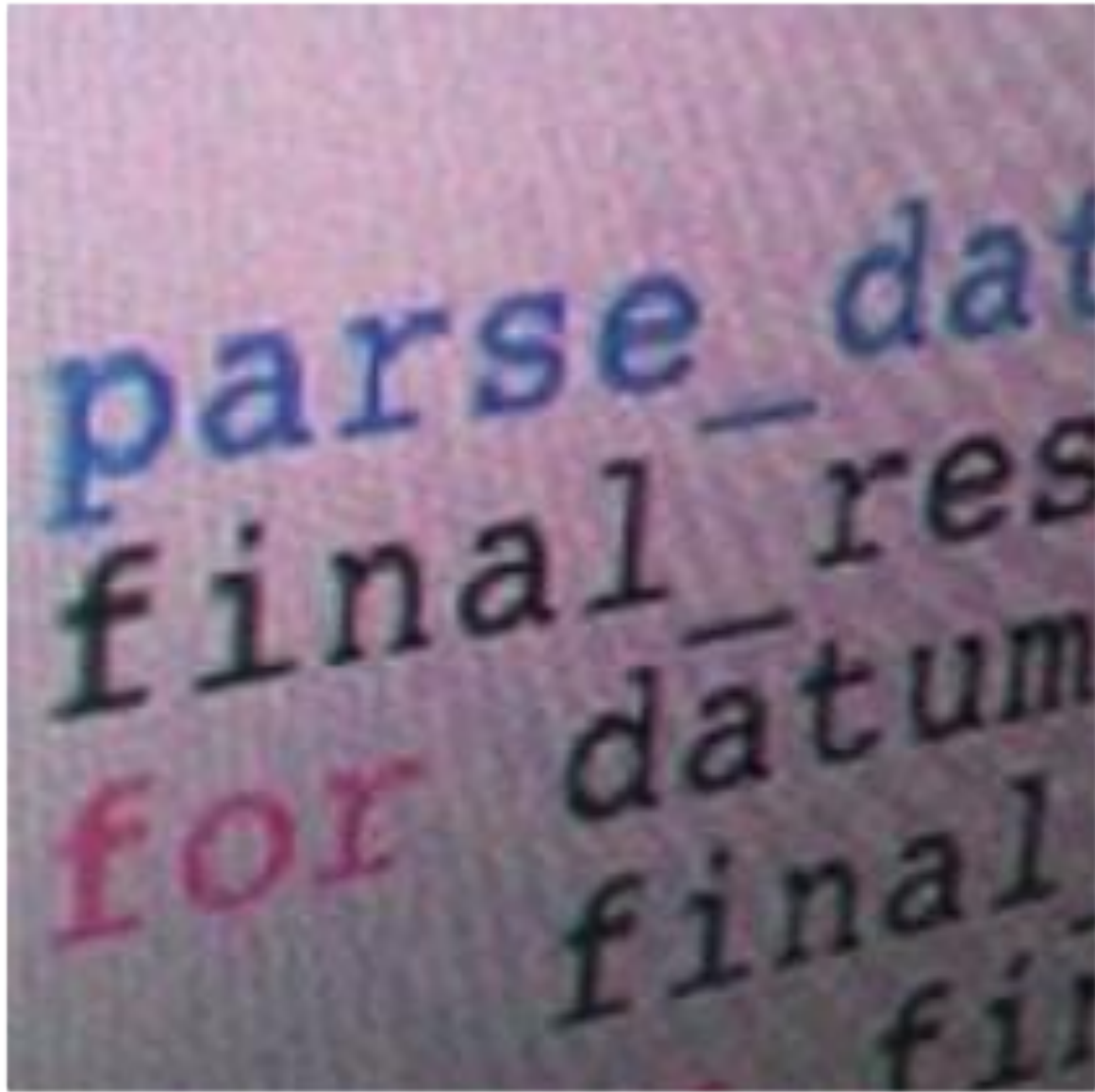
AULA 01

Prof. Dr. Luciano Silva

Centro de Radioastronomia e Astrofísica Mackenzie (CRAAM)

luciano.silva@mackenzie.br

CONTEÚDO



- Apresentar os detalhes administrativos da disciplina
- Conhecer o ambiente de programação a ser utilizado na disciplina (Jupyter)

