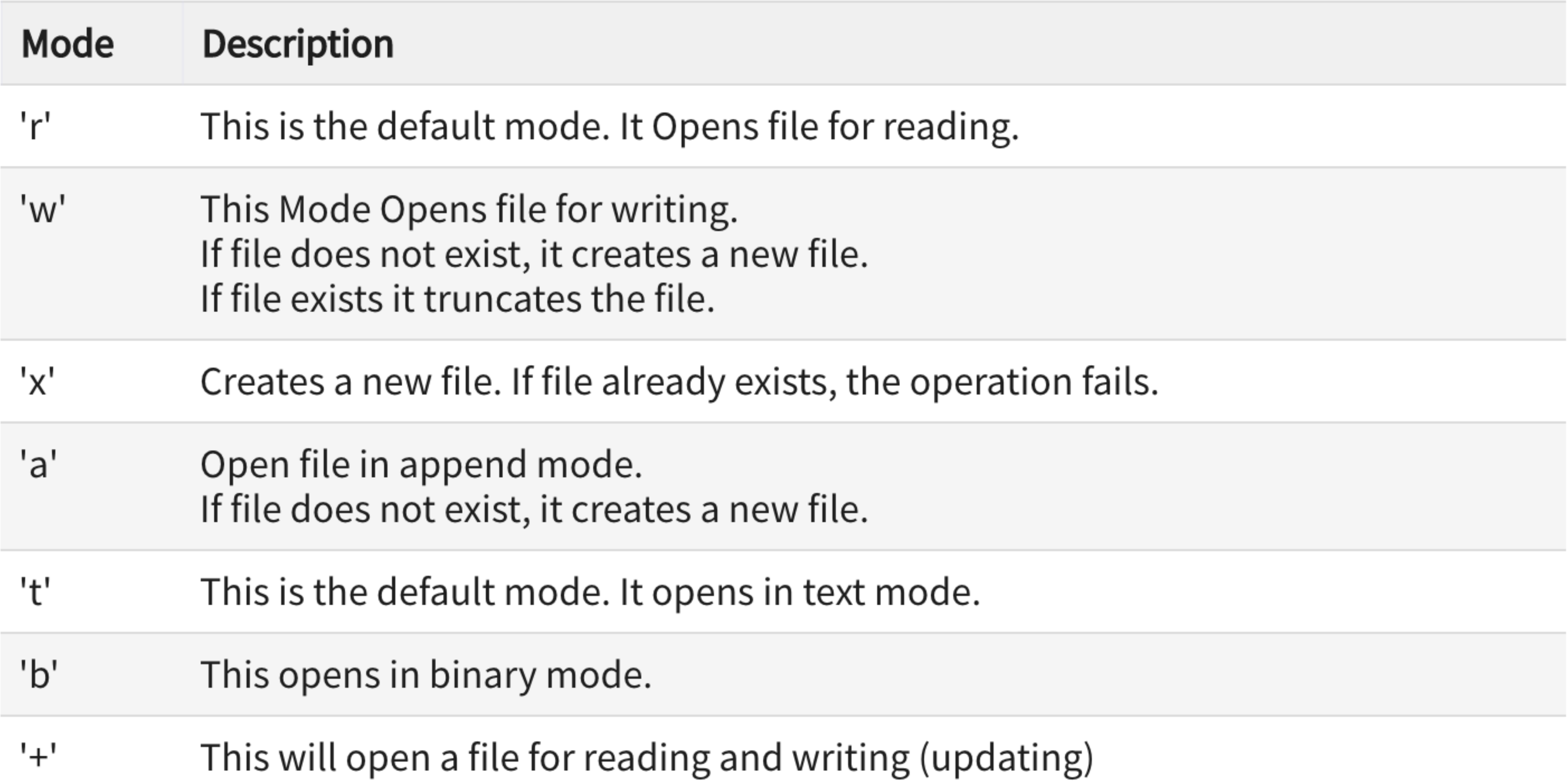
Files dan Error Handlings di Python

Muhammad Iqbal

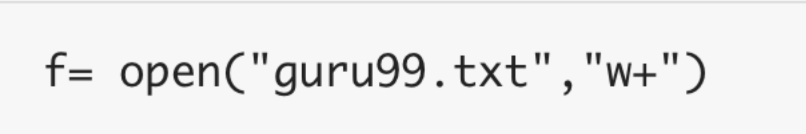
Laboratorium Instrumentasi dan Robotika Kelautan
Departemen Ilmu dan Teknologi Kelautan
FPIK – IPB University

File Modes in Python

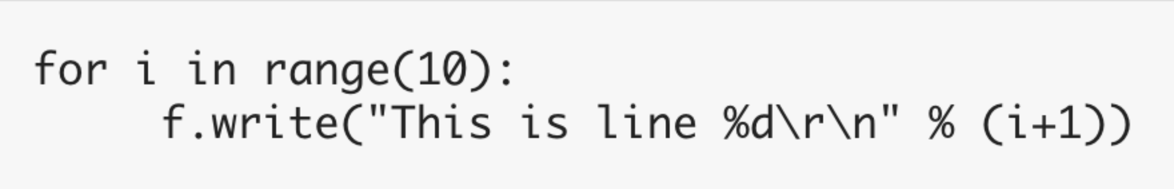


Mode	Description
'r'	This is the default mode. It Opens file for reading.
'w'	This Mode Opens file for writing. If file does not exist, it creates a new file. If file exists it truncates the file.
'x'	Creates a new file. If file already exists, the operation fails.
'a'	Open file in append mode. If file does not exist, it creates a new file.
't'	This is the default mode. It opens in text mode.
'b'	This opens in binary mode.
'+'	This will open a file for reading and writing (updating)

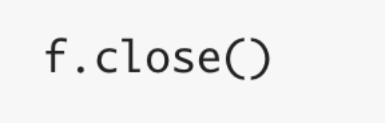
Open Text File



```
f= open("guru99.txt","w+")
```

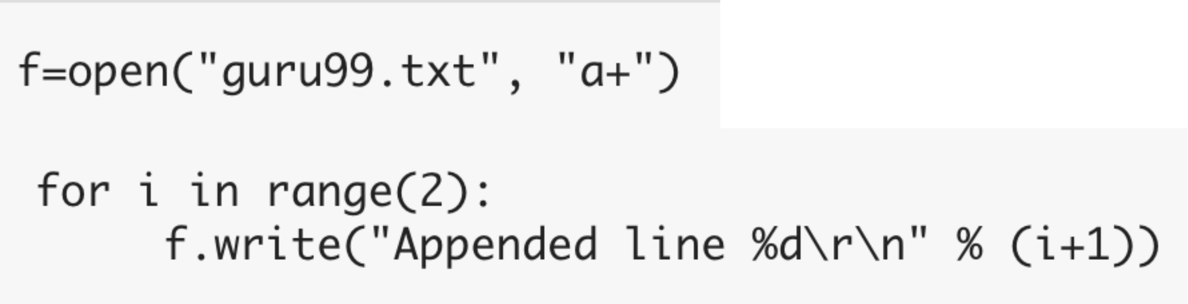


```
for i in range(10):
     f.write("This is line %d\r\n" % (i+1))
```



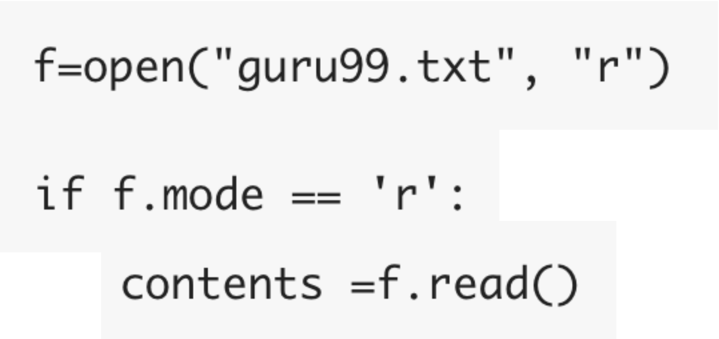
```
f.close()
```

Append Data

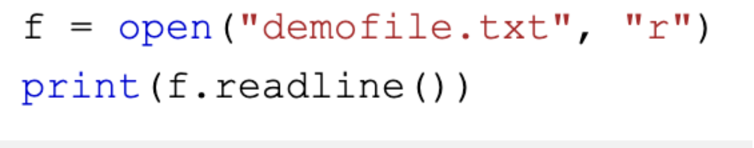


```
f=open("guru99.txt", "a+")

 for i in range(2):
      f.write("Appended line %d\r\n" % (i+1))
```

Read File



```
f=open("guru99.txt", "r")

if f.mode == 'r':
    contents =f.read()
```



```
f = open("demofile.txt", "r")
print(f.readline())
```

Files Handling menggunakan Pandas

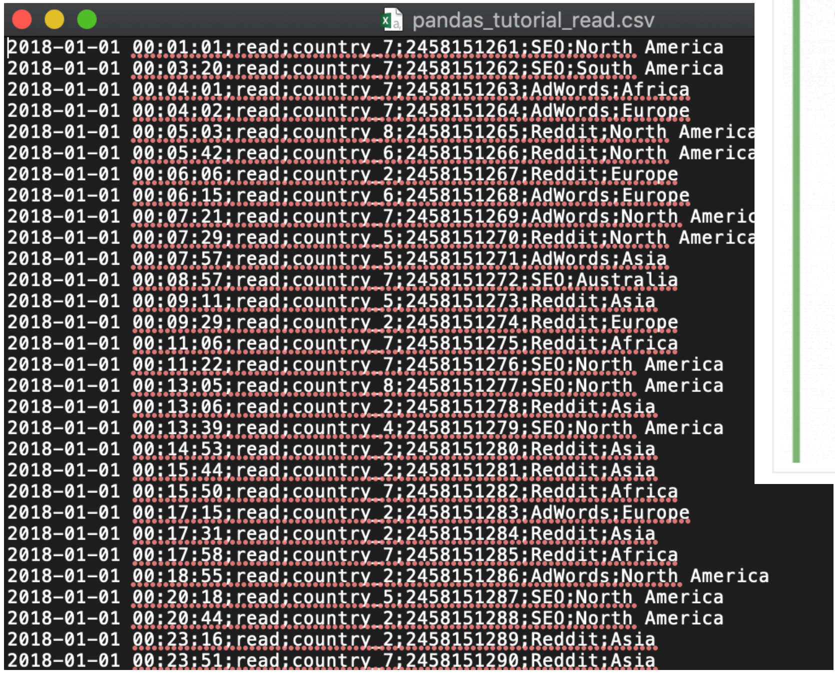
Pandas adalah Library yang **paling banyak** digunakan oleh pengguna Python untuk Pembacaan File dan pengolahan data.

Umumnya digunakan untuk Data sains

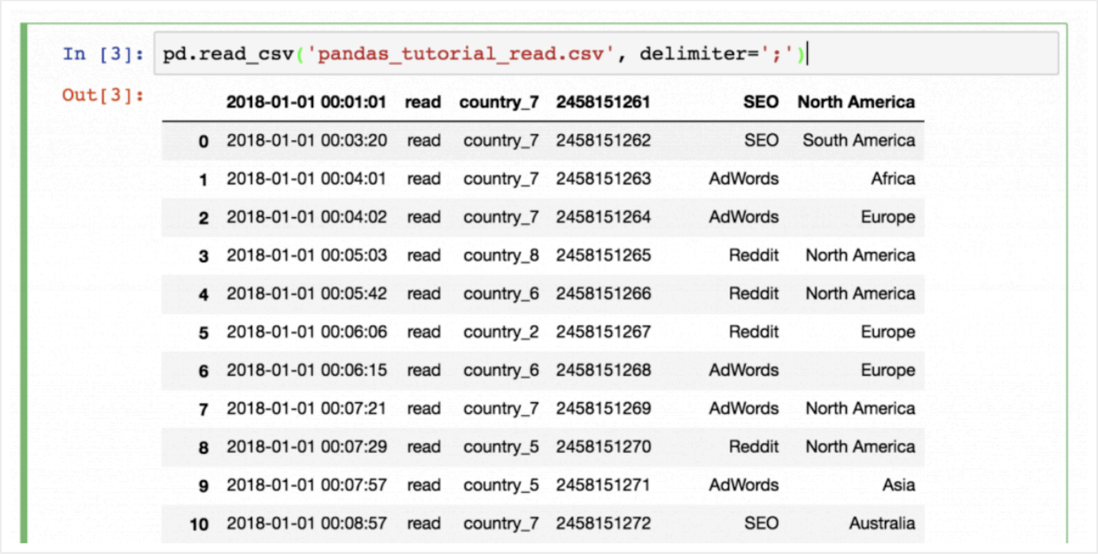
Contoh

```
pd.read_csv('pandas_tutorial_read.csv', delimiter=';')
```

The data is loaded into pandas!



```
2018-01-01 00:01:01;read;country_7;2458151261;SEO;North America
2018-01-01 00:03:20;read;country_7;2458151262;SEO;South America
2018-01-01 00:04:01;read;country_7;2458151263;AdWords;Africa
2018-01-01 00:04:02;read;country_7;2458151264;AdWords;Europe
2018-01-01 00:05:03;read;country_8;2458151265;Reddit;North America
2018-01-01 00:05:42;read;country_6;2458151266;Reddit;North America
2018-01-01 00:06:06;read;country_2;2458151267;Reddit;Europe
2018-01-01 00:06:15;read;country_6;2458151268;AdWords;Europe
2018-01-01 00:07:21;read;country_7;2458151269;AdWords;North Americ
2018-01-01 00:07:29;read;country_5;2458151270;Reddit;North America
2018-01-01 00:07:57;read;country_5;2458151271;AdWords;Asia
2018-01-01 00:08:57;read;country_7;2458151272;SEO;Australia
2018-01-01 00:09:11;read;country_5;2458151273;Reddit;Asia
2018-01-01 00:09:29;read;country_2;2458151274;Reddit;Europe
2018-01-01 00:11:06;read;country_7;2458151275;Reddit;Africa
2018-01-01 00:11:22;read;country_7;2458151276;SEO;North America
2018-01-01 00:13:05;read;country_8;2458151277;SEO;North America
2018-01-01 00:13:06;read;country_2;2458151278;Reddit;Asia
2018-01-01 00:13:39;read;country_4;2458151279;SEO;North America
2018-01-01 00:14:53;read;country_2;2458151280;Reddit;Asia
2018-01-01 00:15:44;read;country_2;2458151281;Reddit;Asia
2018-01-01 00:15:50;read;country_7;2458151282;Reddit;Africa
2018-01-01 00:17:15;read;country_2;2458151283;AdWords;Europe
2018-01-01 00:17:31;read;country_2;2458151284;Reddit;Asia
2018-01-01 00:17:58;read;country_7;2458151285;Reddit;Africa
2018-01-01 00:18:55;read;country_2;2458151286;AdWords;North America
2018-01-01 00:20:18;read;country_5;2458151287;SEO;North America
2018-01-01 00:20:44;read;country_2;2458151288;SEO;North America
2018-01-01 00:23:16;read;country_2;2458151289;Reddit;Asia
2018-01-01 00:23:51;read;country_7;2458151290;Reddit;Asia
```



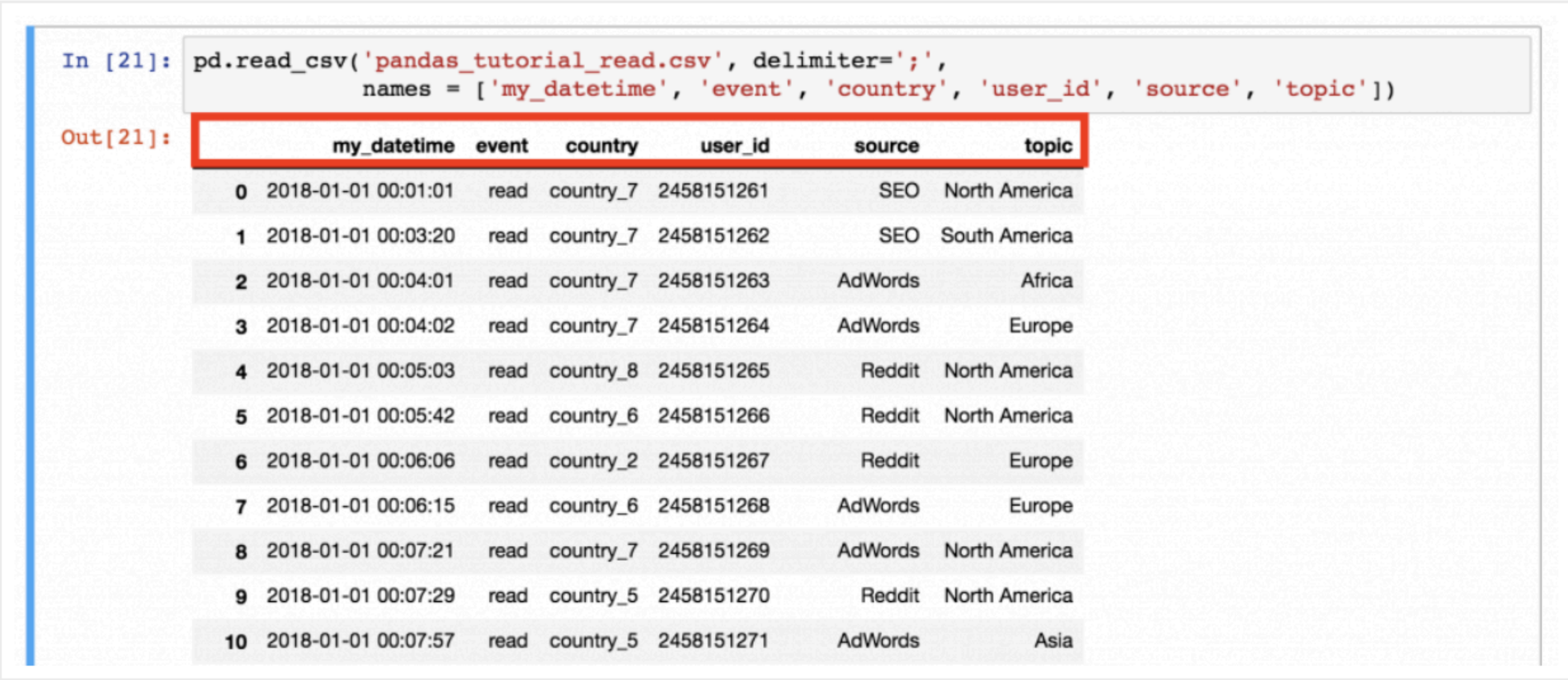
```
In [3]: pd.read_csv('pandas_tutorial_read.csv', delimiter=';')
```

Out[3]:

	2018-01-01 00:01:01	read	country_7	2458151261	SEO	North America
0	2018-01-01 00:03:20	read	country_7	2458151262	SEO	South America
1	2018-01-01 00:04:01	read	country_7	2458151263	AdWords	Africa
2	2018-01-01 00:04:02	read	country_7	2458151264	AdWords	Europe
3	2018-01-01 00:05:03	read	country_8	2458151265	Reddit	North America
4	2018-01-01 00:05:42	read	country_6	2458151266	Reddit	North America
5	2018-01-01 00:06:06	read	country_2	2458151267	Reddit	Europe
6	2018-01-01 00:06:15	read	country_6	2458151268	AdWords	Europe
7	2018-01-01 00:07:21	read	country_7	2458151269	AdWords	North America
8	2018-01-01 00:07:29	read	country_5	2458151270	Reddit	North America
9	2018-01-01 00:07:57	read	country_5	2458151271	AdWords	Asia
10	2018-01-01 00:08:57	read	country_7	2458151272	SEO	Australia

Memberi Judul Kolom

```
pd.read_csv('pandas_tutorial_read.csv', delimiter=';', names =
['my_datetime', 'event', 'country', 'user_id', 'source', 'topic'])
```



```
In [21]: pd.read_csv('pandas_tutorial_read.csv', delimiter=';',
                     names = ['my_datetime', 'event', 'country', 'user_id', 'source', 'topic'])
```

```
Out[21]:
           my_datetime  event    country     user_id     source          topic
     0  2018-01-01 00:01:01   read  country_7  2458151261        SEO  North America
     1  2018-01-01 00:03:20   read  country_7  2458151262        SEO  South America
     2  2018-01-01 00:04:01   read  country_7  2458151263    AdWords         Africa
     3  2018-01-01 00:04:02   read  country_7  2458151264    AdWords         Europe
     4  2018-01-01 00:05:03   read  country_8  2458151265     Reddit  North America
     5  2018-01-01 00:05:42   read  country_6  2458151266     Reddit  North America
     6  2018-01-01 00:06:06   read  country_2  2458151267     Reddit         Europe
     7  2018-01-01 00:06:15   read  country_6  2458151268    AdWords         Europe
     8  2018-01-01 00:07:21   read  country_7  2458151269    AdWords  North America
     9  2018-01-01 00:07:29   read  country_5  2458151270     Reddit  North America
    10  2018-01-01 00:07:57   read  country_5  2458151271    AdWords           Asia
```

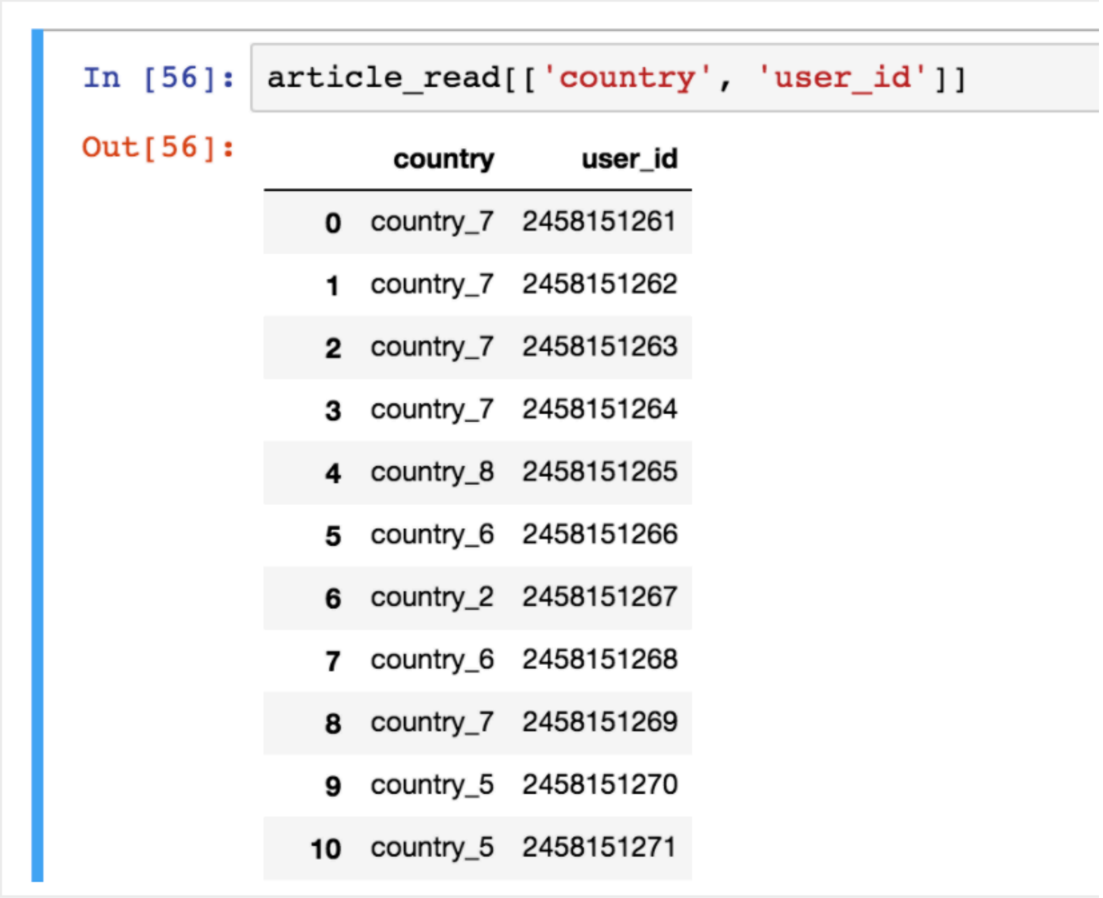
Selecting Data

```
article_read.head()
```

```
article_read.tail()
```

```
article_read.sample(5)
```

```
article_read[['country', 'user_id']]
```



```
In [56]: article_read[['country', 'user_id']]
Out[56]:
```

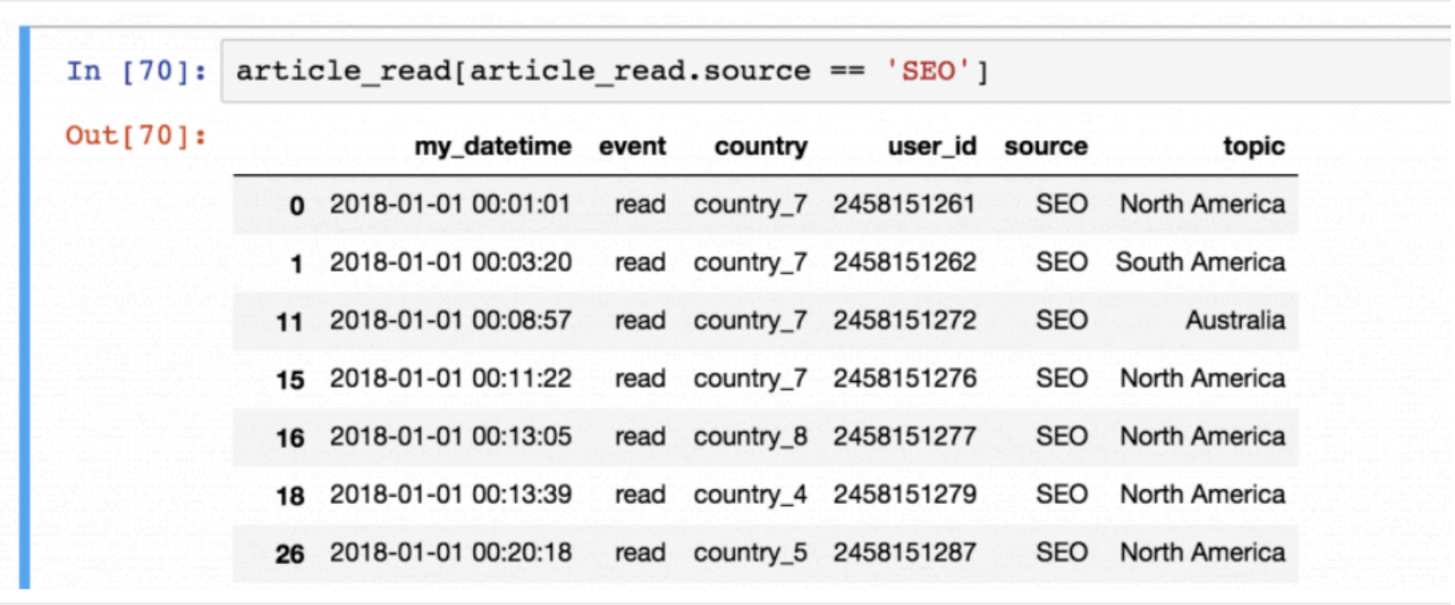
	country	user_id
0	country_7	2458151261
1	country_7	2458151262
2	country_7	2458151263
3	country_7	2458151264
4	country_8	2458151265
5	country_6	2458151266
6	country_2	2458151267
7	country_6	2458151268
8	country_7	2458151269
9	country_5	2458151270
10	country_5	2458151271

Query Data

```
article_read.user_id
```

```
article_read['user_id']
```

```
article_read[article_read.source == 'SEO']
```



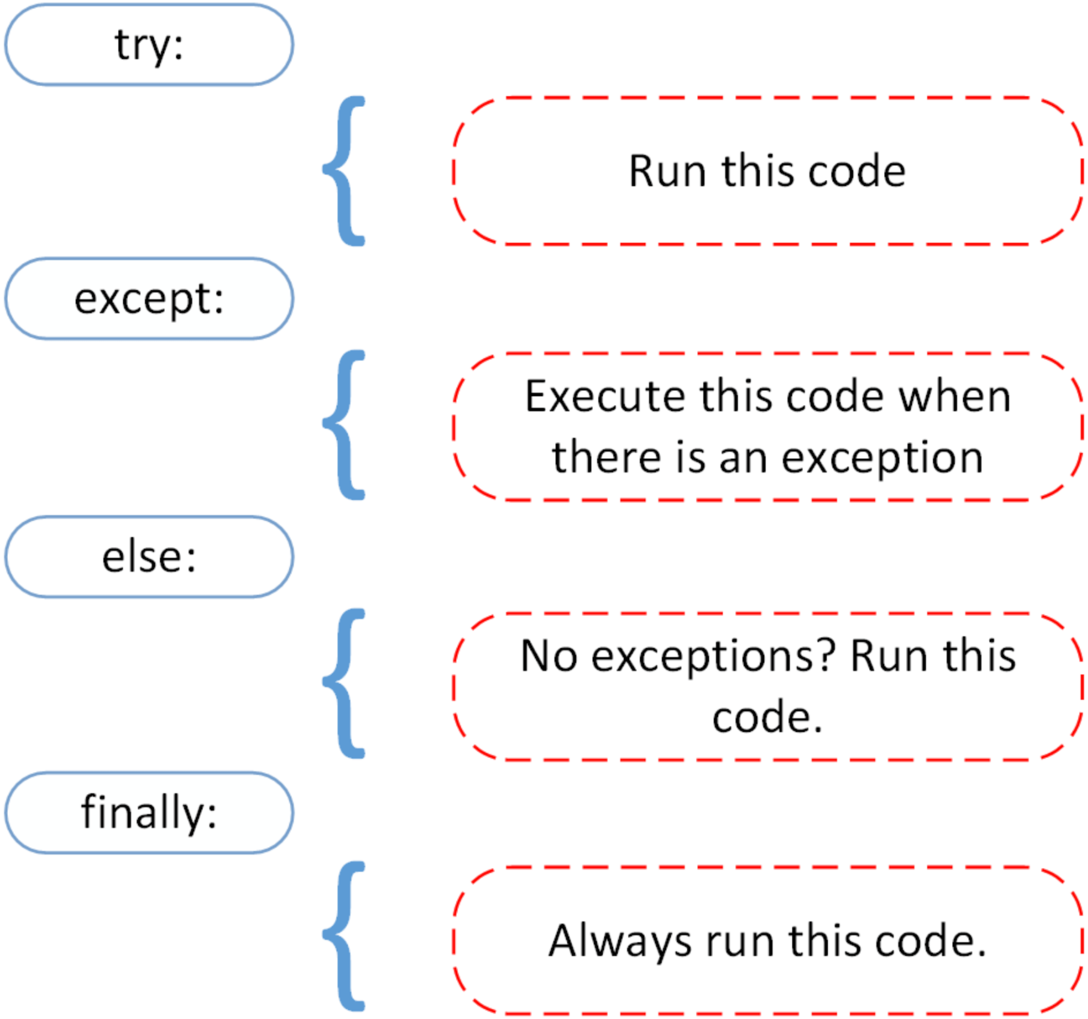
```
In [70]: article_read[article_read.source == 'SEO']
```

Out[70]:

	my_datetime	event	country	user_id	source	topic
0	2018-01-01 00:01:01	read	country_7	2458151261	SEO	North America
1	2018-01-01 00:03:20	read	country_7	2458151262	SEO	South America
11	2018-01-01 00:08:57	read	country_7	2458151272	SEO	Australia
15	2018-01-01 00:11:22	read	country_7	2458151276	SEO	North America
16	2018-01-01 00:13:05	read	country_8	2458151277	SEO	North America
18	2018-01-01 00:13:39	read	country_4	2458151279	SEO	North America
26	2018-01-01 00:20:18	read	country_5	2458151287	SEO	North America

Banyak berlatih dan biasakan diri
Mengolah data menggunakan
Pandas

Try Except



try:

Run this code

except:

Execute this code when there is an exception

else:

No exceptions? Run this code.

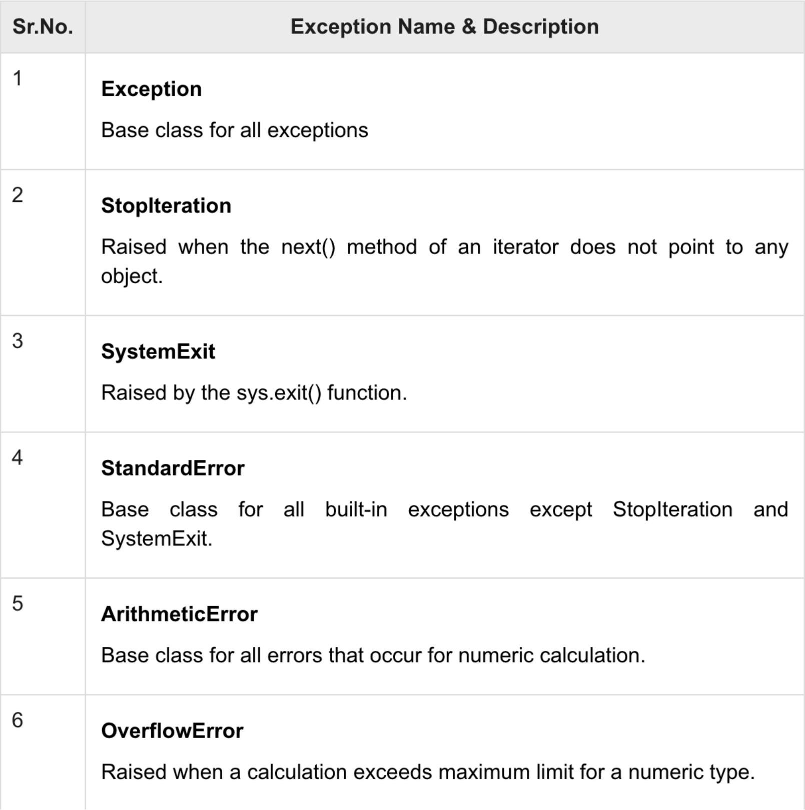
finally:

