Ctrl+Play
escola de programação
e robótica

PYTHON

CONCEITOS BÁSICOS E SUAS APLICAÇÕES

Aula 4:
Tratamentos de Exceções e modulos

Tratamento de Exceções:

Tratamento de Exceções:

O tratamento de exceções em Python é uma técnica fundamental para lidar com erros e situações inesperadas que podem ocorrer durante a execução de um programa. O objetivo do tratamento de exceções é permitir que o programa continue funcionando de maneira controlada, mesmo quando ocorrem erros.

Modulos:

Módulos em Python são arquivos que contêm definições e instruções Python, como funções, classes e variáveis, que podem ser importadas e reutilizadas em outros programas Python. Os módulos ajudam a organizar e estruturar o código, permitindo a reutilização e a manutenção mais fácil. Aqui está um resumo detalhado sobre módulos em Python

The name "Python" for the programming language was inspired by the British comedy group "Monty Python"

Ctrl+Play
escola de programação
e robótica

PYTHON

CONCEITOS BÁSICOS E SUAS APLICAÇÕES

Tratamentos de Exceções

Tratamento de Exceções:

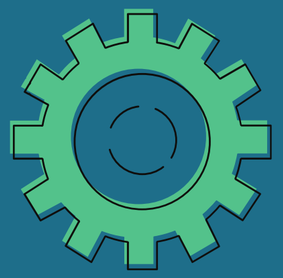
Try:

Bloco try
O bloco try contém o código que você deseja monitorar para possíveis exceções. Se uma exceção ocorrer dentro deste bloco, a execução é imediatamente interrompida, e o controle é passado para o bloco except correspondente

try:

x = 1 / 0

Código que pode gerar uma exceção



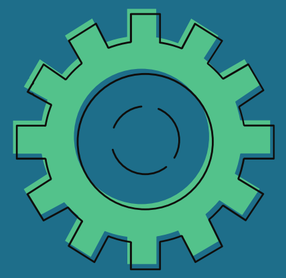
Tratamento de Exceções:

Except:

Bloco except
O bloco except contém o código que será executado se uma exceção ocorrer no bloco try. Você pode capturar exceções específicas ou usar um bloco except genérico para capturar qualquer exceção

```
try:
    exceção x = 1 / 0

except ZeroDivisionError:
    print("Divisão por zero não é permitida!")
```



Código que pode gerar uma exceção

Tratamento de Exceções:

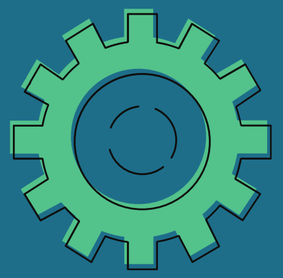
Finally

Bloco finally
O bloco finally é opcional e será sempre executado, independentemente de uma exceção ter ocorrido ou não. Ele é útil para liberar recursos ou executar ações de limpeza.

```
try:
    exceção x = 1 / 0

except ZeroDivisionError:
    print("Divisão por zero não é permitida!")

Finally:
    print("Esta mensagem é sempre exibida, independentemente de erros.")
```



Liberar recursos

Tratamento de Exceções:

Except:

Os comandos try e except são utilizados para tratar exceções em Python, permitindo que o programa lide com erros de forma controlada. A principal finalidade de utilizar try e except é garantir que o programa possa continuar executando mesmo quando ocorrem erros, proporcionando uma experiência de usuário mais robusta e prevenindo que o programa termine abruptamente devido a erros não tratados. Aqui estão algumas razões específicas para utilizar try e except:

Tratamento de Exceções:

Tipos de exceções comuns

ArithmeticError

A classe base para todos os erros numéricos.
ZeroDivisionError: Levantada quando uma divisão ou módulo por zero é tentado.
OverflowError: Levantada quando um cálculo numérico resulta em um overflow

AttributeError

Levantada quando um atributo de objeto não é encontrado.

EOFError

Levantada quando a função input() chega ao final do arquivo (EOF) sem ler qualquer dado.

NameError

Levantada quando uma variável local ou global não é encontrada.
UnboundLocalError: Subclasse de NameError, levantada quando uma referência local é feita a uma variável antes de ser atribuída.

KeyError

Levantada quando uma chave de mapeamento (dicionário) não é encontrada.

OSError

Levantada quando uma função de sistema causa um erro relacionado ao sistema.
FileNotFoundError: Levantada quando um arquivo ou diretório não é encontrado.
PermissionError: Levantada quando uma operação não tem a permissão necessária.

Tratamento de Exceções:

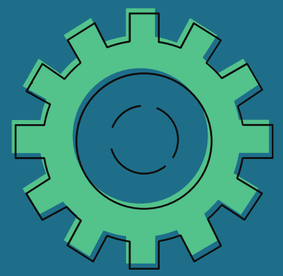
Tipos de exceções comuns

RuntimeError

Levantada quando um erro não categorizado de outra forma ocorre.
NotImplementedError: Levantada quando um método abstrato que requer uma implementação em uma subclasse é chamado.

Warning

A classe base para todas as advertências.
DeprecationWarning: Advertência sobre o uso de uma função ou recurso obsoleto.
RuntimeWarning: Advertência sobre problemas que podem ocorrer durante a execução.



SyntaxError

Levantada quando ocorre um erro de sintaxe.

IndentationError: Subclasse de SyntaxError, levantada quando a indentação não está correta.
TabError: Subclasse de IndentationError, levantada quando há uma mistura de tabulações e espaços na indentação.

TypeError

Levantada quando uma operação ou função é aplicada a um objeto de tipo inadequado.

ValueError

Levantada quando uma função recebe um argumento com o tipo certo, mas valor inapropriado.
UnicodeError: Subclasse de ValueError, levantada quando ocorre um erro relacionado ao Unicode.

Ctrl+Play
escola de programação
e robótica

PYTHON

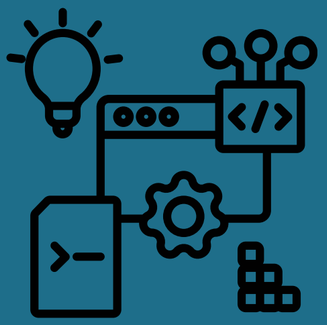
CONCEITOS BÁSICOS E SUAS APLICAÇÕES

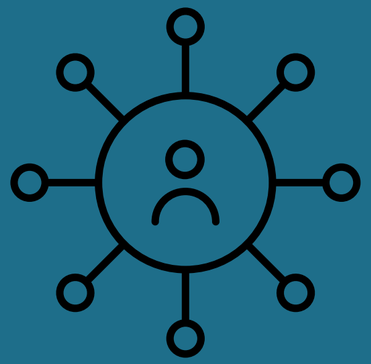
Modulos

Modulos

Biblioteca:

Uma biblioteca em Python é um conjunto de módulos e pacotes que oferecem funcionalidades específicas para facilitar o desenvolvimento de software. Elas são compostas por módulos que podem ser importados em seus programas para fornecer funcionalidades adicionais que não estão disponíveis na biblioteca padrão do Python.





Ampla Variedade:

Python possui uma vasta gama de bibliotecas disponíveis para uma variedade de fins, incluindo processamento de dados, aprendizado de máquina, visualização, desenvolvimento web, automação, entre outros. Alguns exemplos populares incluem NumPy, pandas, TensorFlow, Matplotlib, Flask e requests.

Modulos

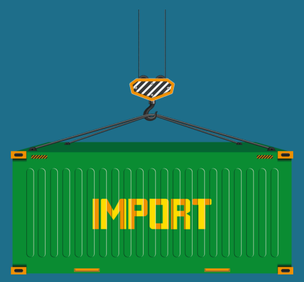
Biblioteca:

Instalação: Nem todas as bibliotecas vêm pré-instaladas com o Python. Você pode instalar bibliotecas adicionais usando o pip, o gerenciador de pacotes padrão do Python. Por exemplo, para instalar a biblioteca requests, você pode usar o seguinte comando:

pip install requests

Importação:
Assim como os módulos, você pode importar bibliotecas em seus programas Python usando a declaração import. Por exemplo:

import requests



Documentação:

Cada biblioteca geralmente vem com documentação detalhada que descreve suas funcionalidades, exemplos de uso e instruções de instalação. É sempre uma boa prática consultar a documentação oficial de uma biblioteca ao usá-la pela primeira vez.