

Erros comuns em Python

Erros e a Teoria dos Erros

O conceito de erro é algo bem abstrato. Podemos ter o mesmo como uma situação, acontecimento ou entidade que é surgida toda vez que algo inesperado ou factivamente impossível ocorre.

Isto pode ser bem mais do que abstrato, dado que cada um possui seu próprio conceito de “erro” ou, indo ainda mais a fundo, conceito de “errado”.

Em sentido mais exato, trataremos aqui de erros como ocorrências de “situações inesperadas e/ou ilógicas”.

Desde a antiguidade, estudiosos das áreas da matemática buscam desenvolver formas e métricas para quantificação. Com o passar do tempo, destes estudos surgiu uma vertente conhecida como a **Teoria dos Erros**.

Muito aplicada nas disciplinas da Física, esta teoria diz respeito a questões como ruídos em conjuntos, imprecisão e incertezas.

Muitas das técnicas para tratamento de erros em diversas áreas das ciências e engenharias tomam por base os conceitos nela desenvolvidos.

Erros em Software

Tratamento de erros e debugging deve ser uma parte rotineira da vida de todo e qualquer profissional que se envolva com programação.

Muitas das dificuldades em se desenvolver um bom software não residem no ato de se desenvolver, mas sim, no ato de realizar correções naquilo que foi desenvolvido.

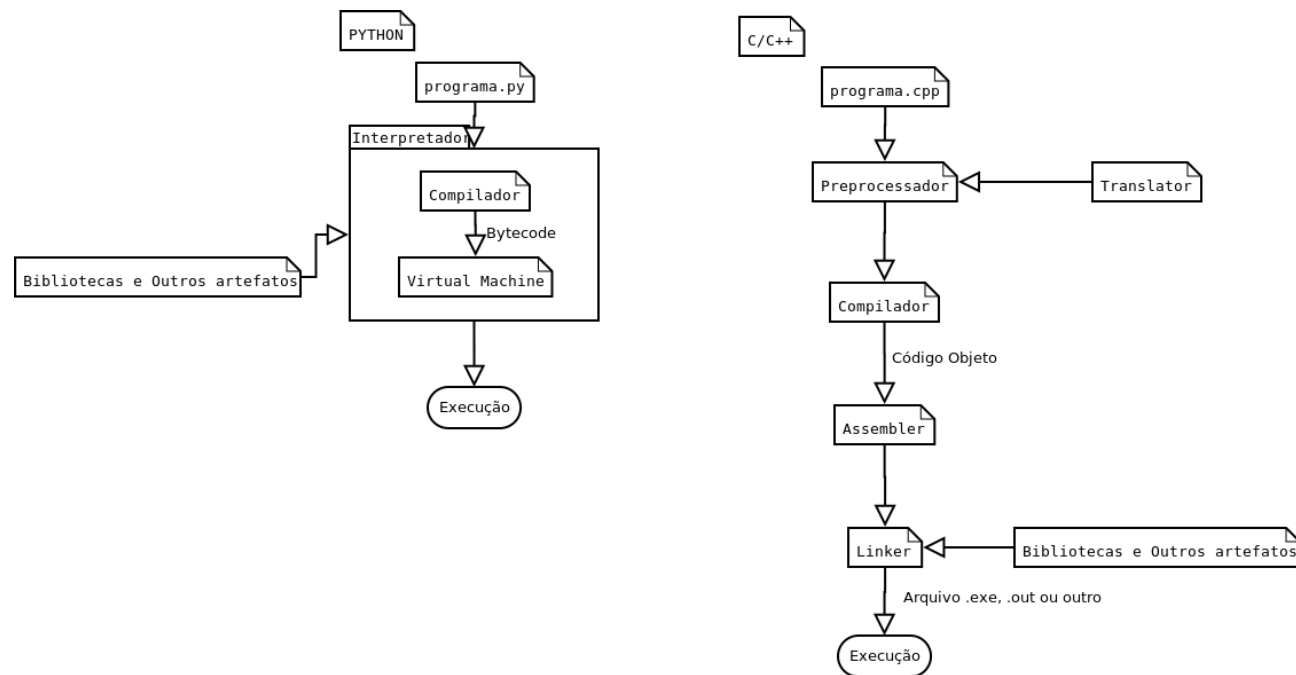
Além disso, com a dependência incessante por software, no mundo de hoje, o tratamento de erros é com certeza mais útil que nunca.

O que se pode fazer é, simplesmente, tomar prática nisto e conhecer as ferramentas as quais são dispostas a nós, para tratamento de erros.

De uma forma geral, um programa pode apresentar erros em dois momentos distintos:

- Erros em tempo de Compilação: Mais “tratáveis computacionalmente”, são erros surgidos durante a compilação de um programa, que geralmente, impedem o programa de executar. Ex.: Parênteses únicos.
- Erros em Runtime (Tempo de execução): Mais abstratos, são erros surgido durante a execução de um programa. Estes não causam impedimento na execução do programa, porém geralmente causam o “crash” do programa ao serem atingidos. Ex.: Divisão por 0

Vale ressaltar que, apesar do python, em teoria, empregar o conceito de linguagem interpretada, o tratamento a erros em tempo de compilação é perfeitamente plausível e possível.



Especificações dos erros mais comuns

Dentre os erros mais comuns durante o desenvolvimento em python, destacamos:

- Nomenclatura;
- Identação;
- Fluxo;
- Entrada;
- Cálculo;
- Hardware;

Interpretando debuggers

Debuggers são ferramentas indispensáveis para todo e qualquer programador. Tanto, que raras são as linguagens de programação que não apresentam alguma ferramenta para tal funcionalidade.

Em python, temos a ferramenta Traceback, que, apesar de não ser “oficialmente” um debugger, nos auxilia neste processo.

A interpretação do resultado gerado pelo Traceback se dá sempre de baixo para cima. Deve se atentar para linhas de código e para as mensagens resultantes da mesma.

Uma outra dica aqui é que habilidades de leitura em língua inglesa contam como uma grande vantagem para se interpretar o Traceback.

```
1  
2 print ("Eu acho que nao vai dar nao...")  
3
```

```
antonio@natsuki ~/Documentos/Ificina/Educacional/Minicursos/MinicursoPythonBásico/Slides/Exemplos/Traceback $ python errado.py  
File "errado.py", line 3  
                                ^  
SyntaxError: invalid syntax
```