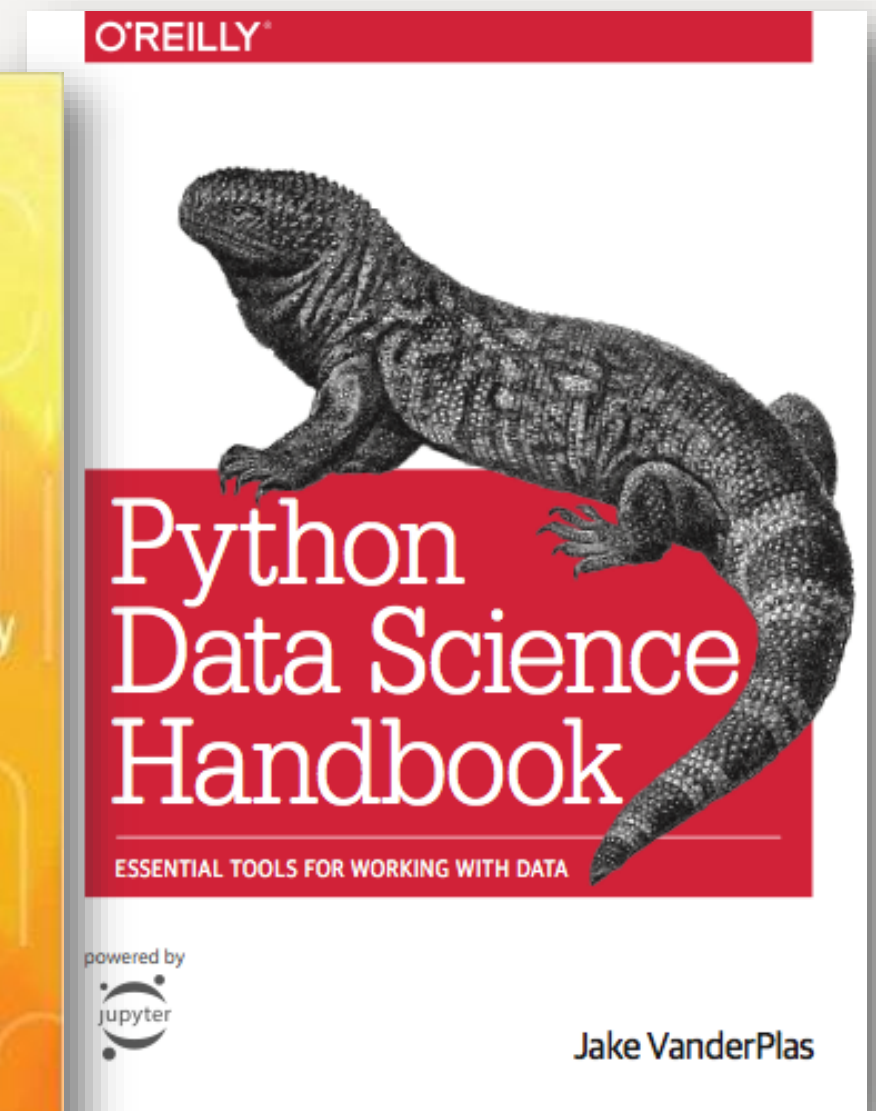
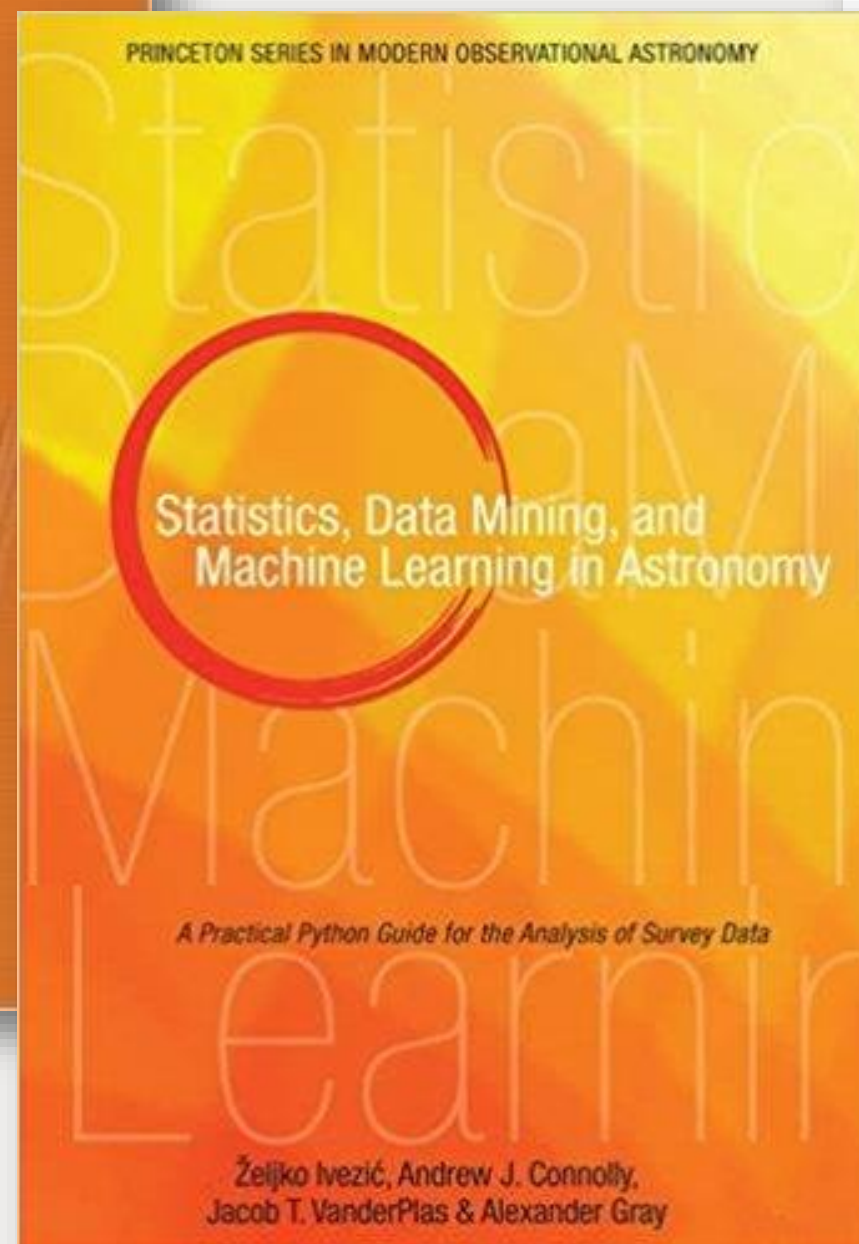
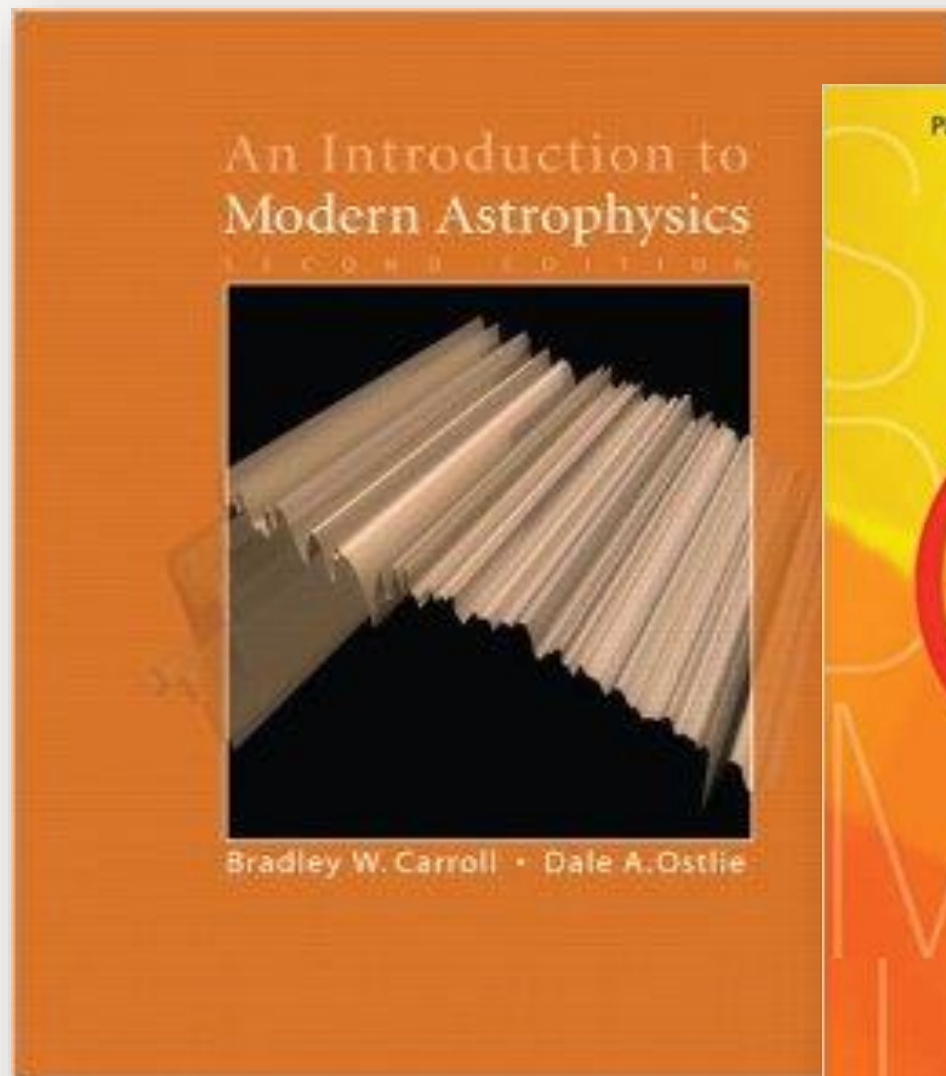
OBJETIVOS

- Conhecer alguns **problemas clássicos** de **análise de sinais** astronômicos e astrofísicos em uma, duas e três dimensões
 - Conhecer e acessar **fontes de dados astronômicas e astrofísicas**
 - Conhecer e praticar com **módulos Python** para **acesso e processamento** de sinais astronômicos e astrofísicos
-

CONTEÚDO

- Fundamentos de programação em Python.
 - Entrada e modelos de dados astronômicos (arquivos genéricos, arquivos FITS, observatórios virtuais).
 - Visualização de dados.
 - Análise de dados unidimensionais.
 - Análise de dados bidimensionais.
 - Análise de dados tridimensionais.
-

REFERENCIAS



AVALIAÇÃO

- **01 artigo** descrevendo algum problema de seu interesse em Astronomia/Astrofísica/Geofísica/Partículas, com uma abordagem via Python (~ 10 páginas) – 70%
 - **Apresentação** do artigo em um mini-workshop a ser realizado no último dia de aula – 30%
-

LINGUAGEM PYTHON

- Guido Van Rossum - 1989
- Instituto de Pesquisa Nacional para Matemática e Ciência da Computação (CWI)
- Sucessor da linguagem ABC
- Aumentar Produtividade do Programador
- Lançamento: 1991
 - Versão 0.90:
 - Módulos de Modula-3
 - Classes, heranças, tratamento de exceções, funções
 - Tipos de dados nativos (list, dict, str)



LINGUAGEM PYTHON

Histórico

- 1994
 - Fórum de discussão `comp.lang.python`
 - Versão 1.0:
 - Adicionado de ferramentas para computação funcional (`lambda`, `map`, `filter` e `reduce`)
- 1995:
 - Corporação para Iniciativas Nacionais de Pesquisa - CNRI (Reston-Estados Unidos)
 - Versão 1.2
 - Versão 1.4:
 - Linguagem ganha a capacidade de passar parâmetro pelo nome



LINGUAGEM PYTHON

Histórico

- *Ganha forma de encapsulamento*
- *Suporte Nativo a número complexos*
- *Versão 1.6*
- *Versão 1.61:*
 - *Licença CNRI*
 - *Compatível com GPL*
- *2000:*
 - *BeOpen*
 - *Versão 2.0:*
 - *Listcomprehension, sistema coletor de lixo*
 - *Analizador Sintático SAX*



LINGUAGEM PYTHON

Histórico

- *2001:*
 - *PythonLabs → Digital Creations*
 - *Python Software Foundation*
 - *Versão 2.1*
- *Versão 2.2:*
 - *Unificação do tipos de Python (escritos em C) e classes (escritas em Python) em somente uma hierarquia*
 - *Modelo de Objetos Python consistentemente orientado a objeto*



LINGUAGEM PYTHON

Histórico

- *Versão 2.3:*
 - *Pacote logging*
- *Versão 2.4*
- *Versão 2.6:*
 - *Transição*
 - *Bibliotecas para multiprocessing, JSON, E/S*
 - *Nova forma de formatação de cadeias de caracteres*
- *Python 3000:*
 - *O print passa a ser uma função*
 - *Mudança de todas as cadeias de caractere para UNICODE*



LINGUAGEM PYTHON

Características

- *Construções*
 - *Estrutura de Seleção*
 - *if, elif, else*
 - *Estrutura de Repetição*
 - *for, while*
 - *Estrutura de classes*
 - *class*
 - *Construção de Subrotina*
 - *def*
 - *Construção de escopo*
 - *with*



LINGUAGEM PYTHON

Características

- *Tipos de dados*
 - *Valores e Objetos possuem tipos bem definidos*
 - *str* → `"Python", 'pet-computacao'`
 - *list* → `['ufcg', 10, True]`
 - *dict* → `{"oi": "hello", "gordo": "big"}`
 - *int* → `13, 55555555555555555555L`
 - *float* → `12.987765432`
 - *complex* → `3+4j`
 - *bool* → `True, False`
 - *set, frozenset* → `set([1, "diogo", "diogo", 13, 13])`



LINGUAGEM PYTHON

Características

- *Permite definição de tipo de dados próprios*
- *Métodos são definidos como funções anexados à classe*
- *Exemplo de criação de classe:*

```
class Bag:  
    def __init__(self):  
        self.data = []  
    def add(self, x):  
        self.data.append(x)  
    def addtwice(self, x):  
        self.add(x)  
        self.add(x)
```



LINGUAGEM PYTHON

Características

- *Palavras reservadas*
 - `and, del, from, not, while, as, elif, global, or, with, assert, else, if, pass, yield, break, except, import, print, class, exec, in, raise, continue, finally, is, return, def, for, lambda, try`
- *Operadores*
 - *Básicos:*
 - `==, >, <, <=, >=`
 - *Lógicos:*
 - `and, or, not`



