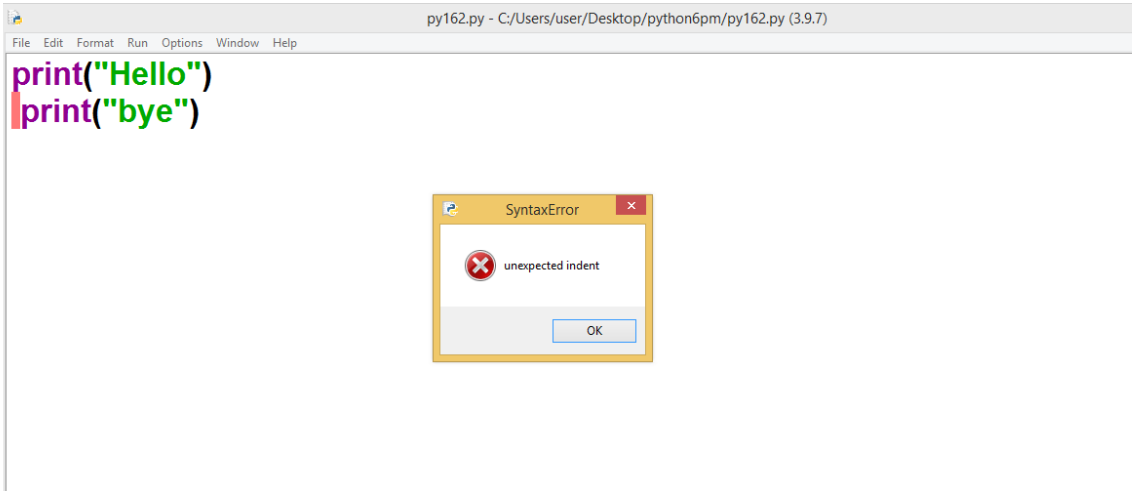
## Exception Handling or Error Handling

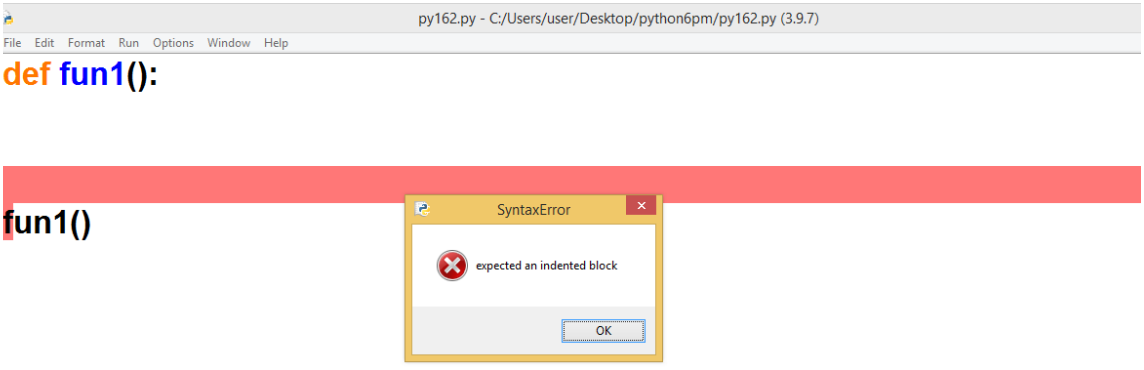
### Types of errors

1. Compile time errors
2. Runtime errors

### Compile time error

The errors which occur during compiling of program are called compile time error. These compile time errors are also called syntax errors.
If there is syntax error within program, program is not executed.




These errors must be rectified by programmer in order to execute program.

### Runtime errors

The errors which occur during execution of program are called runtime errors. All logical errors are called runtime errors

These errors occur because of wrong input given by end user or input given by end user does not satisfy condition given within program.
When there is runtime error, program execution is terminated. To avoid abnormal termination of program we use exception handlers or error handlers.

**What is exception?**
Exception is a runtime error or logical error
Exception is an error which occur during execution of program

**Advantage of exception handling or error handling**

1. Avoiding abnormal termination of program
2. To convert predefined error message to user defined error message
3. To separate business and error logic

Every exception is one data type or class
Creating object of exception class and giving to PVM is called generating exception.
Functions generate exception/error
Every error or exception is one class/data type
These exceptions are two

1. Predefined exception
2. User defined exception

**Predefined exception**
The exception type/class provided by python is called predefined exceptions. These exceptions are used by python libraries or library functions.

Int → ValueError
Float → ValueError
List → TypeError
Dict → KeyError

**User defined exception or custom exception**
The error type build by programmer are called user defined exception or custom exception

**The keywords used to handle exceptions or errors**

1. **try**
2. **except**
3. **finally**

4. **raise**

**try block**
this block contains statements which has to monitored for exception handling or error handling (OR) the statements which raises error or exception are written inside try block

**Syntax:**
```
try:
   statement1
   statement2
```

**except block**
except is keyword
except block is error handler, which handle error raised by try block
OR if there is an error inside try block it is handled by except block

**Syntax:**
```
try:
  statement-1
  statement-2
  statement-3
except <error-type> as <object-name>:
        statement-4
except <error-type> as <object-name>:
        statement-5
```

try block followed by one or more than one except block. If multiple types of errors within try block, it is handled by multiple except blocks.

Example:
```
def main():
    try:
        n1=int(input("enter first number"))
        n2=int(input("enter second number"))
        n3=n1/n2
        print(n1,n2,n3)
    except ZeroDivisionError as a:
        print("cannot divide number with zero")
```

```
    print("continue...")

main()
```

**Output:**

```
enter first number5
enter second number2
5 2 2.5
continue...
>>>
========= RESTART: C:/Users/user/Desktop/python6pm/py164.py
=========
enter first number5
enter second number0
cannot divide number with zero
continue...
>>>
========= RESTART: C:/Users/user/Desktop/python6pm/py164.py
=========
enter first numberabc
Traceback (most recent call last):
  File "C:/Users/user/Desktop/python6pm/py164.py", line 14, in <module>
    main()
  File "C:/Users/user/Desktop/python6pm/py164.py", line 3, in main
    n1=int(input("enter first number"))
ValueError: invalid literal for int() with base 10: 'abc'
>>>
```