py4kids (https://github.com/wgong/py4kids)

Built-in libraries - Python standard lib

In this lesson, we learn several useful libraries in python:

- sys
- datetime
- random
- pickle
- urllib

Expanding our toolbox:

Use pre-built package:

In [1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

Out[1]:

System Info

In [2]:
```
# Which Version of Python Am I Using?

import sys
print(sys.version)
```

```
3.5.2 |Anaconda custom (64-bit)| (default, Jul  5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)]
```

In [3]:
```
sys.version_info
```

Out[3]:
```
sys.version_info(major=3, minor=5, micro=2, releaselevel='final', serial=0)
```

In [4]:
```
sys.version_info.major, sys.version_info.minor, sys.version_info.micro
```

Out[4]:
```
(3, 5, 2)
```

In [5]:
```
import platform
platform.python_version()
```

Out[5]:
```
'3.5.2'
```

Date and Time

In [6]:
```
# tell me the time
import time
print(time.time())
# number of seconds since January 1, 1970, at 00:00:00 AM
```

```
1508624518.9013386
```

In [7]:
```
# double-check
(2017-1970) * 365 * 24 * 60 * 60 + 10 * 30 * 24 * 60 * 60  # seconds
```

Out[7]:
```
1508112000
```

In [8]:
```
# measure time lapse of each loop
t1 = time.time()
for x in range(0, 1000):
    if x%100 == 0:
        print(x)
t2 = time.time()
print('it took %s seconds' % (t2-t1))
```

```
0
100
200
300
400
500
600
700
800
900
it took 0.0010001659393310547 seconds
```

In [9]:
```
t = time.asctime()
print(t)
```

```
Sat Oct 21 18:21:58 2017
```

In [10]:
```
local_time = time.localtime()
print(local_time)
year, month, day = t[0], t[1], t[2]
```

```
time.struct_time(tm_year=2017, tm_mon=10, tm_mday=21, tm_hour=18, tm_min=21, tm_sec=58, tm_wday=5, tm_yday=294,
tm_isdst=1)
```

In [11]:
```
# time for a nap
for x in range(1, 10):
    print(x)
    time.sleep(3)
```

```
1
2
3
4
5
6
7
8
9
```

In [12]:
```
# dates are easily constructed and formatted
from datetime import date
now = date.today()
now
```

Out[12]:
```
datetime.date(2017, 10, 21)
```

In [13]:
```
import datetime
past = datetime.date(2012, 12, 12)
past.strftime("%m-%d-%y. %d %b %Y is a %A on the %d day of %B.")
```

Out[13]:
```
'12-12-12. 12 Dec 2012 is a Wednesday on the 12 day of December.'
```

everything about string formating of time: http://strftime.org/ (http://strftime.org/)

In [14]:
```
# dates support calendar arithmetic
birthday = date(1999, 10, 1)
age = now - birthday
age.days
#14368
```

Out[14]:
```
6595
```

Random Number Generator

Throw a dice:

In [15]:
```
import random
print(random.randint(1, 100))     # pick a number randomly between 1 and 100
print(random.randint(100, 1000))  # pick a number randomly between 100 and 1000
```

```
56
147
```

In [25]:
```
# Guess a number between 1 and 100
import random
num = random.randint(1, 100)
while True:
    print('Guess a number between 1 and 100')
    guess = input()
    i = int(guess)
    if i == num:
        print('You guessed right')
        break
    elif i < num:
        print('Try higher')
    elif i > num:
        print('Try lower')
```

```
Guess a number between 1 and 100
50
Try higher
Guess a number between 1 and 100
75
Try lower
Guess a number between 1 and 100
62
Try lower
Guess a number between 1 and 100
56
Try lower
Guess a number between 1 and 100
53
Try lower
Guess a number between 1 and 100
52
Try lower
Guess a number between 1 and 100
51
You guessed right
```

In [27]:
```
import random
desserts = ['ice cream', 'pancakes', 'brownies', 'cookies', 'candy']
print(random.choice(desserts))

random.shuffle(desserts)
print(desserts)
```

```
pancakes
['pancakes', 'cookies', 'candy', 'brownies', 'ice cream']
```

a little lottory app

In [28]:
```
my_lucky_number = 7
l_counter = 0
while True:           # this is an infinite loop
    your_number_pick = random.randint(0,1000)
    l_counter = l_counter + 1
    print("[%03d] %5d" % (l_counter,your_number_pick))
    if your_number_pick % my_lucky_number == 0:
        print('\n*** Good Luck ***  You got a %d-multiple' % my_lucky_number)
        break         # will terminate the infinite loop
    else:
        continue      # go on forever
```

```
[001]   444
[002]   253
[003]   133

*** Good Luck ***  You got a 7-multiple
```

Persist python

In [29]:
```
import pickle
game_data = {
    'player-position' : 'N23 E45',
    'pockets' : ['keys', 'pocket knife', 'polished stone'],
    'backpack' : ['rope', 'hammer', 'apple'],
    'money' : 158.50
}
save_file = open('py-pickle.dat', 'wb')
pickle.dump(game_data, save_file)
save_file.close()
# look at the file
```

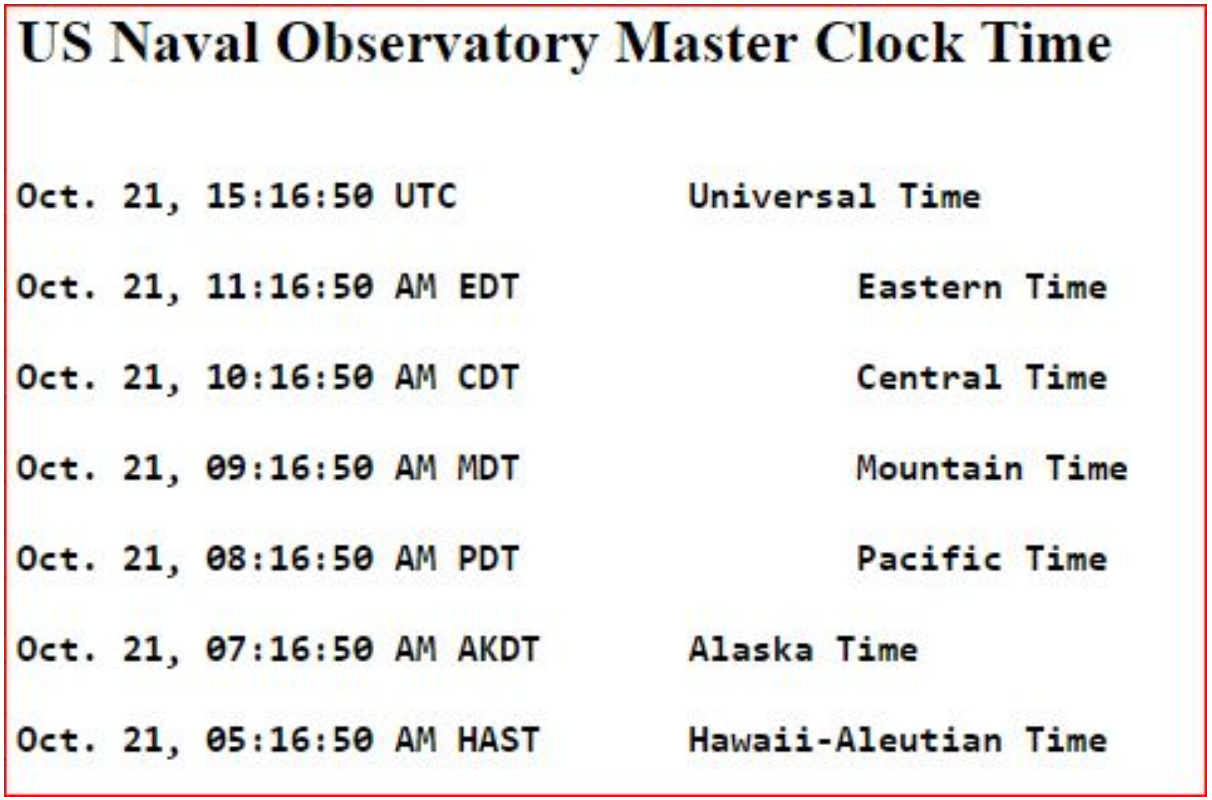
In [30]:
```
load_file = open('py-pickle.dat', 'rb')
loaded_game_data = pickle.load(load_file)
load_file.close()

print(loaded_game_data)
```

```
{'player-position': 'N23 E45', 'pockets': ['keys', 'pocket knife', 'polished stone'], 'backpack': ['rope', 'ham
mer', 'apple'], 'money': 158.5}
```

Read the web

what time is it? (http://tycho.usno.navy.mil/cgi-bin/timer.pl)



In [31]:
```
from urllib.request import urlopen
with urlopen('http://tycho.usno.navy.mil/cgi-bin/timer.pl') as response:
    for line in response:
        line = line.decode('utf-8')  # Decoding the binary data to text.
        if 'EST' in line or 'EDT' in line:  # look for Eastern Time
            line=line.replace('<BR>','')
            print(line)
```

```
Oct. 21, 06:24:29 PM EDT                Eastern Time
```

In [32]:
```
fd = open('dream_poem.html','wb')
with urlopen('https://100.best-poems.net/dream-within-dream.html') as response:
    for line in response:
        fd.write(line)
fd.close()
```

Statistics

In [33]:
```
import statistics as stat
data = [2.75, 1.75, 1.25, 0.25, 0.5, 1.25, 3.5]
min(data), max(data), stat.mean(data), stat.mode(data), stat.stdev(data), stat.median(data), stat.variance(data)
```

Out[33]:
```
(0.25,
 3.5,
 1.6071428571428572,
 1.25,
 1.171334200612195,
 1.25,
 1.3720238095238095)
```

In [34]:
```
dir(stat)
```

Out[34]:
```
['Decimal',
 'Fraction',
 'StatisticsError',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 '_coerce',
 '_convert',
 '_counts',
 '_decimal_to_ratio',
 '_exact_ratio',
 '_isfinite',
 '_ss',
 '_sum',
 'collections',
 'groupby',
 'math',
 'mean',
 'median',
 'median_grouped',
 'median_high',
 'median_low',
 'mode',
 'pstdev',
 'pvariance',
 'stdev',
 'variance']
```

Brief Tour of the Standard Library (https://docs.python.org/3/tutorial/stdlib.html)

- 10.2. File Wildcards
- 10.3. Command Line Arguments
- 10.4. Error Output Redirection and Program Termination
- 10.5. String Pattern Matching
- 10.6. Mathematics
- 10.7. Internet Access
- 10.8. Dates and Times
- 10.9. Data Compression
- 10.10. Performance Measurement
- 10.11. Quality Control
- 10.12. Batteries Included

Python Quick Ref (http://www.cs.put.poznan.pl/csobaniec/software/python/py-qrc.html)

In [ ]: