

# MC#16: Modelagem gravimétrica utilizando fontes pontuais

## PALESTRANTES:

Victor Ribeiro Carreira - Observatório Nacional (ON-MCTIC)  
([carreiravr@gmail.com](mailto:carreiravr@gmail.com))

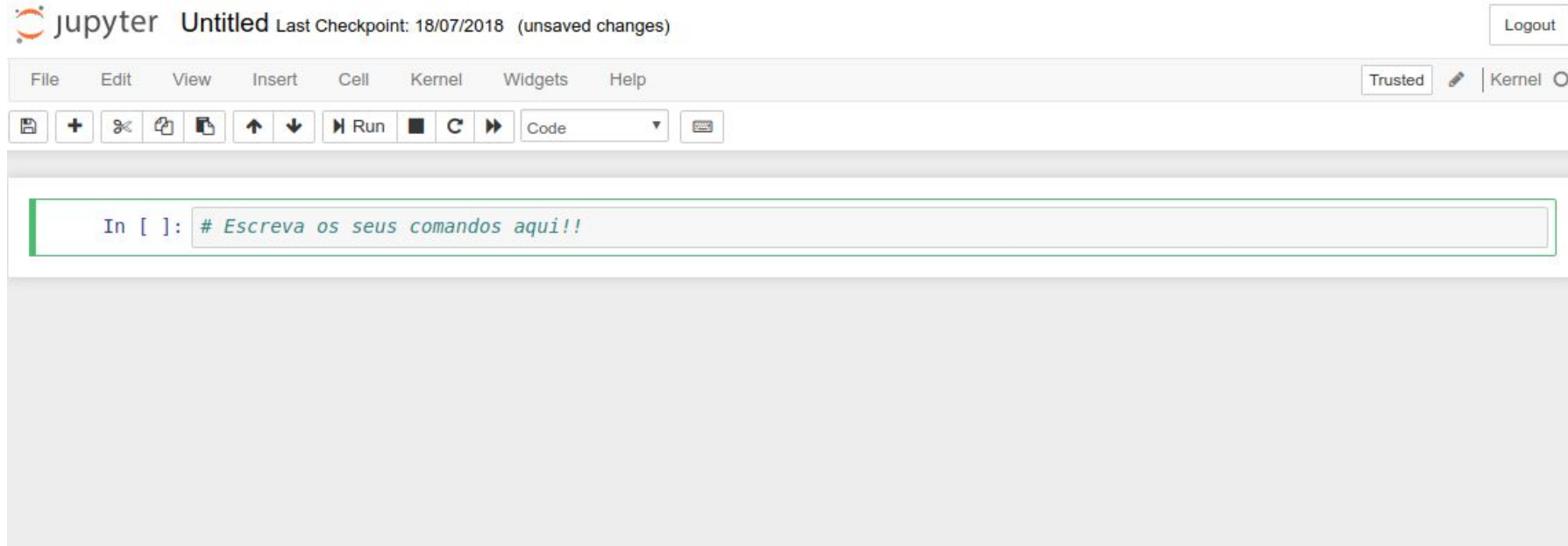
Rodrigo Bijani - Universidade Federal Fluminense (UFF)  
([rodrigobijani@gmail.com](mailto:rodrigobijani@gmail.com))

# Filosofia do curso

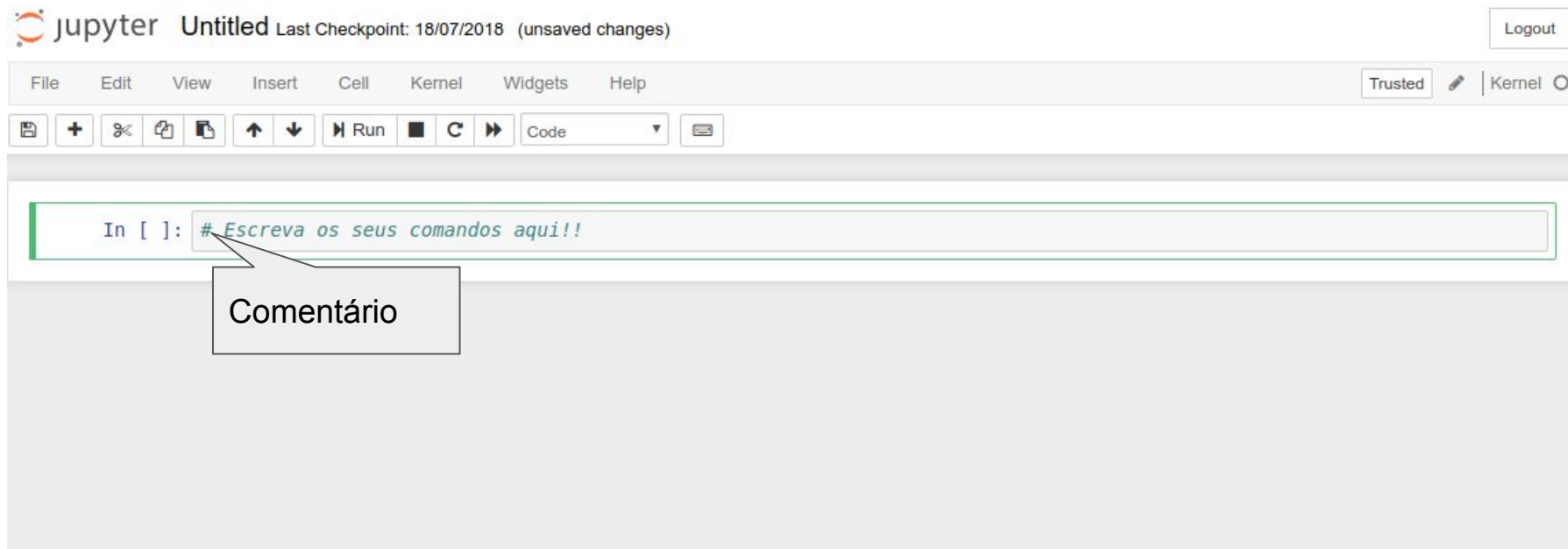


# Conceitos básicos de programação científica

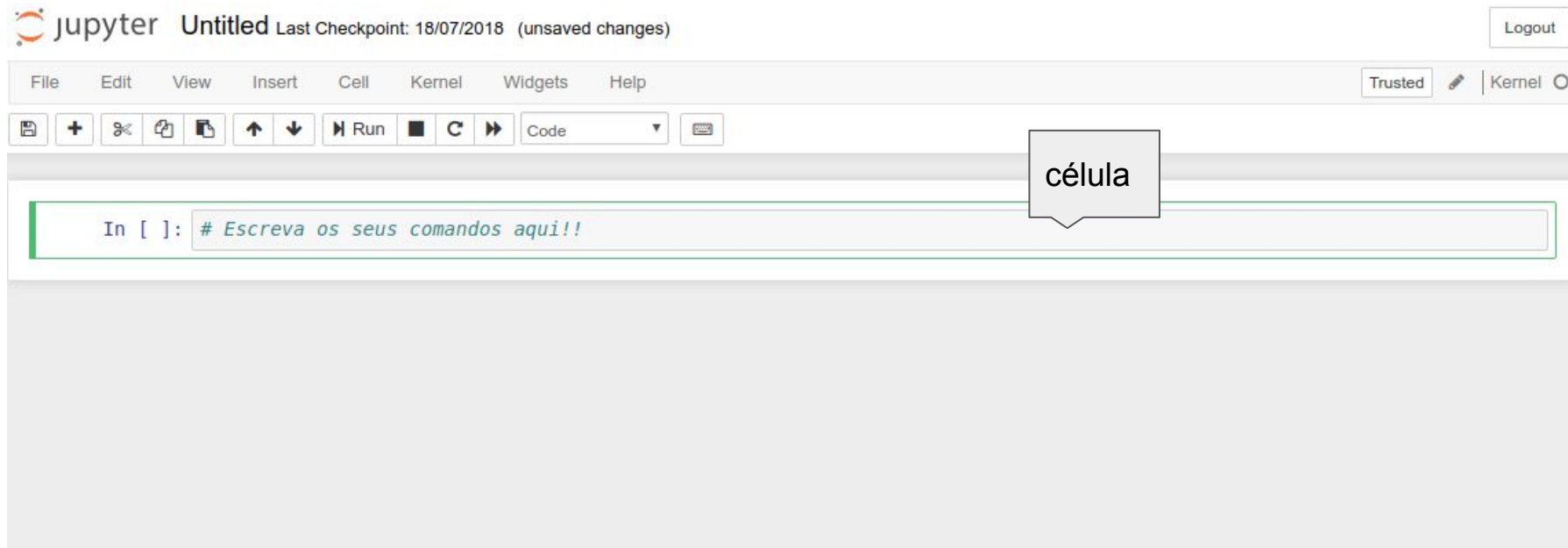
# O ambiente de trabalho: jupyter notebook



