FORMAÇÃO COMPLEMENTAR

Mentor: Felipe Moreira de Assunção
Conteúdo para formação complementar do Módulo de arquivos, erros e exceções

INFORMAÇÕES DO MÓDULO

Descrição

Conheça os conceitos básicos sobre Arquivos, erros e exceções em Python

Objetivos do Ensino

Espera-se que o aluno consiga, ao final do módulo, trabalhar com:

- Arquivos (abrir, ler, escrever, armazenar, fechar, imprimir)
- Erros, exceções, tratamento de exceções

Demonstração das Ferramentas

Utilização do Anaconda com Jupyter Notebook e Python ou editor de código de sua escolha

ARQUIVOS ERROS E EXCEÇÕES

1. Arquivos

Ao trabalhar com arquivos, tente também criar documentos com outras extensões como .docx .xlsx dentre outros. Assim você poderá ampliar a utilização de arquivos em Python e verificar algumas possibilidades ou limitações na criação de determinados arquivos.

Sempre que criar um arquivo, não esqueça de fecha-lo. Alguns erros podem ocorrer se você criar o arquivo e tentar acessá-lo em seu diretório sem ter concluído, em código, o processo de fechamento. Fique atento.

2. Dicas para evitar erros de sintaxe

Aqui estão algumas maneiras de se evitar os erros de sintaxe mais comuns.

1. Certifique-se que você não está usando uma palavra reservada do Python (*Python keyword*) como nome de variável.
2. Verifique que você colocou um dois-pontos (`:`) no final do cabeçalho de cada comando composto (*compound statement*), incluindo laço `for`, laço `while`, comandos de seleção `if`, `elif` e `else` e declaração de funções `def`.
3. Verifique se a tabulação é consistente. Você pode tabular com espaços ou tabs mas não é aconselhável misturá-los, Cada nível deve ter a mesma quantidade de espaços ou tabs.
4. Certifique de que as aspas ou apóstrofos de qualquer string no código estão emparelhados.
5. Se você tem strings que se estendem por várias linhas delimitados por citação tripla (com `'` ou `"`), certifique-se que você terminou o strings de maneira apropriada. Um string não terminado pode causar o erro `invalid token` no final do seu programa ou pode tratar o trecho seguinte de código como um string até encontrar o próximo string. No segundo caso pode até não produzir uma mensagem de erro!
6. Uma falta de fecha parêntese, chave ou colchete – `(`, `{` ou `[` – faz o Python continuar com a próxima linha como parte da expressão corrente. Geralmente, um erro ocorre quase que imediatamente na próxima linha.

7. Verifique pelo clássico `=` em vez de `==` em uma condição.

Fonte: https://panda.ime.usp.br/pensepy/static/pensepy/Appendices/app_a.html

3. Um pouco mais sobre exceções

Em nossas aulas trabalhamos com algumas exceções e verificamos como podemos trata-las através dos blocos de instrução try, except, else e finally. O ponto chave para o tratamento de exceções específicas é dar visibilidade ao erro e informar ao usuário de maneira mais amigável e específica, o erro que está ocorrendo. Saber exatamente o que aconteceu fará com que tempo e esforço empregados para conter o incidente sejam significativamente menores.

3.1 Hierarquia de exceções

Vamos a um exemplo sobre hierarquia de exceções. Considere a seguinte hierarquia:

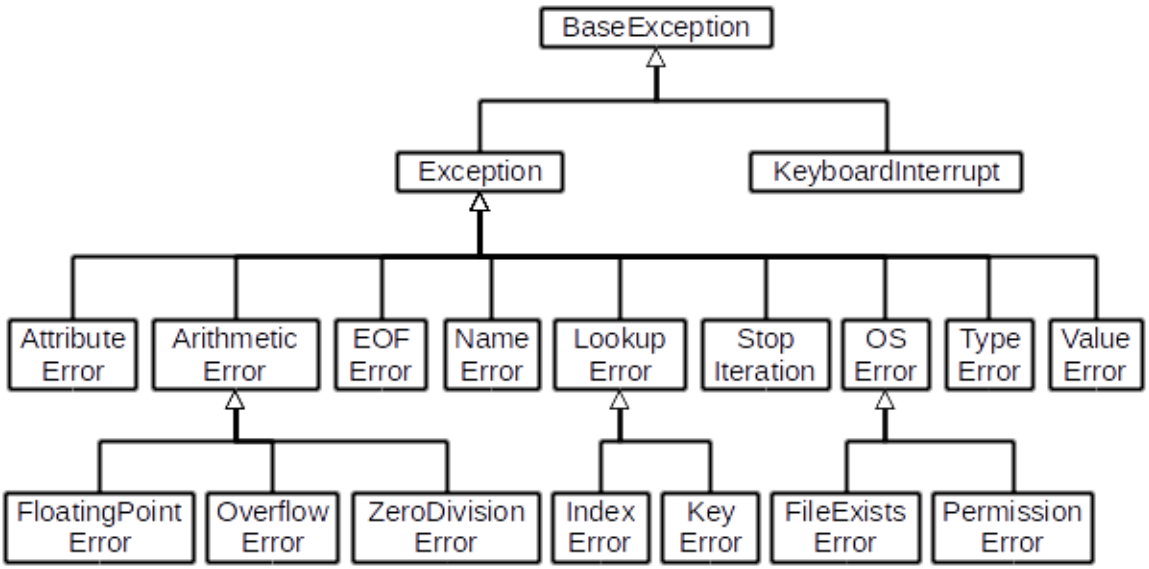

Figura 1 – Exemplos de exceções e suas hierarquias

Como você saberá qual é o erro específico quando ocorre um *"Arithmetic Error"*? Imagine que ao tratar esse erro, você colocaria uma exceção para *"Arithmetic Error"* apenas...O ideal é ser mais específico, tornando mais claro o erro ocorrido de modo que o usuário possa agir com mais precisão na resolução do problema. Neste caso, se o erro específico é uma divisão por zero, podemos dizer que como regra, é importante tratar a exceção da mais especifica para a mais generalizada. Sendo assim, reafirmamos mais alguns detalhes:

- É importante que você saiba sobre a hierarquia de exceções da linguagem de programação;
- A ordem importa: utilize a exceção mais específica para a mais generalizada, **SEMPRE**;
- Evite tratamentos excessivamente genéricos, você pode acabar suprimindo exceções que não deveria.

LINKS INTERESSANTES

- Quando falamos sobre o tratamento de erros em nossos códigos, vários sentimentos podem vir à tona. Existe uma teoria geral da comunicação chamada “Equação da Mídia” que afirma que as pessoas tendem a tratar os computadores e outras mídias como se fossem pessoas reais ou lugares reais. Saiba mais em sobre essa equação em: https://en.wikipedia.org/wiki/The_Media_Equation

BIBLIOGRAFIA

- Python Books: https://wiki.python.org/moin/PythonBooks
- Introdução à Programação com Python: Algoritmos e Lógica de Programação Para Iniciantes
- Python Fluente: Programação Clara, Concisa e Eficaz