Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Estrutura de controle de fluxo

Compreensão de listas

Aula 4

Código da aula: [DADOS]ANO1C2B2S13A4

Exposição

Objetivos da Aula

Conhecer as boas práticas do Python e do código limpo.

Competências da Unidade (Técnicas e Socioemocionais)

- Ser proficiente em linguagens de programação para manipular e analisar grandes conjuntos de dados.
- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões fundamentadas em evidências.
- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados; trabalhar em equipes multifuncionais colaborando com colegas, gestores e clientes.

Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Acesso ao laboratório de informática e/ou à internet.
- Software Anaconda/Jupyter Notebook instalado ou similar.

Duração da Aula

50 minutos

Motivação: código limpo

O seu colega de trabalho saiu de férias e você precisa usar o código dele. O código a ser utilizado encontra-se a seguir.

O que você vai fazer? Ligar para o colega? Falar com seu gestor? Desistir?

```
def fatorial(n):if n==0 or n==1:return 1
else:return n*fatorial(n-1)
numeros=[]
for i in range(5):numeros.append(i+1)
print(numeros)
if numeros[0]<3:print("menor que 3")
else:print("maior ou igual a 3")
fatoriais = []
for i in numeros:fatoriais.append(fatorial(i))
print(fatoriais)
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Momento de **reflexão**

Exposição

Código limpo

Código limpo refere-se:

- a um estilo de programação que promove a **legibilidade**, a **simplicidade** e a **manutenibilidade** do código-fonte.
- àquele que é fácil de entender, modificar e manter. Ele segue boas práticas e princípios de design de software que visam criar um software eficiente, robusto e de alta qualidade.

Tome nota:

O código limpo não é apenas uma questão de estética; é uma prática que contribui para a eficiência no desenvolvimento de software e para a criação de sistemas mais robustos e sustentáveis ao longo do tempo.

Exposição

Código limpo

Alguns dos princípios e características associados ao código limpo incluem:

	Princípios e características
Nomes significativos	Escolher nomes de variáveis, funções, classes e métodos que sejam descritivos e que reflitam o propósito e a função do elemento.
Funções e métodos concisos	Manter funções e métodos curtos e com foco em uma única responsabilidade. Evitar funções muito extensas que realizam diversas tarefas.
Comentários úteis	Utilizar comentários, quando necessário, para explicar partes complexas do código ou fornecer informações adicionais. Comentários devem ser claros e relevantes.
Indentação consistente	Adotar uma indentação consistente para melhorar a legibilidade do código. Isso facilita a visualização da estrutura e do fluxo do programa.
Evitar códigos repetitivos	Evitar duplicação de código sempre que possível. Utilizar abstrações como funções, classes e módulos para promover a reutilização.

Exposição

Código limpo

Alguns dos princípios e características associados ao código limpo incluem:

	Princípios e características
Testabilidade	Escrever um código que seja fácil de testar. Códigos testáveis tendem a ser mais robustos e menos propensos a erros.
Organização lógica	Organizar o código de maneira lógica, agrupando funcionalidades relacionadas e mantendo uma estrutura coerente.
Refatoração	Estar disposto a refatorar o código, conforme necessário, para melhorar sua estrutura e clareza, sem alterar seu comportamento.
Documentação adequada	Incluir documentação, como docstrings, em funções e módulos, para fornecer informações sobre como usar o código.
Atenção aos detalhes de estilo	Seguir as convenções de estilo da linguagem de programação utilizada. Isso inclui a formatação adequada do código, o uso consistente de espaços em branco e outros detalhes estilísticos.

Exposição

PEP 8

O PEP 8 (Python Enhancement Proposal 8) é um guia de estilo para a escrita de código em Python. Ele fornece:

- ✓ recomendações sobre a formatação do código.
- ✓ a escolha de nomes de variáveis.
- ✓ a estruturação de imports.
- ✓ a organização de código.
- ✓ outras convenções que visam melhorar a legibilidade e a consistência do código Python.

Dica:

O PEP 8 é amplamente aceito na comunidade Python e é seguido para manter a consistência no ecossistema Python. Ferramentas podem ajudar a automatizar a verificação do código em relação às diretrizes do PEP 8. Adotar essas convenções ajuda a melhorar a colaboração e a legibilidade do código em projetos Python.

Exposição

PEP 8

Algumas das principais diretrizes do PEP 8 incluem:

1

Indentação

Use espaços em branco para a indentação em vez de *tabs*. A convenção recomendada é usar quatro espaços para cada nível de indentação.

2

Comprimento das linhas

Limite o comprimento das linhas a 79 caracteres para código, e 72 para docstrings. Se uma expressão não couber em uma única linha, use parênteses para dividir em várias linhas.

3

Importações

Agrupe as importações em três seções, sendo elas: bibliotecas padrão, bibliotecas relacionadas a terceiros e suas próprias bibliotecas.

4

Espaços em branco em expressões e instruções

Use espaços em branco de forma consistente para melhorar a legibilidade. Evite espaços em branco em excesso.

Exposição

PEP 8

Algumas das principais diretrizes do PEP 8 incluem:

5

Nomeação de variáveis e funções

Use nomes descritivos e evite nomes únicos que possam ser confundidos com palavras-chave do Python.

6

Comentários

Use comentários para explicar partes complexas do código, mas evite comentários óbvios e redundantes.

7

Docstrings

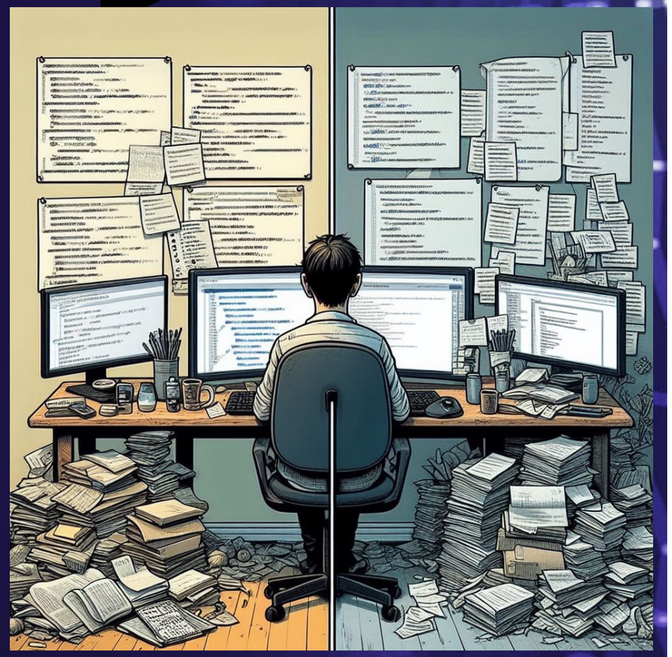
Inclua docstrings para documentar classes, funções e módulos. Siga o formato do Docstring Convention PEP 257.

8

Operadores

Adicione espaços em torno de operadores para melhorar a clareza.

Vamos fazer uma **atividade**



Elaborado especialmente para o curso com a ferramenta Microsoft Copilot.

Secretaria da Educação SÃO PAULO GOVERNO DO ESTADO

Código limpo

O seu colega de trabalho saiu de férias e você precisa usar o código dele. O código do seu colega encontra-se a seguir:

