

প্রোগ্রাম করার সময় আমরা প্রায়ই ভুল করে থাকি। কিন্তু তা আমাদের চোখ এড়িয়ে যায়। ফলে আমরা যখন প্রোগ্রাম রান করি, তখন কমপিউটার এরর মেসেজ দেখায় এবং প্রোগ্রাম বন্ধ করে দেয়। ফলে আমরা আমাদের কাজক্ষিত ফল পাই না। পূর্ববর্তী অধ্যায়গুলোতে যে প্রোগ্রামগুলো দেখানো হয়েছিল, তাতে অনেক সময়ই এরর দেখা গেছে। এর মধ্যে অন্তত দুই ধরনের এরর খুব স্বাভাবিক- সিনট্যাক্স এরর এবং এক্সেপশন। সাধারণত যেসব এরর বেশি দেখা যায় সে এররগুলো কীভাবে হ্যান্ডেল করতে হয় তা দেখানো হয়েছে এ লেখায়।

### সিনট্যাক্স এরর

সিনট্যাক্স এরর, যা পার্সিং এরর নামেও পরিচিত, যা প্রোগ্রামিং শেখার সময়ে সবচেয়ে

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'spam' is not defined
>>> '2'+2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Can't convert 'int' object to
str implicitly
```

এরর ম্যাসেজের লাস্ট লাইনে কি টাইপের এরর হয়েছে তা উল্লেখ থাকে। উপরিলিখিত উদাহরণে তিন ধরনের এক্সেপশন দেখা যাচ্ছে- ZeroDivisionError, NameError এবং TypeError। এগুলো বিল্টইন এররের নাম। ম্যাসেজের শুরুতে কী ধরনের এরর হয়েছে তার নাম উল্লেখ থাকে। এরপরের অংশে কী কারণে এই এরর হয়েছে তা বলা থাকে।

```
... print("Unexpected error:",
sys.exc_info()[0])
... raise
...
OS error: [Errno 2] No such file or
directory: 'myfile.txt'
```

Try ... except ব্লকের সাথে else ব্যবহার করা যায়। try-এর ভেতরে অতিরিক্ত কোড লেখার চেয়ে else ব্যবহার করা বেশি সুবিধাজনক।

```
for arg in sys.argv[1:]:
  try:
    f = open(arg, 'r')
  except IOError:
    print('cannot open', arg)
  else:
    print(arg, 'has', len(f.readlines()),
'lines')
  f.close()
```

try-এর আরও একটি অপশনাল ক্লজ আছে ফাইনাল নামে। এটি try-এর শেষে লেখা হয়। এর ভেতরের কোডগুলো শেষ পর্যন্ত রান হয়। এ ক্ষেত্রে দেখা হয় না try-এর ভেতরের কোডগুলো এক্সিকিউট হলো কি না।

```
>>> try:
      raise KeyboardInterrupt
finally:
  print("Goodbye, world!")
Goodbye, world!
Traceback (most recent call last):
  File "<pyshell#6>", line 2, in <mod-
ule>
    raise KeyboardInterrupt
KeyboardInterrupt
```

বাস্তবে finally ক্লজ খুবই গুরুত্বপূর্ণ, বিশেষ করে এক্সটারনাল রিসোর্স রিলিজ করার ক্ষেত্রে। এ ক্ষেত্রে রিসোর্সটি ব্যবহার হলো কি না তা মুখ্য নয়।

ফিডব্যাক : [ahmadalsajid@gmail.com](mailto:ahmadalsajid@gmail.com)



## পাইথনে হাতেখড়ি

আহমাদ আল-সাজিদ

বেশি দেখা যায়। সাধারণত টাইপিংয়ের ভুলের কারণে এই ধরনের এরর হয়।