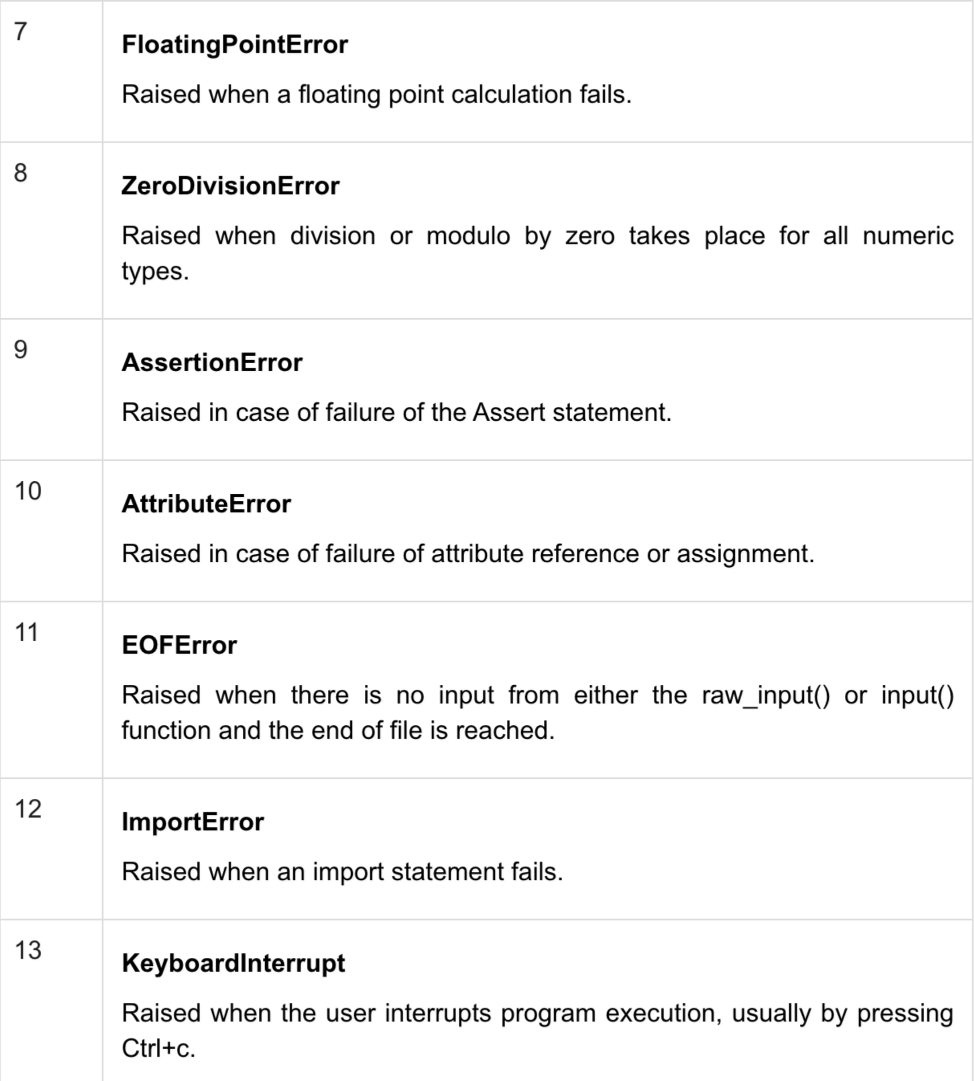
Always run this code.

