

String Methods

capitalize()	Capitalizes first letter of string
center-(width)	String is centered within X characters
endswith(sub)	True if string ends with substring
expandtabs(width)	Turns \t into X blank spaces
find(sub, start, end)	Can just put in string as variable to find loc
index(sub start end)	Finds the index of a list or string
isalnum()	True if only letter or number (no spaces)
isalpha()	True if it's a letter
isdigit()	True if a number
islower()	True if all lowercase
isspace()	True if a space exists in the string
istitle()	True if first letter of each word is capitalized
isupper()	True if all of string is upper case
join()	':'.join(list) would combine all with ':'
replace(old, new)	Replaces all instances of old with new
startswith(sub)	True if string starts with substring
rpartition(sub)	Turns string into 3 part tuple based on substr

Threading w/Threading Module

```
import threading

t = threading.Thread(
    target=func, args=(arg1,))

t.start() Start the thread right after threading.Thread
```

Threading w/Threading Module (cont)

threads.append(t)	Appends thread to your defined list for sync
for threads in threads: threads.join()	Joins threads so they end at same time

Datetime Module

```
import datetime

datetime.today().weekday() Returns 0 thru 6. 0 is Monday

datetime.today().isoweekday() Returns 0 thru 6. 1 is Monday

datetime.datetime.today().strftime("%A") Returns name for day of the week

datetime.datetime.now().strftime("%H:%M:%S") Returns hour minute and second of current time

datetime.datetime.now() Datetime object of current time
```

Find time between two datetime objects

```
import datetime

first_time = datetime.datetime.now()

later_time = datetime.datetime.now()

difference = later_time - first_time

seconds_in_day = 24 * 60 * 60

divmod(difference.days * seconds_in_day + difference.seconds, 60)

Returns a tuple of minutes comma seconds
ex. (8, 0)
```

sys.argv for python foo.py bar -c qux -h

sys.argv[0]	foo.py
sys.argv[1]	bar
sys.argv[2]	-c
sys.argv[3]	qux
sys.argv[4]	--h

os Module

os.path.isdir(loc)	True if loc is a directory
os.path.isfile(loc)	True if loc is a file
os.path.normpath(loc)	Gives path normalized for Win *

* Removes redundant separators and up-level references

Django Module

pip install Django	External module
django-admin startproject mysite	Starts a new project folder called mysite
python manage.py runserver [ip][:port]	Runs the project stuff in brackets is optional
python manage.py startapp blog	Creates application in your project folder called blog
python manage.py makemigrations blog	Propagates blog app changes to database
python manage.py migrate	Syncs database with new model
python manage.py createsuperuser	Creates a superuser, prompts will follow