```
>>> while True print("Hello World")
File "<stdin>", line 1
  while True print("Hello World")
      ^
SyntaxError: invalid syntax
```

SyntaxError : invalid syntax

এই ক্ষেত্রে পার্সার যে লাইনে ভুল আছে, তা দেখায় এবং কোথায় ভুলটা ধরা পড়েছে তা একটি ছোট অ্যারোর সাহায্যে নির্দেশ করে। উপরের উদাহরণের ক্ষেত্রে বলা যায়, print() ফাংশনে ভুল ডিটেক্ট হয়েছে। কারণ, এর আগে কোলন (;) চিহ্ন থাকার কথা, যা এখানে অনুপস্থিত। কোন ফাইলে ভুল হয়েছে এবং কত নম্বর লাইনে ভুল আছে তা দেখানো হয় যাতে স্ক্রিপ্ট থেকে ইনপুটের সময় সহজে ভুলটা সংশোধন করা যায়।

### এক্সেপশন

যদি একটি স্টেটমেন্ট বা এক্সপ্রেশন সিনট্যাক্সের দিক থেকে ঠিকও হয়, এটি এক্সিকিউটের সময় এরর দেখাতে পারে। এক্সিকিউটের সময় যে এরর বা ভুল পাওয়া যায় তাকে এক্সেপশন বলে আর এটা খুব গুরুতর নাও হতে পারে। বেশিরভাগ এক্সেপশন প্রোগ্রাম নিজে হ্যান্ডেল করে না, শুধু এরর ম্যাসেজ প্রিন্ট করে।

```
>>> 10*(1/0)
```

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
>>> 4 + spam*3
```

নির্দিষ্ট ধরনের এরর হ্যান্ডেল করার জন্য প্রোগ্রাম লেখা সম্ভব। যেমন- নিচের উদাহরণে দেখা যায় একটি প্রোগ্রাম লেখা হয়েছে, যা ইউজারের থেকে ভ্যালিড নাম্বার ইনপুট নিতেই থাকবে যতক্ষণ না Ctrl+c চেপে প্রোগ্রাম বন্ধ করে দেয়া হয়।

```
>>> while True:
...   try:
...     x = int(input("enter a number:"))
...     break
...   except ValueError:
...     print("that is not a number")
... 
```

Try...except-এর কাজের ধারা হচ্ছে এটি প্রথমে Try ব্লক এক্সিকিউট করার চেষ্টা করে, যদি কোনো এরর না থাকে তাহলে প্রোগ্রাম আর except ব্লকে যায় না। কিন্তু যদি কোনো এরর দেখা দেয়, তাহলে Try ব্লকের আর কোনো এক্সপ্রেশন এক্সিকিউট হয় না, প্রোগ্রাম এরপর except ব্লকে চলে যায়।

```
>>> import sys
>>> try:
...   f = open('myfile.txt')
...   s = f.readline()
...   i = int(s.strip())
... except OSError as err:
...   print("OS error: {0}".format(err))
... except ValueError:
...   print("Could not convert data to an
integer.")
... except:
```

## জেনে নিন

### এক্সেল ব্যবহারকারীদের জন্য গুরুত্বপূর্ণ শর্টকাট কী

Ctrl+Arrow : ডানে, বামে, ওপরে এবং নিচে লেখার শেষে কারসর যাবে।

Ctrl+Page Up : আগের পৃষ্ঠা বা ওয়ার্কশিটে যাওয়া।

Ctrl+Page Down : পরের পৃষ্ঠা বা ওয়ার্কশিটে যাওয়া।

Atl+Page Up : ডকুমেন্টের প্রথম কলামে অবস্থান করা।

Atl+Page Down : ডকুমেন্টের শেষ কলামে অবস্থান করা।

Atl+Enter : ফিল্ডে কারসর রেখে দুই ক্লিকের মাধ্যমে পরের লাইন তৈরি করা।

Shift+TAB : পেছনের ফিল্ড থেকে প্রথম ফিল্ডে একেক করে যাওয়া।