Features disediakan Python untuk Penanganan Error

- Exceptions
- Assertions

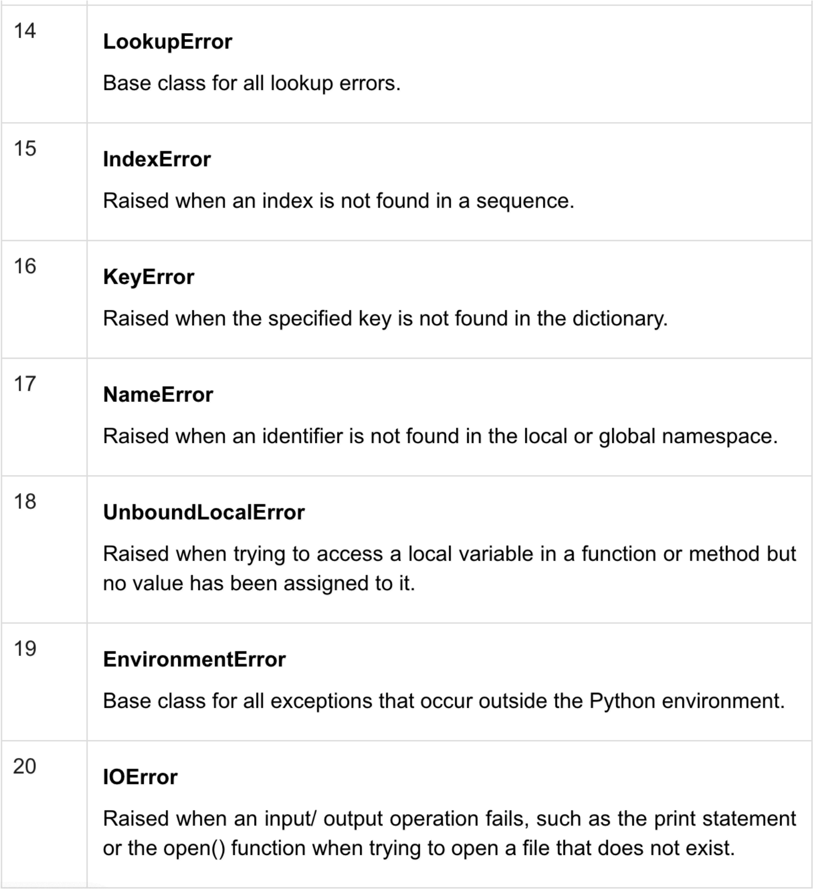
Exceptions



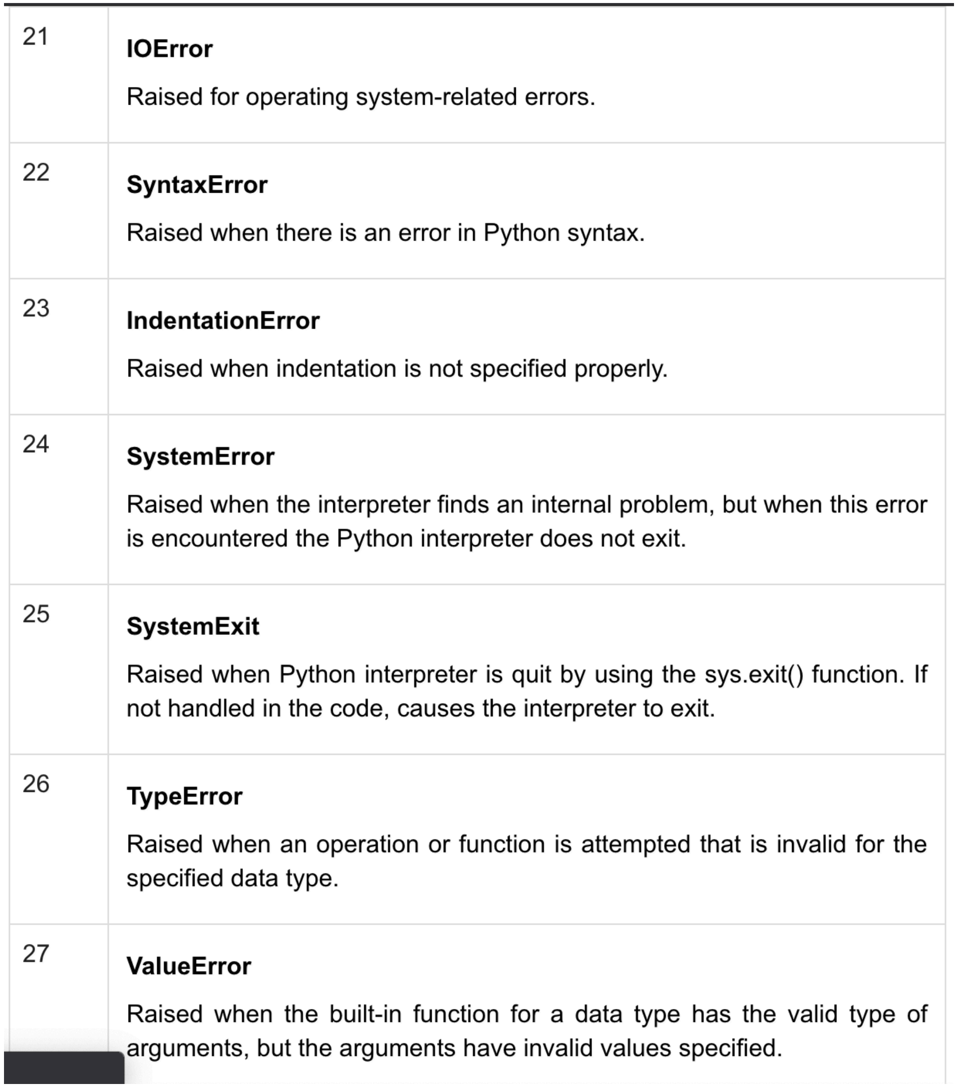


Sr.No.	Exception Name & Description
1	**Exception** Base class for all exceptions
2	**StopIteration** Raised when the next() method of an iterator does not point to any object.
3	**SystemExit** Raised by the sys.exit() function.
4	**StandardError** Base class for all built-in exceptions except StopIteration and SystemExit.
5	**ArithmeticError** Base class for all errors that occur for numeric calculation.
6	**OverflowError** Raised when a calculation exceeds maximum limit for a numeric type.
7	**FloatingPointError** Raised when a floating point calculation fails.
8	**ZeroDivisionError** Raised when division or modulo by zero takes place for all numeric types.
9	**AssertionError** Raised in case of failure of the Assert statement.
10	**AttributeError** Raised in case of failure of attribute reference or assignment.
11	**EOFError** Raised when there is no input from either the raw_input() or input() function and the end of file is reached.
12	**ImportError** Raised when an import statement fails.
13	**KeyboardInterrupt** Raised when the user interrupts program execution, usually by pressing Ctrl+c.

Exceptions

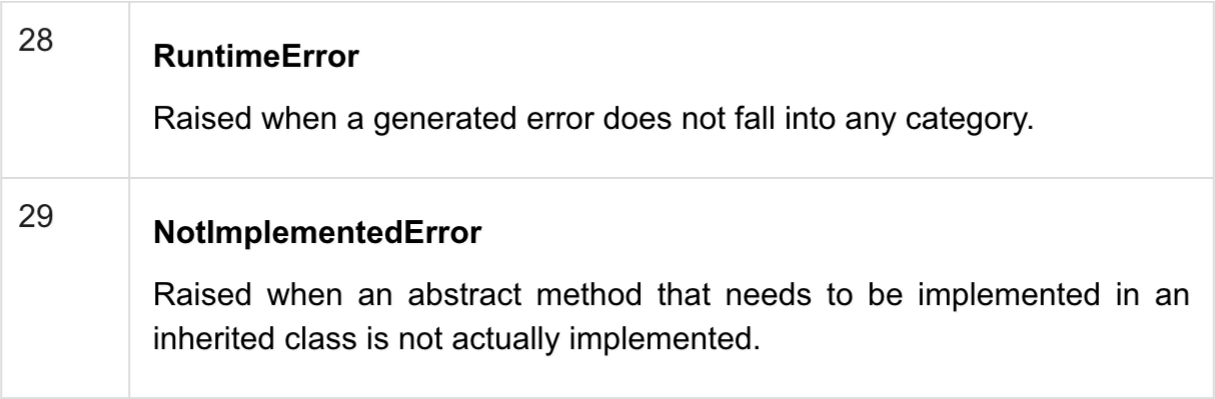


14	**LookupError** Base class for all lookup errors.
15	**IndexError** Raised when an index is not found in a sequence.
16	**KeyError** Raised when the specified key is not found in the dictionary.
17	**NameError** Raised when an identifier is not found in the local or global namespace.
18	**UnboundLocalError** Raised when trying to access a local variable in a function or method but no value has been assigned to it.
19	**EnvironmentError** Base class for all exceptions that occur outside the Python environment.
20	**IOError** Raised when an input/ output operation fails, such as the print statement or the open() function when trying to open a file that does not exist.



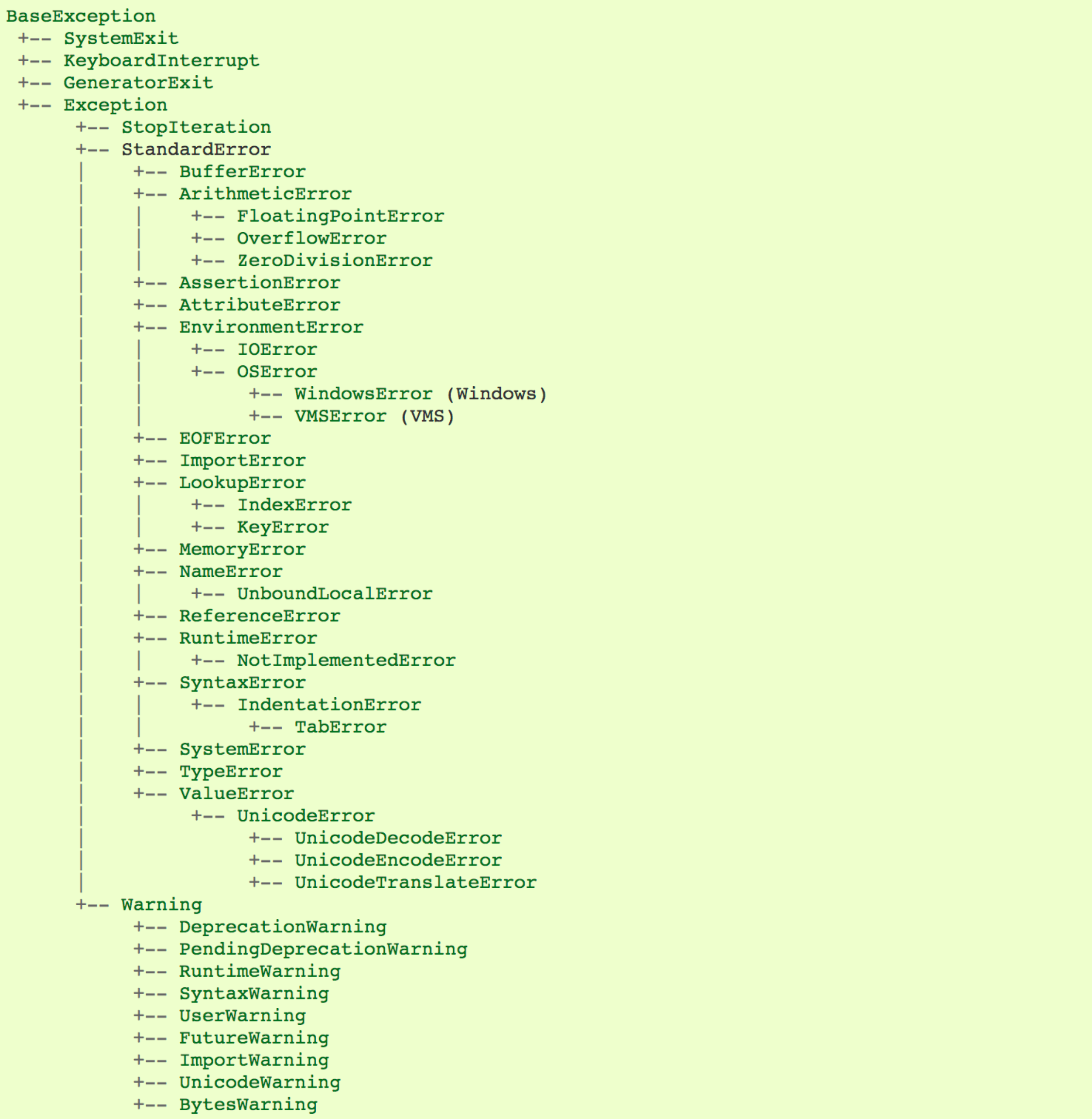
21	**IOError** Raised for operating system-related errors.
22	**SyntaxError** Raised when there is an error in Python syntax.
23	**IndentationError** Raised when indentation is not specified properly.
24	**SystemError** Raised when the interpreter finds an internal problem, but when this error is encountered the Python interpreter does not exit.
25	**SystemExit** Raised when Python interpreter is quit by using the sys.exit() function. If not handled in the code, causes the interpreter to exit.
26	**TypeError** Raised when an operation or function is attempted that is invalid for the specified data type.
27	**ValueError** Raised when the built-in function for a data type has the valid type of arguments, but the arguments have invalid values specified.

Exceptions



28	**RuntimeError** Raised when a generated error does not fall into any category.
29	**NotImplementedError** Raised when an abstract method that needs to be implemented in an inherited class is not actually implemented.

Tingkatan Exceptions



```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- StopIteration
      +-- StandardError
      |    +-- BufferError
      |    +-- ArithmeticError
      |    |    +-- FloatingPointError
      |    |    +-- OverflowError
      |    |    +-- ZeroDivisionError
      |    +-- AssertionError
      |    +-- AttributeError
      |    +-- EnvironmentError
      |    |    +-- IOError
      |    |    +-- OSError
      |    |         +-- WindowsError (Windows)
      |    |         +-- VMSError (VMS)
      |    +-- EOFError
      |    +-- ImportError
      |    +-- LookupError
      |    |    +-- IndexError
      |    |    +-- KeyError
      |    +-- MemoryError
      |    +-- NameError
      |    |    +-- UnboundLocalError
      |    +-- ReferenceError
      |    +-- RuntimeError
      |    |    +-- NotImplementedError
      |    +-- SyntaxError
      |    |    +-- IndentationError
      |    |         +-- TabError
      |    +-- SystemError
      |    +-- TypeError
      |    +-- ValueError
      |         +-- UnicodeError
      |              +-- UnicodeDecodeError
      |              +-- UnicodeEncodeError
      |              +-- UnicodeTranslateError
      +-- Warning
           +-- DeprecationWarning
           +-- PendingDeprecationWarning
           +-- RuntimeWarning
           +-- SyntaxWarning
           +-- UserWarning
           +-- FutureWarning
           +-- ImportWarning
           +-- UnicodeWarning
           +-- BytesWarning
```

Example: nocatch.py

```
fin = open('bad_file')
for line in fin:
    print line
fin.close()
```

Example: catch.py

```
try:
    fin = open('bad_file')
    for line in fin:
        print line
    fin.close()
except:
    print 'Something went wrong.'
```

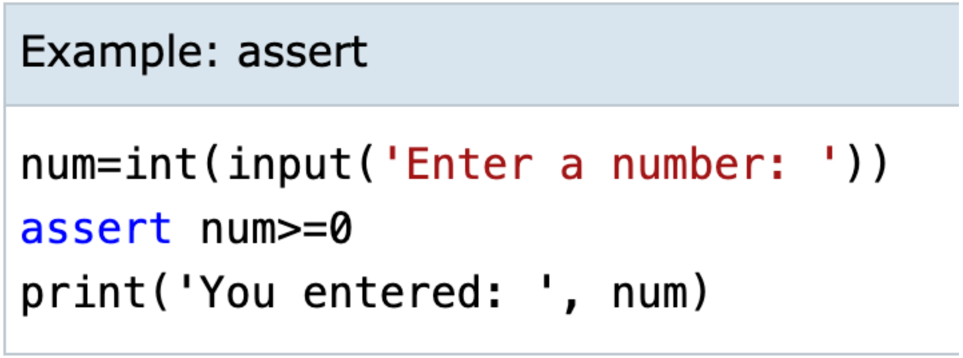
Example: catch2.py

```
try:
    fin = open('bad_file')
    for line in fin:
        print line
    fin.close()
except IOError:
    print 'Something went wrong.'
```

Assertions

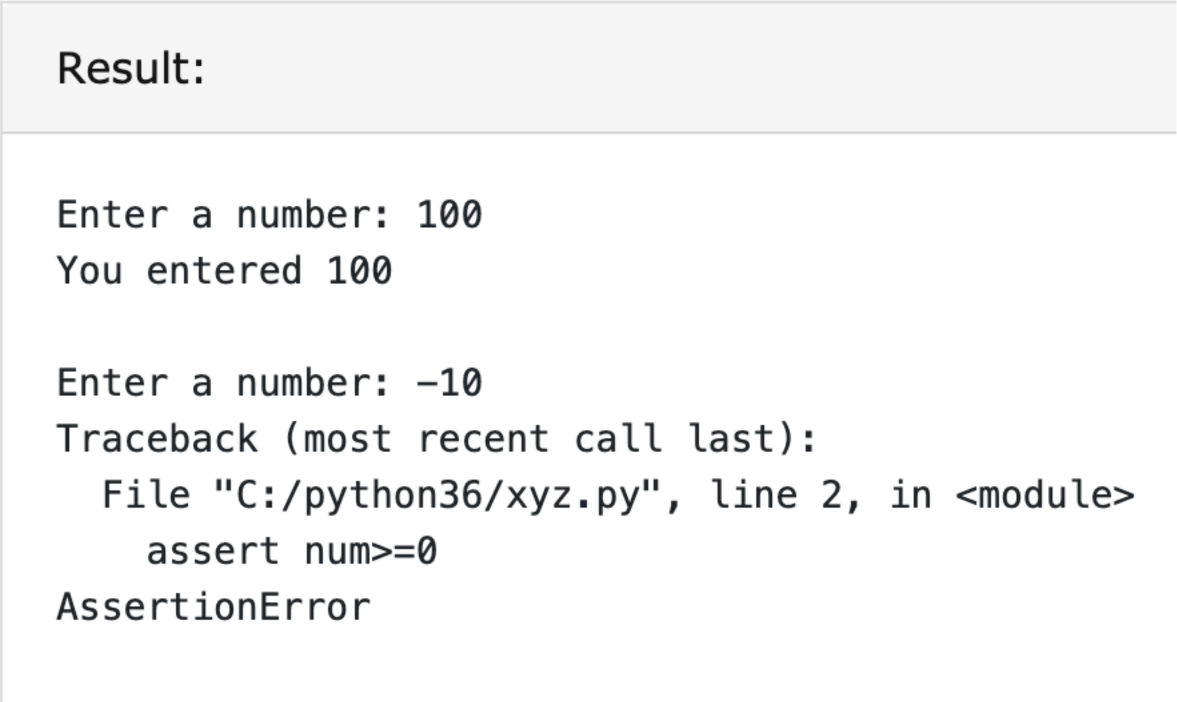
- Ekspresi Logika (True/False)
- Program hanya akan dieksekusi ketika pernyataan benar (True) dan akan menghasilkan **AssertionError** ketika salah (False)

Contoh Assert



Example: assert

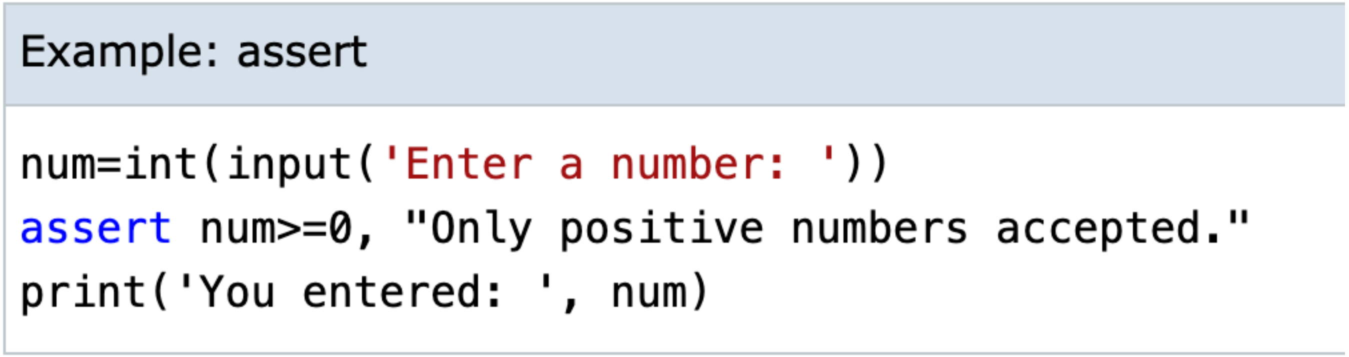
```
num=int(input('Enter a number: '))
assert num>=0
print('You entered: ', num)
```



Result:

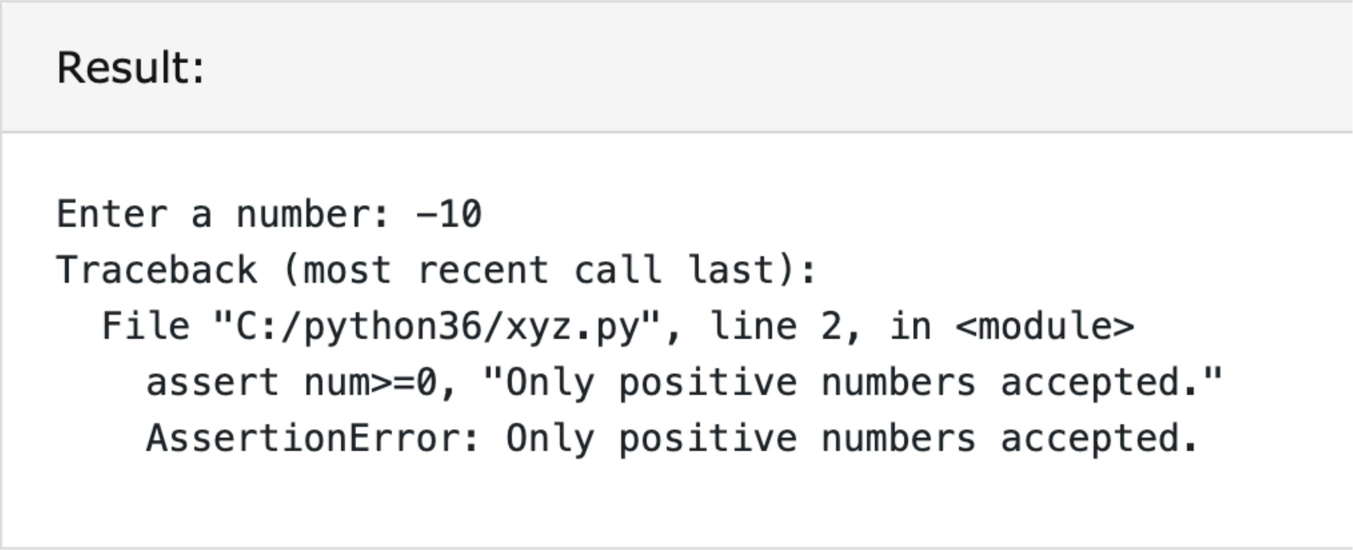
```
Enter a number: 100
You entered 100

Enter a number: -10
Traceback (most recent call last):
  File "C:/python36/xyz.py", line 2, in <module>
    assert num>=0
AssertionError
```

Contoh Assert



Example: assert

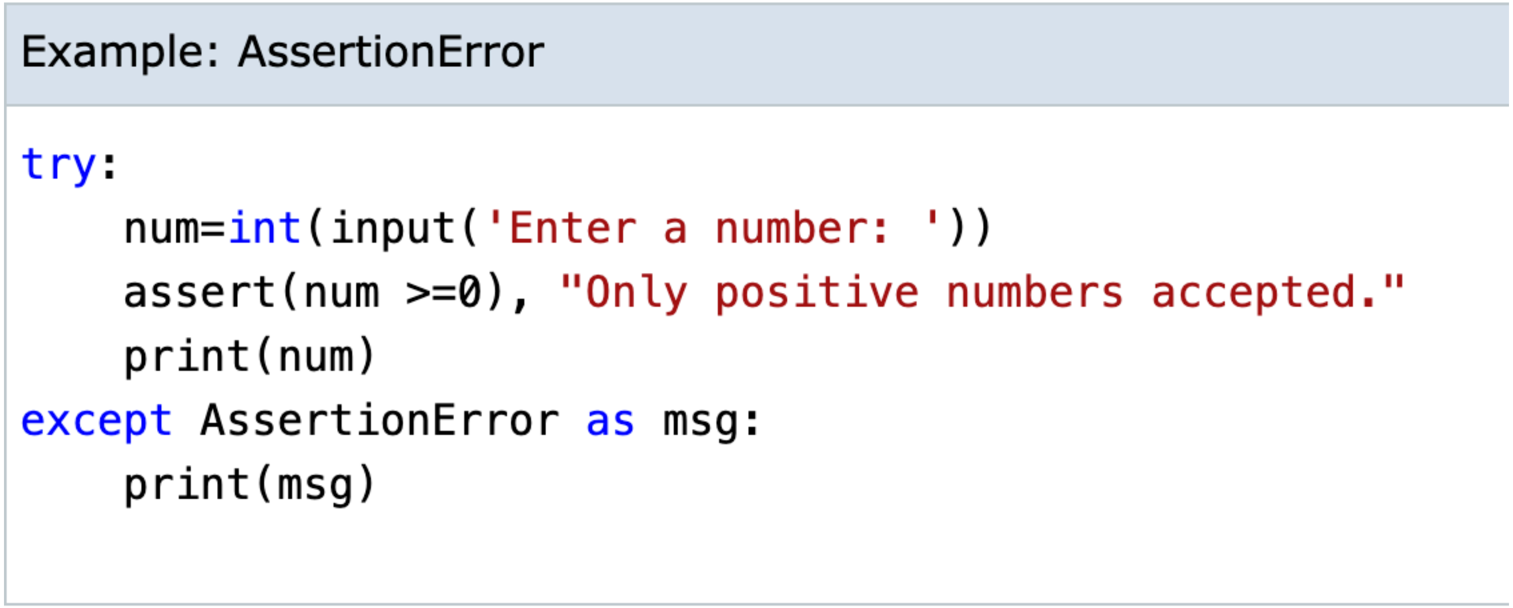
```
num=int(input('Enter a number: '))
assert num>=0, "Only positive numbers accepted."
print('You entered: ', num)
```



Result:

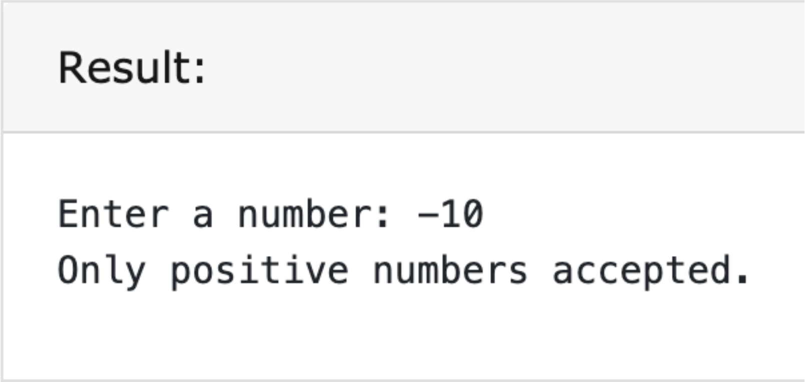
```
Enter a number: -10
Traceback (most recent call last):
  File "C:/python36/xyz.py", line 2, in <module>
    assert num>=0, "Only positive numbers accepted."
    AssertionError: Only positive numbers accepted.
```

Penanganan Assert



Example: AssertionError

```
try:
    num=int(input('Enter a number: '))
    assert(num >=0), "Only positive numbers accepted."
    print(num)
except AssertionError as msg:
    print(msg)
```

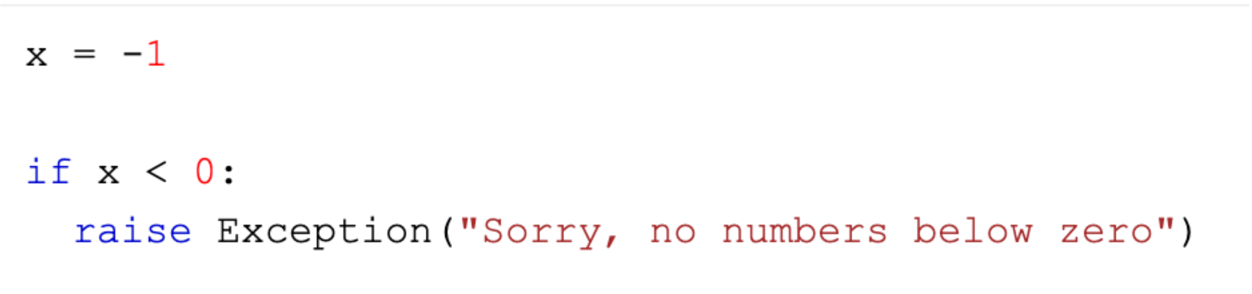


Result:

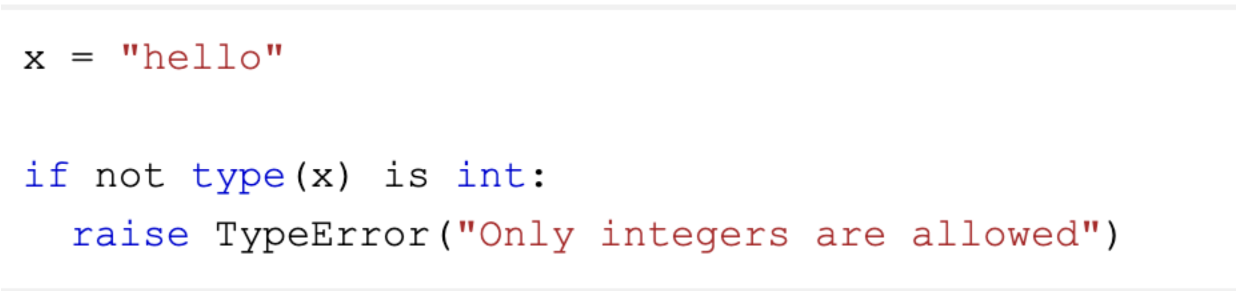
```
Enter a number: -10
Only positive numbers accepted.
```

Raise

- Perintah untuk memunculkan Exception / Error ketika kondisi terpenuhi



```
x = -1

if x < 0:
  raise Exception("Sorry, no numbers below zero")
```



```
x = "hello"

if not type(x) is int:
  raise TypeError("Only integers are allowed")
```

Terima Kasih