

# Python - Configuração do Ambiente

---

Advanced Institute for Artificial Intelligence

<https://advancedinstitute.ai>



# Gerenciador de Ambiente ***Conda***

---

- ❑ <https://www.digitalocean.com/community/tutorials/how-to-install-the-anaconda-python-distribution-on-ubuntu-20-04-pt>
- ❑ <https://www.digitalocean.com/community/tutorials/how-to-install-anaconda-on-ubuntu-18-04-quickstart-pt>

- ☐ Conda
- ☐ Funcionalidades
- ☐ Comandos

- ❑ Sistema de gerenciamento de pacotes e ambientes que roda em diversos Sistemas Operacionais
- ❑ Permite criar diversos ambientes, cada um com seus próprios softwares e versões de pacotes/dependências
- ❑ Vantagens:
  - Dispensa necessidade de acesso de administrador ao computador para realizar modificações no ambiente de bibliotecas
  - Permite criar ambientes especializados para cada aplicação

- ❑ Facilita a distribuição de pacotes python, R, julia, javascript, entre outros.;
- ❑ Mantém instalações isoladas em um único ambiente;
- ❑ Unifica diversas fontes de distribuição de pacotes em um único ponto
- ❑ Compatibilidade com o uso concomitante do **pip**
- ❑ O pacote conda e o gerenciador de ambiente estão incluídos em todas as versões do *Anaconda* e *Miniconda*;
- ❑ Criação de ambientes virtuais em um nível mais amplo, onde o próprio **python é uma dependência**;

## Feito para Data Science



Figure: Ecosistema Conda

## Funcionalidades do Conda

- ☐ Manter ambientes
- ☐ Manter pacotes em cada ambiente
- ☐ Importar e Exportar Ambientes

## Comandos Conda

- ☐ `conda`
- ☐ `conda env`



## Criação e manipulação de ambientes

- Criando um ambiente
  - `conda create -n (nome do ambiente)`
- Acessando um ambiente
  - `conda activate (nome do ambiente)`
- Listando ambientes disponíveis:
  - `conda env list`

## Comandos disponíveis dentro de um ambiente (gerência de pacotes)

- Buscar pacotes
  - `conda search (pacote)`
- Instalar pacotes
  - `conda install (pacote1, pacote2, etc)`
- Listar pacotes
  - `conda list`
- Remover pacotes
  - `conda remove (pacote)`

## Parâmetros para instalação de pacotes

- ❑ A operação `conda search` retorna pacotes, junto de suas respectivas versões e canal de distribuição;
- ❑ É possível escolher uma versão por meio dos operadores: `=` , `>=` , `<=` , `>` , `<`
- ❑ Escolhendo uma versão:
  - `conda install pacote1=2.2.3`

## Canais de distribuição

- ❑ Locais onde os pacotes são armazenados;
- ❑ Servem como base para hospedagem e gerenciamento de pacotes;
- ❑ Os pacotes Conda são baixados de canais remotos, que são *URLs* para diretórios contendo pacotes conda;
- ❑ O comando conda procura em um conjunto padrão de canais
  - Os pacotes são baixados e atualizados automaticamente de:  
<https://repo.anaconda.com/pkg>

## Canais de distribuição

- parâmetro `--channel`
  - Define a prioridade de canal de distribuição para realizar a busca
  - `--channel (canal 1) --channel (canal 2) , etc`
    - Permite definir canais de distribuição padrão para a busca, mas busca em outros canais disponíveis também)
  - `--override-channels`
    - Define que deve ser feita a busca apenas em um nos canais especificados com `--channel`

## Importar e Exportar Ambientes

- Importação e Exportação de Ambiente
  - `conda env export -n (nome ambiente) -f (arquivo exportado)`
  - `conda env create -f (arquivo a ser importado)`



Escrevendo notebooks no *Jupyter*

---

## Definições

- Documentos que contêm elementos de código e rich text, como figuras, links, equações, e etc.
- Tipo de documento é ideal para:
  - Reunir uma descrição da análise de resultados;
  - Realize a análise dos dados em tempo real;
  - Criar apresentações dinâmicas;
  - Criar documentações
- "Jupyter" é uma sigla solta que significa **J**ulia, **P**ython e **R**.
- A replicação e a reprodutibilidade são dois pilares do método científico. Os Jupyter notebooks facilitam a realização de ambos;



