Ch 13 - Exception handling, environments

When we code, there is a probability of getting error. If we don't handle these errors, our program gets crashed. They can be handled by a try statement.

Be it a TypeError or ValueError, all types of errors can be handled in Python.

```
def fun(k):
    print("good")

fun(5,8) → TypeError
```

```
try:
    fun(5,8)
except:
    print("only 1 parameter accepted")
```

We can even get the produced errors like this:-

```
try:
    fun(5,8)
except Exception as e:
    print(e) → Fun() takes 1 positional argument but 2 were given.
```

- Exception type can also be specified

```
try: # Code to try

except KeyError:
    # Code

except SyntaxError:
    # Code

except ValueError:
    # Code

except KeyboardInterrupt: # Code
```

©OceanProgrammer

An example explaining multiple exceptions -

```
while 1: → (While True:)
    try:
        fun = lambda k: print("good", 1/int(k))
        fun(input())

    except ValueError as e:
        print("Enter an integer")

    except ZeroDivisionError as d:
        print("Enter a value not equal to zero")

    except TypeError as c:
        print(f"Some other error: {c}")

    except:
        print("Some other error occured")
```

try with else

```
try:
    i = int(input())

except Exception as e:
    print(e)

else: → This is executed only when code under try gets completely executed
    print("Done")
```

©OceanProgrammer

Exception handling with finally :-

```
try :
    # try-code
except:
    # except- code
finally:
    # Code → gets executed ignoring error/exception code.
```

raising exceptions in Python

```
try : #----

except Exception:
    raise TypeError("Your machine will blast")
```

error type / error message

⇒ Virtual enviornment

A virtual enviornment is used to manage Python packages for different projects. Using a virtualenv can let us install required packages for a project locally.

pip install virtualenv → install this module

virtualenv newenv → create a new detached interpretor

source mypython/bin/activate → activate env (Mac/Linux)

newenv\Scripts\activate → activate env (windows)

sometimes you may get error while activating env. solutions are always available on internet.

©OceanProgrammer

deactivate → Deactivate environment

☆ All modules in an environment along with their versions can be stored in a file.

pip freeze > requirements.txt

used to store output into a file.

Even the modules can be installed like →

pip install -r requirements.txt.

Map, filter and reduce functions

- map() function allows you to transform items in an iterable. It is useful when you use any transformation function.

```
def multiply(num):
    return num*2

l = [1,2,4]

print(list(map(multiply,l)))
→ [2,4,8]
```

- filter() function can be used to process an iterable an extract items that satisfy condition

```
def func(n):
    if (n<55): return True
    else: return False
```

©CodeProgrammer

```
l = [23,34,87,28]

print(list(filter(func, l)))
→ [23,34,28]
```

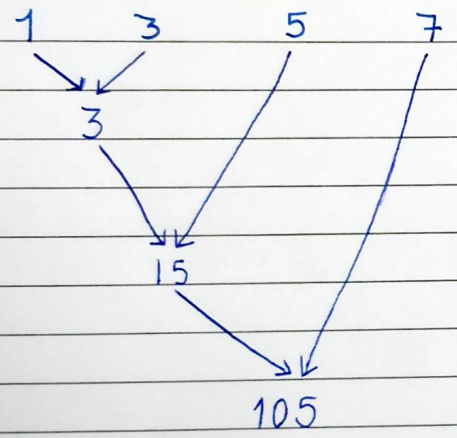
- reduce() function helps you to take a function, and then one by one, apply them to an iterable and at the end, get a final value.

```
from functools import reduce

def multiply(a,b):
    return a*b

l = [1,3,5,7]

prod = reduce(multiply, l)
print(prod)
→ 105
```



©OceanProgrammer