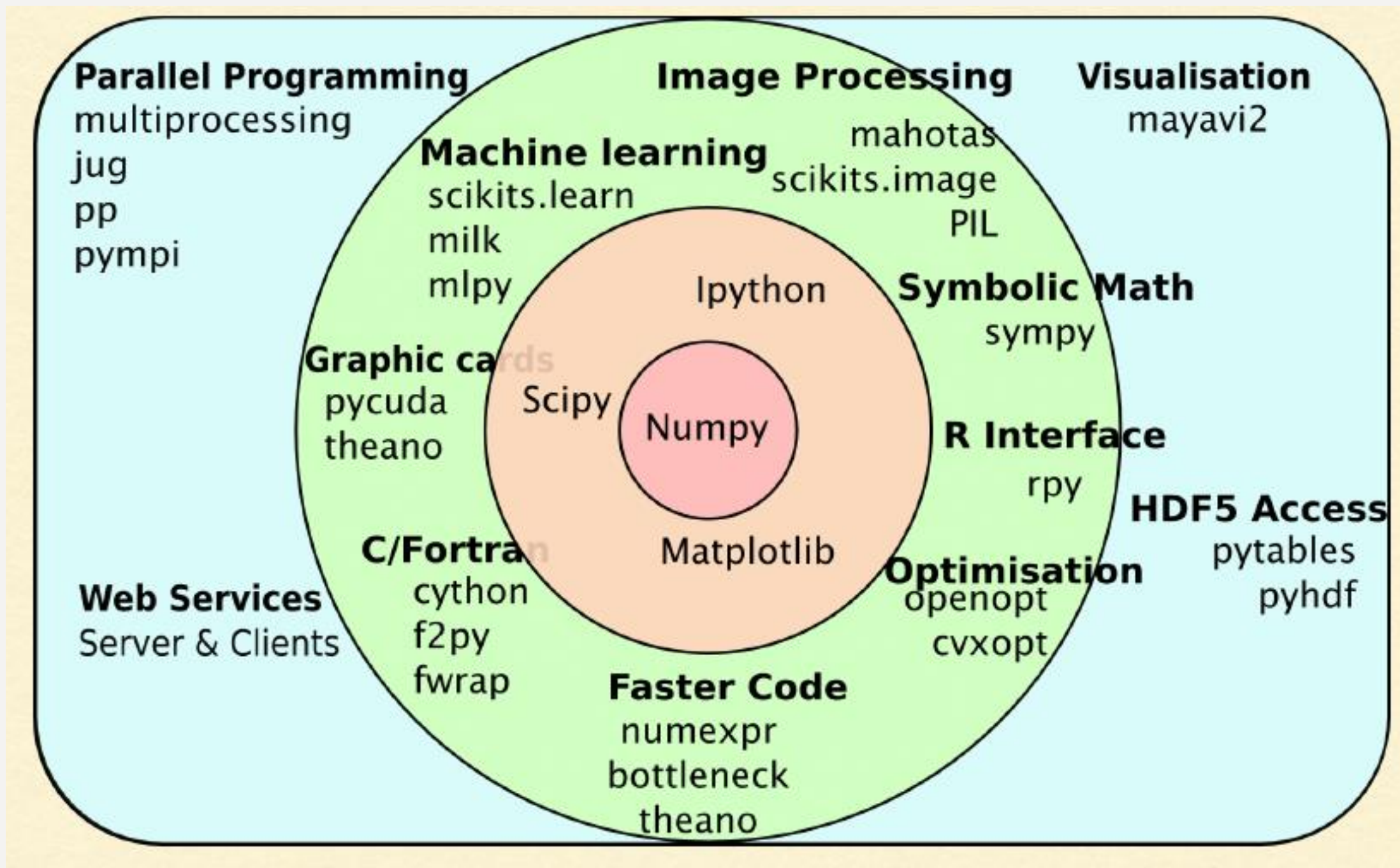
LINGUAGEM PYTHON

Características

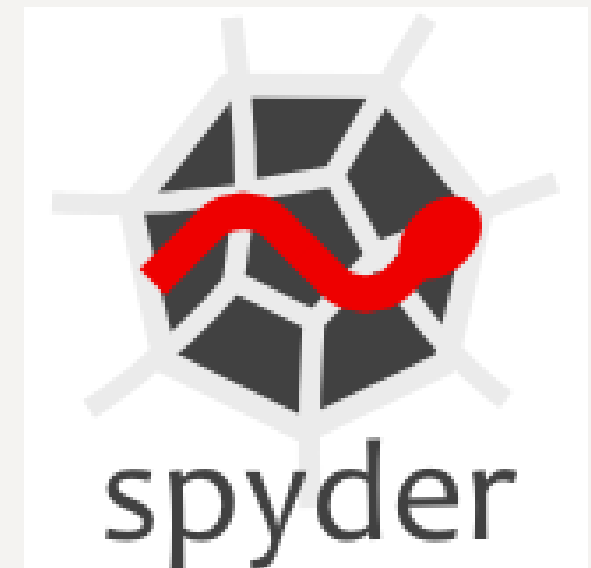
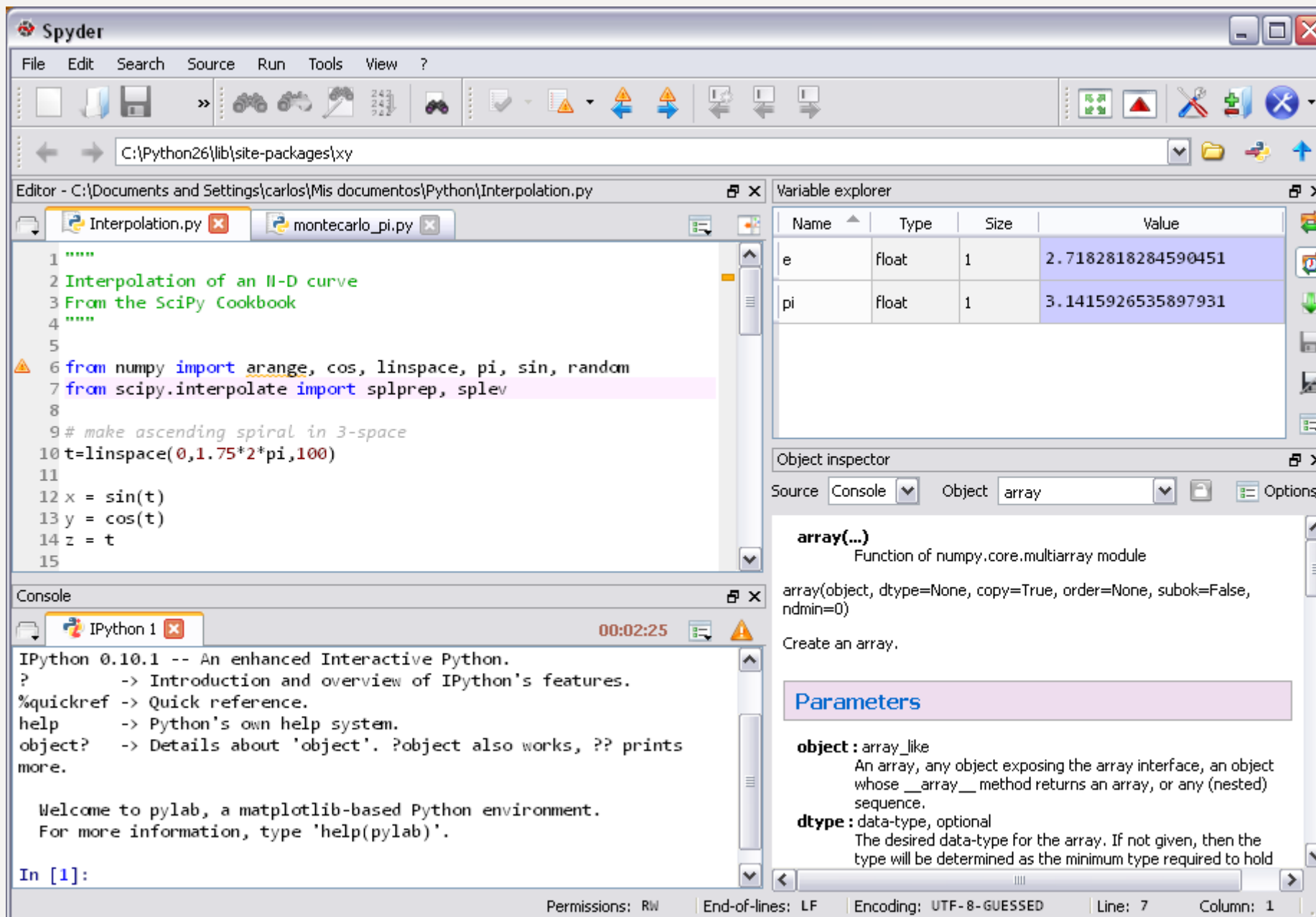
- *Tipagem Dinâmica*
- *Interpretador Interativo*
- *Análise Léxica*
- *Analizador Sintático*
- *Indentação e Espaços em branco*
- *Compilador de Bytecode*
 - *Pode ser compilada*
 - *.pyc e .pyo*
 - *Módulo de Biblioteca*
 - *Grupos de Arquivos*



