

# Python - exceções integradas

Aqui está uma lista de exceções padrão disponíveis em Python -

Sr. Não.	Nome e descrição da exceção
1	<b>Exceção</b> Classe base para todas as exceções
2	<b>PararIteração</b> Gerado quando o método next() de um iterador não aponta para nenhum objeto.
3	<b>Saída do sistema</b> Gerado pela função sys.exit().
4	<b>Erro padrão</b> Classe base para todas as exceções integradas, exceto StopIteration e SystemExit.
5	<b>Erro Aritmético</b> Classe base para todos os erros que ocorrem no cálculo numérico.
6	<b>Erro de estouro</b> Gerado quando um cálculo excede o limite máximo para um tipo numérico.
7	<b>Erro de ponto flutuante</b> Gerado quando um cálculo de ponto flutuante falha.
8	<b>Erro ZeroDivison</b> Gerado quando ocorre divisão ou módulo por zero para todos os tipos numéricos.
9	<b>Erro de afirmação</b> Gerado em caso de falha da instrução Assert.
10	<b>AtributoError</b> Gerado em caso de falha na referência ou atribuição de atributos.
11	<b>EOFError</b> Gerado quando não há entrada da função raw_input() ou input() e o final do arquivo é atingido.
12	<b>Erro de importação</b> Gerado quando uma instrução de importação falha.
13	<b>TecladoInterrupção</b>



Gerado quando o usuário interrompe a execução do programa, geralmente pressionando Ctrl+C.

14 **Erro de pesquisa**  
Classe base para todos os erros de pesquisa.

15 **Erro de índice**  
Gerado quando um índice não é encontrado em uma sequência.

16 **KeyError**  
Gerado quando a chave especificada não é encontrada no dicionário.

17 **NomeErro**  
Gerado quando um identificador não é encontrado no namespace local ou global.

18 **UnboundLocalError**  
Gerado ao tentar acessar uma variável local em uma função ou método, mas nenhum valor foi atribuído a ela.

19 **Erro Ambiental**  
Classe base para todas as exceções que ocorrem fora do ambiente Python.

20 **Erro IO**  
Gerado quando uma operação de entrada/saída falha, como a instrução print ou a função open() ao tentar abrir um arquivo que não existe.

21 **Erro de sistema operacional**  
Gerado para erros relacionados ao sistema operacional.

22 **Erro de sintaxe**  
Gerado quando há um erro na sintaxe do Python.

23 **Erro de recuo**  
Gerado quando o recuo não é especificado corretamente.

24 **Erro no sistema**  
Gerado quando o interpretador encontra um problema interno, mas quando esse erro é encontrado, o interpretador Python não sai.

25 **Saída do sistema**  
Gerado quando o interpretador Python é encerrado usando a função sys.exit(). Se não for tratado no código, faz com que o interpretador saia.

26 **Erro de tipo**  
Gerado quando é tentada uma operação ou função inválida para o tipo de dados especificado.

27 **Erro de valor**

Gerado quando a função interna para um tipo de dados tem o tipo válido de argumentos, mas os argumentos têm valores inválidos especificados.

28

### **Erro de tempo de execução**

Gerado quando um erro gerado não se enquadra em nenhuma categoria.

29

### **Erro NotImplemented**

Gerado quando um método abstrato que precisa ser implementado em uma classe herdada não é realmente implementado.

Aqui estão alguns exemplos de exceções padrão -

## Erro de índice

É mostrado ao tentar acessar o item em um índice inválido.

```
numbers=[10,20,30,40]
for n in range(5):
    print (numbers[n])
```

Ele produzirá a seguinte **saída** -

```
10
20
30
40
Traceback (most recent call last):

  print (numbers[n])
IndexError: list index out of range
```

## MóduloNotFoundError

Isso é exibido quando o módulo não foi encontrado.

```
import notamodule
Traceback (most recent call last):

  import notamodule
ModuleNotFoundError: No module named 'notamodule'
```

## KeyError

Ocorre porque a chave do dicionário não foi encontrada.

```
D1={'1':"aa", '2':"bb", '3':"cc"}
print ( D1['4'])
Traceback (most recent call last):

  D1['4']
KeyError: '4'
```

## Erro de importação

É mostrado quando a função especificada não está disponível para importação.

```
from math import cube
Traceback (most recent call last):

  from math import cube
ImportError: cannot import name 'cube'
```

## PararIteração

Este erro aparece quando a função next() é chamada após o esgotamento do fluxo do iterador.

```
.it=iter([1,2,3])
next(it)
next(it)
next(it)
next(it)
Traceback (most recent call last):

  next(it)
StopIteration
```

## Erro de tipo

Isto é mostrado quando o operador ou função é aplicado a um objeto de tipo inadequado.

```
print ('2'+2)
Traceback (most recent call last):

  '2'+2
TypeError: must be str, not int
```

## Erro de valor

É exibido quando o argumento da função é de tipo inadequado.

```
print (int('xyz'))
Traceback (most recent call last):

  int('xyz')
ValueError: invalid literal for int() with base 10: 'xyz'
```

## NomeErro

Isso é encontrado quando o objeto não foi encontrado.

```
print (age)
Traceback (most recent call last):

  age
NameError: name 'age' is not defined
```

## Erro ZeroDivision

É mostrado quando o segundo operador na divisão é zero.

```
x=100/0
Traceback (most recent call last):

  x=100/0
ZeroDivisionError: division by zero
```

## TecladoInterrupção

Quando o usuário pressiona a tecla de interrupção normalmente Control-C durante a execução do programa.

```
name=input('enter your name')
enter your name^c
Traceback (most recent call last):

  name=input('enter your name')
KeyboardInterrupt
```