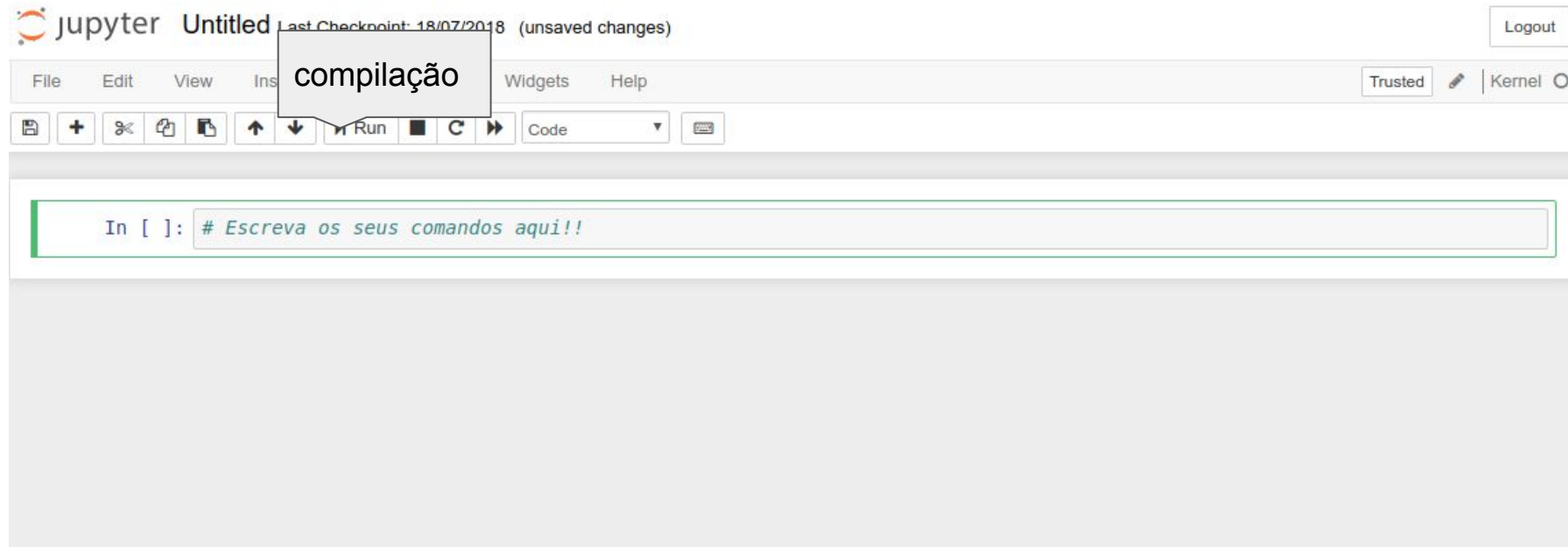
# O ambiente de trabalho: jupyter notebook



