Programação de Sistemas para Internet

Prof. Diego Cirilo

Aula 03: Preparação do Ambiente

Ferramentas utilizadas na disciplina

- Python/HTML/CSS/JS
- Ambiente Virtual Python (`venv`)
- Frameworks Flask e Django
- VSCode
- Git/Github para versionamento (minicurso)

Ambiente Virtual Python (`venv`)

- Ferramenta para gerenciar pacotes Python em um ambiente isolado
- Evita conflitos com outros projetos ou sistemas
- Facilita a instalação e remoção de pacotes
- Mantém a organização e reprodutibilidade do projeto
- Documentação

Benefícios do uso do `venv`

- Isolamento: Pacotes específicos para cada projeto
- Organização: Evita conflitos e facilita a gerenciamento de pacotes
- Reprodutibilidade: Garante que o projeto possa ser executado em diferentes ambientes
- Segurança: Protege o sistema operacional de pacotes maliciosos
- Portabilidade: Facilita o compartilhamento e a execução do projeto em diferentes máquinas

Comandos básicos do `venv`

- Para criar um novo `venv`

```
python -m venv nome-do-venv
```

- O `nome-do-venv` pode ser qualquer coisa, mas na disciplinas usaremos `venv` como convenção. Portanto, o comando completo fica:

```
python -m venv venv
```

Comandos básicos do `venv`

- Para funcionar, o `venv` precisa ser ativado. No *Windows* use:

```
.\venv\Scripts\Activate.ps1
```

- Dica: use sempre a tecla "Tab" no teclado (ao lado da letra Q) para completar os comandos. Digite apenas as primeiras letras e complete com "Tab".
- Para desativar:

```
deactivate
```

Pacotes Python

- Bibliotecas que permitem acesso a novas funcionalidades
- Mais cômodo que copiar e usar o código diretamente
- Permitem o reuso
- Agilidade
- Python Package Index (PyPI)

Gerenciador de Pacotes Python

- `pip`
- Gerenciador de pacotes padrão para Python
- Instala, atualiza e remove pacotes Python de um repositório online
- Simplifica o processo de gerenciamento de dependências de projetos Python
- Disponível na maioria das distribuições Python modernas

Usando o *pip*

- Instalar um pacote: `pip install <nome_do_pacote>`
- Exemplo: `pip install requests` (instala o pacote requests)
- Atualizar um pacote: `pip install --upgrade <nome_do_pacote>`
- Remover um pacote: `pip uninstall <nome_do_pacote>`
- Listar pacotes instalados: `pip list`
- Exibir informações de um pacote: `pip show <nome_do_pacote>`

Usando o pip com venv

- Não devemos reutilizar o diretório do venv em outras máquinas
- Podemos usar o `pip` para recriar o ambiente
- Na máquina original:

```
pip freeze > requirements.txt
```

- O arquivo `requirements.txt` armazenará o nome e versões exatas dos pacotes utilizados.

Usando o pip com venv

- Para recriar o ambiente em outra máquina:
- Primeiro crie e ative o `venv` (slides anteriores)
- Depois:

```
pip install -r requirements.txt
```

- O pip instalará exatamente os pacotes que estão no arquivo `requirements.txt`

Dúvidas?