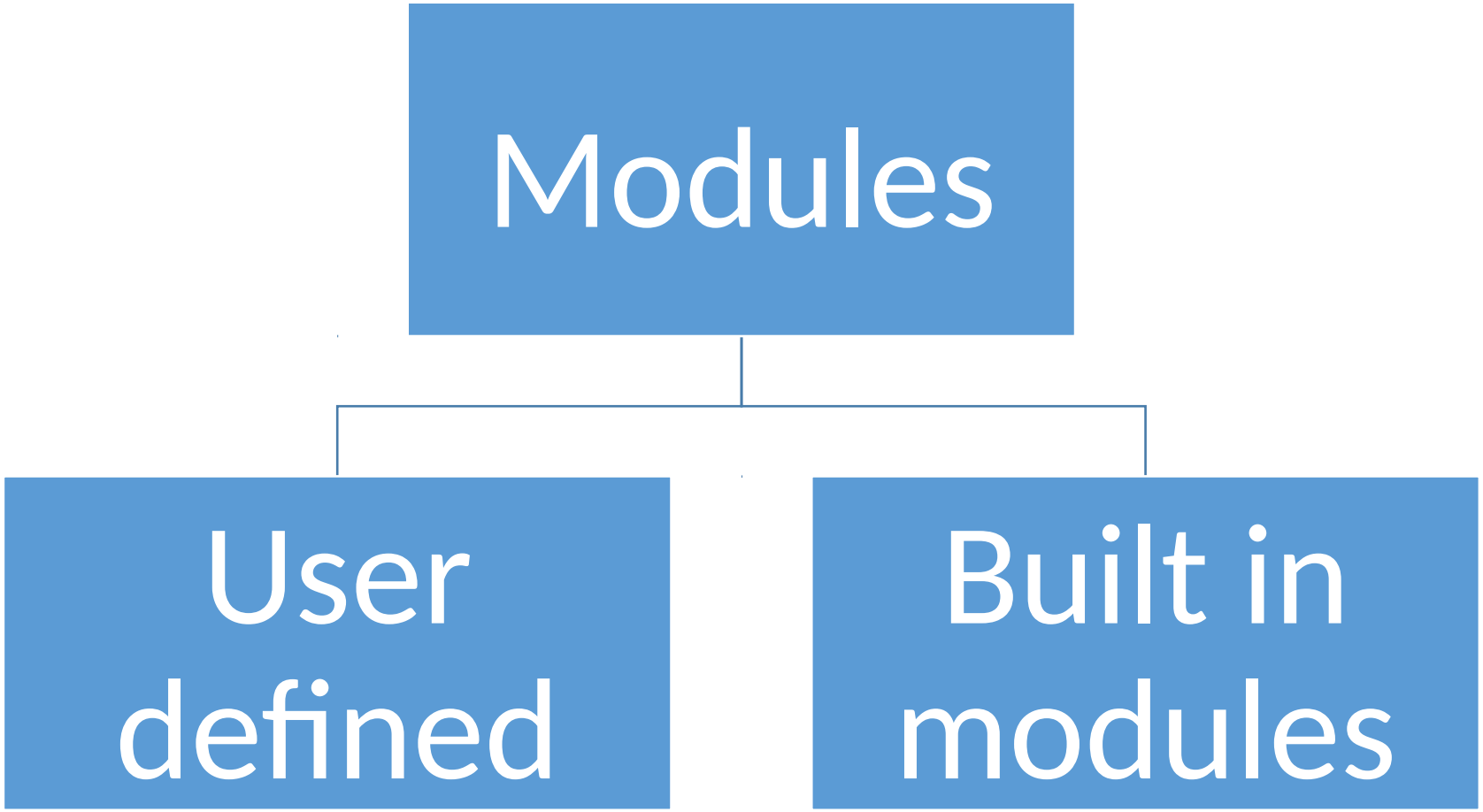
Python problem set 2 a few common modules

Kshithij Iyer

Why modules?

- When people write huge programs in python they write the code in multiple files.
- And in this case each file is a module. And modules can be imported into other modules by writing “import <modulename>”.
- The module name, to import, has the same name of the Python file without the .py extension.
- Modules are nothing but files with Python definitions and statements.

There are 2 types of modules



So last time we saw.

```
import time

#Recording start time of the program

start=time.time()


print("Execution time of program is",time.time()-start)
```

- In this case time is a built in module.
- An the function time() is a part of the time module.

Other methods in time module

>>> import time

>>> time.ctime() #Convert a time expressed in seconds since the epoch to a string representing local time.

'Tue May 9 23:54:59 2017‘

>>> time.time() #Return the time in seconds since the epoch as a floating point number.

1494354395.2106404

>>> time.clock() #returns the time of the processor clock

1.6420785923445476e-06

>>time.sleep(10) #sleeps the thread for 10 sec

Os module

```
>>> import os
>>> os.getuid()
500
>>> os.getpid()
16150
>>> os.getppid()
14847
>>> os.uname()
('Linux', 'd80', '2.6.34.7-56.fc13.i686.PAE', '#1 SMP Wed Sep 15 03:27:15 UTC 2010',
'i686')
```

```
>>> os.getcwd()
'C:\\Users\\kshithij\\AppData\\Local\\Programs\\Python\\Python35‘
>>>os.chdir("G:\\placement\\python\\")
>>> os.mkdir('G:\\placement\\python\\mkdir')
>>>
os.rename('G:\\placement\\python\\mkdir','G:\\placement\\python\\haha')
>>> os.rmdir('G:\\placement\\python\\mkdir')
```

tarfile module

```
>>> import tarfile
>>> mytar=tarfile.open("G:\\placement\\python\\mytar.tar")
>>> mytar.getnames()
>>> mytar.getmembers() #tarinfo object
>>mytar.extractall()
>>> mytar.getmember('./consumerproducer.c')
<TarInfo './consumerproducer.c' at 0x3448688>
>>>mytar.extract("./consumerproducer.c")
>>mytar.close()
```

A user defined module

```
def starbar(num):
    """Prints a bar with *:arg num: Length of the bar"""
    print('*' * num)
def hashbar(num):
    """Prints a bar with #:arg num: Length of the bar"""
    print('#' * num)
def simplebar(num):
    """Prints a bar with -:arg num: Length of the bar"""
    print('-' * num)
```

And this is how built in modules make your life easy!

#Euler 19

#Problem:

#You are given the following information, but you may prefer to do some research for yourself.

#1 Jan 1900 was a Monday.

#Thirty days has September,

#April, June and November.

#All the rest have thirty-one,

#Saving February alone,

#Which has twenty-eight, rain or shine.

#And on leap years, twenty-nine.

#A leap year occurs on any year evenly divisible by 4, but not on a century unless it is divisible by 400.

#How many Sundays fell on the first of the month during the twentieth century (1 Jan 1901 to 31 Dec 2000)?

```
import time
import datetime
#Recording start time of the program
start=time.time()

def Count_Sundays(From_year,To_year):
    "A function returns the count of Sundays"
    #A variable to keep the count of number of sundays
    number_of_sundays = 0
    for year in range(From_year,To_year+1):
        for month in range(1,13):
            # monday == 0, sunday == 6
            if datetime.datetime(year,month,1).weekday() == 6:
                number_of_sundays += 1
    return number_of_sundays

print("The number of sundays between 1901 to 2000 is",Count_Sundays(1901,2000));
print("Execution time of program is",time.time()-start)
```

Thank you

Rich people don’t buy liabilities they buy assets.

(From “Rich dad Poor dad” by Robert T Kiyosaki)