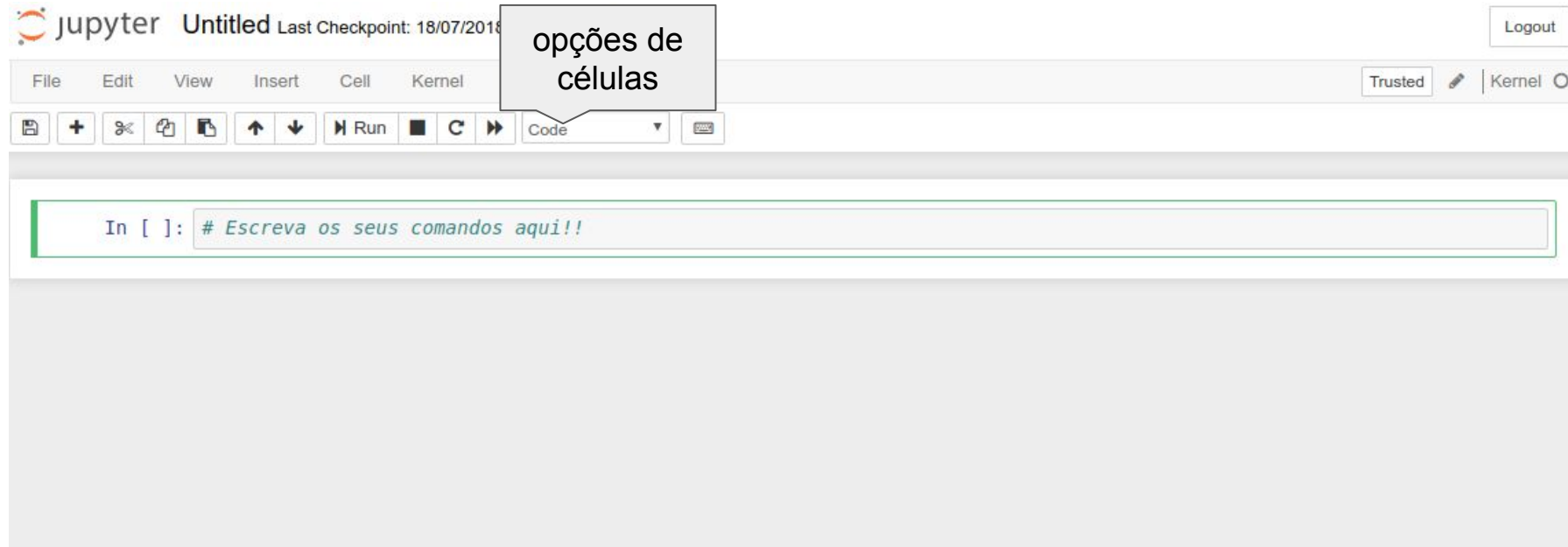
# O ambiente de trabalho: jupyter notebook



# O ambiente de trabalho: jupyter notebook



# O ambiente de trabalho: jupyter notebook





# O ambiente de trabalho: jupyter notebook

vamos ver no código ...

# A linguagem de programação: PYTHON



Por que usar o Python???

# A linguagem de programação: PYTHON



Por que usar o Python???

- linguagem amigável;

# A linguagem de programação: PYTHON



Por que usar o Python???

- linguagem amigável;
- rápida;

# A linguagem de programação: PYTHON



Por que usar o Python???

- linguagem amigável;
- rápida;
- fácil de aprender;

# A linguagem de programação: PYTHON



Por que usar o Python???

- linguagem amigável;
- rápida;
- fácil de aprender;
- multiplataforma;

# A linguagem de programação: PYTHON



Por que usar o Python???

- linguagem amigável;
- rápida;
- fácil de aprender;
- multiplataforma;
- open source;

# A linguagem de programação: PYTHON

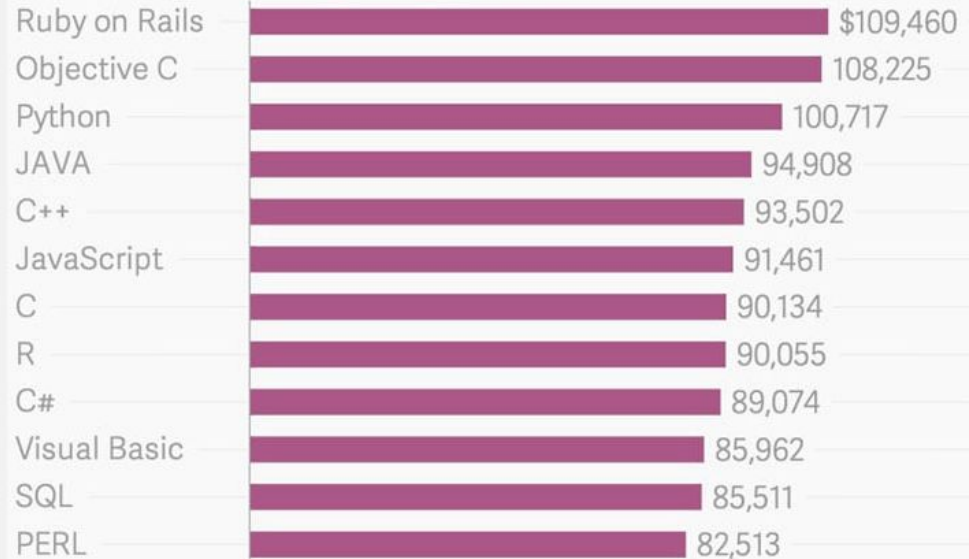


Por que usar o Python???

- linguagem amigável;
- rápida;
- fácil de aprender;
- multiplataforma;
- open source;
- rentável;

## The most valuable programming skills to have on a resume

Average salary value of skill



Quartz | qz.com

Data: Brookings/Burning Glass



# A linguagem de programação: PYTHON



Por que usar o Python???