## Histórico do Projeto

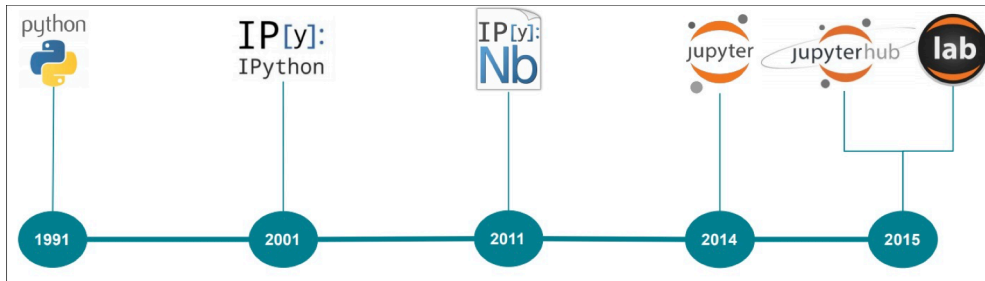


Figure: Histórico do Projeto Jupyter

## Definições

- ☐ Servidor Jupyter é um servidor standalone e single-user
- ☐ Notebook é o arquivo que contém o código e as anotações - Jupyter Notebook;
- ☐ 'Executa' o código fonte;
- ☐ Interface flexível e dinâmica;
- ☐ Armazena o código e sua saída junto com anotações em markdown;
- ☐ Notebook é um JSON com extensão `.ipynb`

## Funcionamento

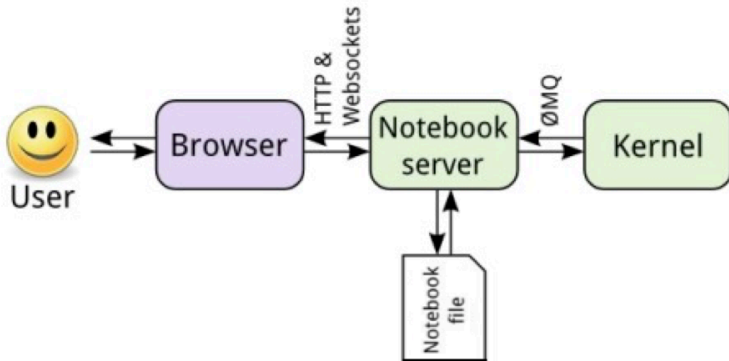


Figure: Funcionamento do servidor Jupyter

## Funcionamento

### ☐ Células:

- Blocos que podem ser executados individualmente;
- Apresentam resultados;
- **Célula de código** contém código a ser executado no kernel e exibe sua saída abaixo;
- **Célula Markdown** contém texto formatado usando *Markdown* e exibe sua saída no local quando é executada.

### ☐ Kernels:

- *Mecanismo computacional* que executa o código contido em um notebook.

## Funcionamento

- ❑ Executando comandos do sistema operacional, utilizando `!COMANDO`
  - Pode ser utilizado para instalação de pacotes utilizando `pip` ou `conda`

In [6]:

```
1 !whoami
```

```
clayton
```

---

Figure: Executando comandos Bash

## Magic Commands

- `%env`: define variável de ambiente
- `%load`: insere código de um script externo;
- `%store [-r]`: passa variáveis entre notebooks;
- `%who`: Lista todas as variáveis de escopo global;
- Medição de tempo de execução:
  - `%%time` fornecerá informações sobre uma única execução do código em sua célula.
  - `%%timeit` usa o módulo Python `timeit` que executa uma instrução 100.000 vezes (por padrão) e então fornece a média das três vezes mais rápidas.
- `%%writefile` e `%pycat`: Exporta o conteúdo de uma célula / Mostra o conteúdo de um script externo;

## Construindo Apresentações

- ❑ Pacote <https://rise.readthedocs.io/en/stable/>
  - Transforma instantaneamente o seu jupyter notebook em uma apresentação
  - Utiliza o `reveal.js`
- ❑ Tipos de slides (Slide Type):
  - **Slide**: Exibição da célula como um slide
  - **Sub-Slide**: Exibição da célula no mesmo slide, mas sobrepe o conteúdo anterior
  - **Fragment**: Exibição da célula abaixo da anterior
  - **Skip**: A célula não será exibida
  - **Notes**: Notas para o apresentador (não será exibida)

Dúvidas?