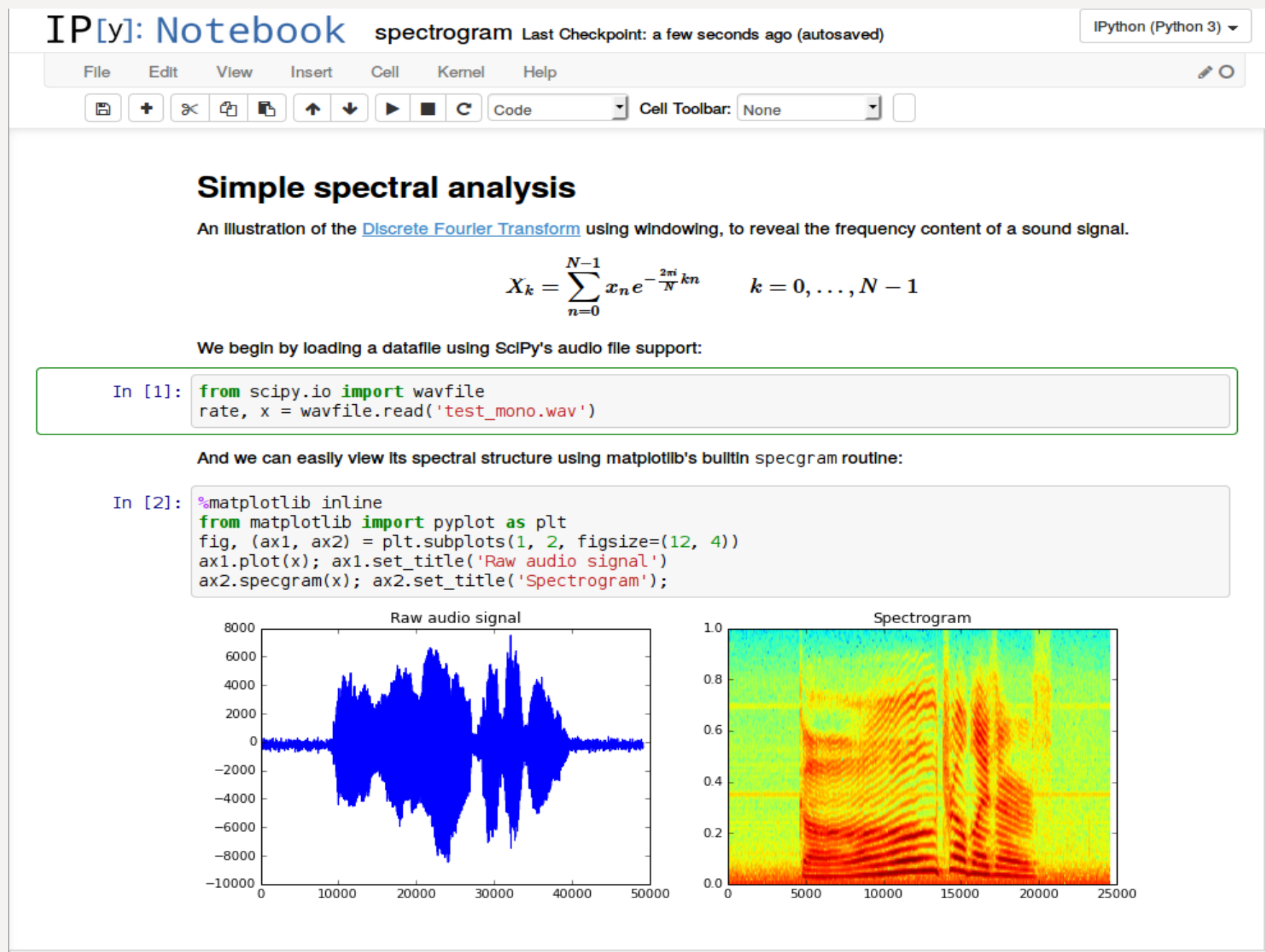
ECOSSISTEMA PYTHON



AMBIENTES PYTHON



AMBIENTES PYTHON



AMBIENTES PYTHON