- linguagem amigável;
- rápida;
- fácil de aprender;
- multiplataforma;
- open source;
- rentável;
- popular;

Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥️	100.0
2. C	📱 🖥️ 🖨️	99.7
3. Java	🌐 📱 🖥️	99.5
4. C++	📱 🖥️ 🖨️	97.1
5. C#	🌐 📱 🖥️	87.7
6. R	🖥️	87.7
7. JavaScript	🌐 📱	85.6
8. PHP	🌐	81.2
9. Go	🌐 🖥️	75.1
10. Swift	📱 🖥️	73.7

Ranking do IEEE

No "TOP 10 geral" a Python foi eleita a mais popular entre os programadores neste ano, seguida pelas linguagens C, Java, C++ e C#. No ano passado, a C foi eleita a linguagem de programação mais popular.

# A linguagem de programação: PYTHON



Por que usar o Python???

- linguagem amigável;
- rápida;
- fácil de aprender;
- multiplataforma;
- open source;
- rentável;
- popular;
- orientada a objeto.

# A linguagem de programação: PYTHON



Por que usar o Python???

- linguagem amigável;
- rápida;
- fácil de aprender;
- multiplataforma;
- open source;
- rentável;
- popular;
- orientada a objeto.



# Objeto

- números;
- funções;
- módulos;

... são todos considerados objetos que usamos dentro da linguagem e não são os únicos.

# Objeto

- números; →
- funções;
- módulos;

Tipo de dado	Descrição	Exemplo
string	cadeia de caracteres imutáveis	'Geologia'
int	número de precisão fixa	47
float	Ponto flutuante ou real	3.1415927
complex	número complexo	3+2j
bool	Booleano	True ou False

# Objeto

- números;
- funções;
- módulos;

→

Bloco organizado de código que pode ser reutilizado para perfazer uma única tarefa.

# Objeto

- números;
- funções;
- módulos;

Bloco organizado de código que pode ser reutilizado para perfazer uma única tarefa.

```
y = 0.0 # variável global
```

```
def reta(a,b):  
    x = 1.0 # variável local  
    y= a*x+b  
    return y;
```

```
y = reta(10,20) # chamando a função
```

```
print "y = ", y
```

# Objeto

- números;
- funções;
- módulos;

O código abaixo retorna 30 como resposta. Mas se quisermos um vetor como resposta como devemos proceder?

```
y = 0.0 # variável global
```

```
def reta(a,b):  
    x = 1.0 # variável local  
    y= a*x+b  
    return y;
```

```
y = reta(10,20) # chamando a função
```

```
print "y = ", y
```



# Objeto

- números;
- funções;
- módulos;

Agora o código abaixo retornará uma lista de 10 valores de números 30.

```
y = [] # variável global
```

```
def reta(a,b):  
    x = 1.0 # variável local  
    for i in range(10):  
        y = a*x[i]+b  
    return y;
```

```
y = reta(10,20) # chamando a função
```

```
print "y = ", y
```

# Objeto

- números;
- funções;
- módulos;


→ Agora o código abaixo retornará uma lista de 10 valores de números 30.

# vamos ver no código ...

# Objeto

- números;
- funções;
- módulos; —————→
- Permite ao usuário organizar logicamente o seu código tornando-o claro e de fácil compreensão.

# Objeto

- números;
  - funções;
  - módulos; 
- Permite ao usuário organizar logicamente o seu código tornando-o claro e de fácil compreensão.
  - Ele é um \*.py separado que pode conter números, variáveis, funções entre outros objetos.

# Objeto

- números;
- funções;
- módulos;

Suponhamos que a *function* do exemplo anterior esteja salva em um arquivo chamado *module.py* ...

Qual seria a sintaxe de chamamento?

```
import module as md
```

```
y = [] # variável global
```

```
y = md.reta(10,20)
```

```
print "y = ", y
```

# Objeto

- números;
- funções;
- módulos;

É notável como conseguimos diminuir as linhas de código!

```
import module as md
```

```
y = [] # variável global
```

```
y = md.reta(10,20)
```

```
print "y = ", y
```

# Objeto

- números;
- funções;
- módulos;

→ É notável como conseguimos diminuir as linhas de código!

# vamos ver no código ...

Quais são os processos físico-matemáticos envolvidos no processo de modelagem geofísica??