```
def fatorial(n):if n==0 or n==1:return 1
else:return n*fatorial(n-1)
numeros=[]
for i in range(5):numeros.append(i+1)
print(numeros)
if numeros[0]<3:print("menor que 3")
else:print("maior ou igual a 3")
fatoriais = []
for i in numeros:fatoriais.append(fatorial(i))
print(fatoriais)
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

- **Reescreva** o código, deixando-o limpo e legível. Use compreensão de lista, se couber.

© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1. O conhecimento sobre as boas práticas em Python;

2. A aplicação do guia de estilo para programar em Python (PEP8);

3. A compreensão do conceito de código limpo.

Saiba mais

Onde aplicar a compreensão de listas?

FELIPE, A. *Compreensão de listas no Python*. Alura, 25 jul. 2016. Disponível em: https://www.alura.com.br/artigos/simplicando-o-processamento-com-compreensao-de-lista-do-python. Acesso em: 24 mar. 2024.

Um artigo completo sobre listas:

ORESTES, Y. *Listas em Python*: operações básicas. Alura, 18 set. 2023. Disponível em: https://www.alura.com.br/artigos/listas-no-python#filtrando-uma-lista-utilizando-_list-comprehensions_. Acesso em: 24 mar. 2024.

Boas práticas em Python:

VAN ROSSUM, G.; WARSAW, B.; COGHLAN, A. *PEP 8 – Style Guide for Python Code*. Python Enhancement Proposals, 5 jul. 2001. Disponível em: https://peps.python.org/pep-0008/. Acesso em: 24 mar. 2024.

Referências da aula

BRANCO, H, J. *PEP 8 e imports em Python*. Medium, 9 jul. 2021. Disponível em: https://medium.com/gbtech/pep-8-e-imports-em-python-78a6fbf53475. Acesso em: 24 mar. 2024.

MARTIN, R. C. *Código limpo*: habilidades práticas do Agile Software. Rio de Janeiro: Alta Books, 2009.

VAN ROSSUM, G.; WARSAW, B.; COGHLAN, A. *PEP 8 – Style Guide for Python Code*. Python Enhancement Proposals, 5 jul. 2001. Disponível em: https://peps.python.org/pep-0008/. Acesso em: 24 mar. 2024.

VAN ROSSUM, G.; WARSAW, B.; COGHLAN, A. *PEP 8 - Guia de estilo para Python*. Traduzido por Pedro Werneck, 16 nov. 2003. Disponível em: https://wiki.python.org.br/GuiaDeEstilo. Acesso em: 24 mar. 2024.

Identidade visual: imagens © Getty Images.

Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO