SI 206 Discussion 3
Classes, Git

SI 206 00x FA 2021 > Modules > Discussion 3 - Assignment

Today

1. Object oriented programing
    a. Create a Dice Class and Methods to roll the Dice, Store the value, etc.
    b. Create Dice instances, and call the methods
2. Git : Commit code after each method and push to GitHub in the end

Basic Terminal / Command Prompt Commands

Will make your life easier….

1. up arrow: Bash history search
2. Tab: Autocomplete the filename. Super helpful with long filenames!

(quick demo)

Delete Pass!!

```
# import turtle module
from turtle import *

def draw_rectangle(turtle, xpos, ypos, width, height, color):
    """
    Write a function to draw a rectangle on the screen
    with the specified parameters.
    """
    pass

def draw_triangle(turtle, xpos, ypos, length, color):
    """
    Write a function to draw a triangle on the screen
    with the specified parameters.
    """
    pass
```

Once you write something in a function / class,
don't forget to delete "pass"

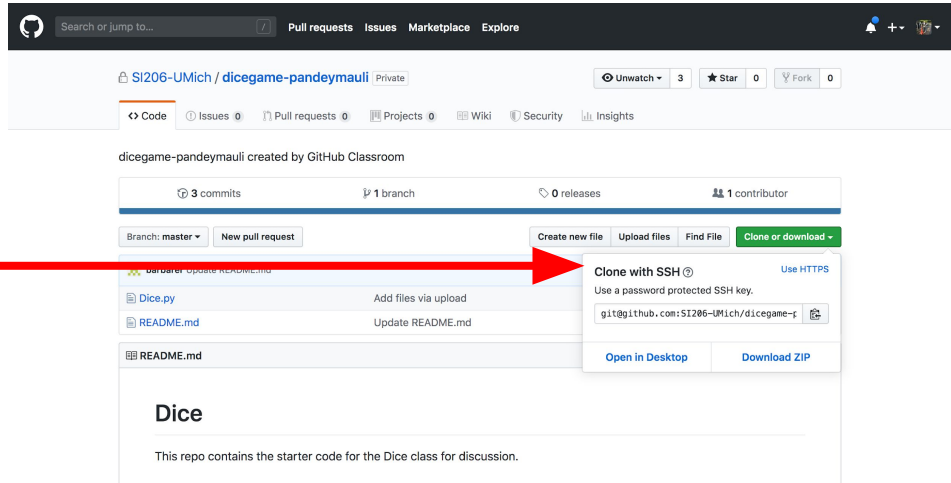
Git

Starter code is here : See Modules > Discussion3 - Assignment

You may have to join the github classroom for SI206. Please do so.

Clone the discussion 3 repository using this command:
git clone <repository_url>

Copy repository_url from here



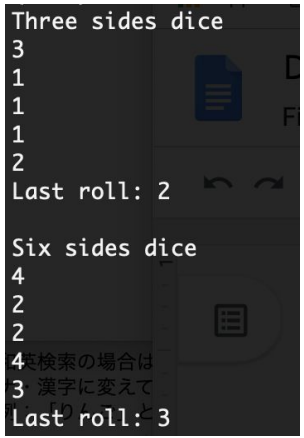
Problems 1 and 2 (Instructions also in starter code)

A dice is an example of an object we can simulate with a Python class. Actually, die is singular and dice is plural, but we will just call one die a dice in this assignment. Our *Dice* class will have a constructor (__init__ method) along with additional methods to track our rolls and get data on them. For this exercise, you will be making a *Dice* class with the following:

- **Constructor (__init__) method**: the constructor will initialize a new dice object that has not yet been rolled. When you create the *Dice* object you will set how many sides the dice has. By default, it will have 6 sides. In the constructor, initialize instance variables for:
  - The number of sides
  - A list keeping track of all of th `Last roll: 6` s rolled
- ***__str__*** method: create a string method so that printing an instance of the dice class outputs the value of the last roll. For example:

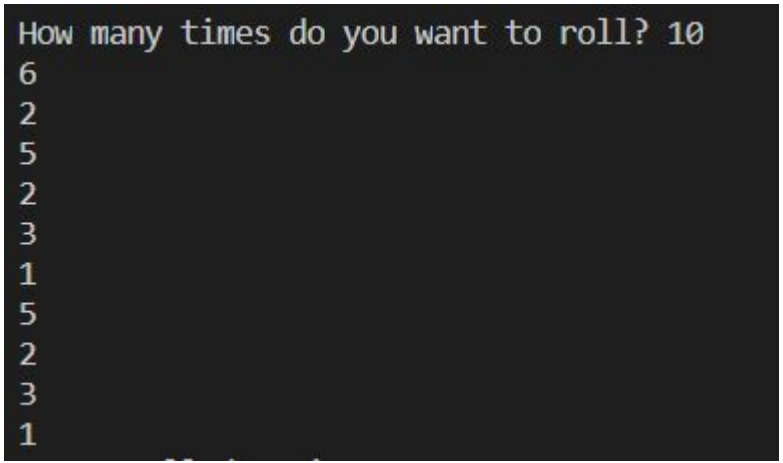
Problems 3 (Instructions also in starter code)

- ***roll*** method: Rolls the dice to get a random integer between 1 and the number of sides (hint: use the random module and include both 1 and the number of sides). Save the value at the end of a list that tracks all the values rolled. Returns the number rolled. For example :



Problems 4 (Instructions also in starter code)

- ***num_rolls*** method: Takes in user input to quantify the amount of rolls. It asks the users, “How many times do you want to roll?” It the prints out each roll. (hint: use the the input function to handle user input). For example :



Bonus Method (Instructions also in starter code)

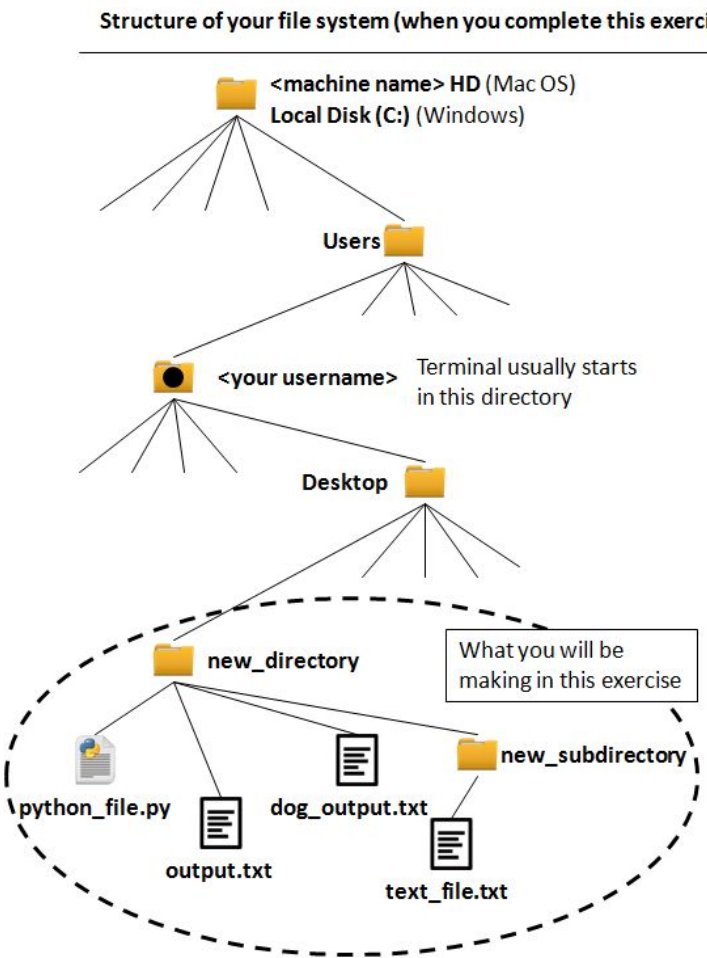
print_count_for_num method: Takes in a parameter num which specifies which roll value to look for. Loop through the roll history list and count how many times that num was rolled. Print the number of times that value was rolled, for example :

```
2 was rolled 4 times
```

Appendix

References for today

Review: Absolute path and relative path



1. Absolute path

   Path from root directory
   `/Users/[Username]/Desktop/new_directory/output.txt`

   Path from home directory
   `~/Desktop/new_directory/output.txt`

1. Relative path

   If you are in "new_directory"
   `output.txt`

   If you are in "Desktop"
   `new_directory/output.txt`

   If you are in "new_subdirectory"
   `../output.txt`

W10. Class and Instance

class vehicle():
body color
tire
type
owner

Class

A = vehicle(...):
body color: Red
tire: 4
type: Sedan
owner: Mary

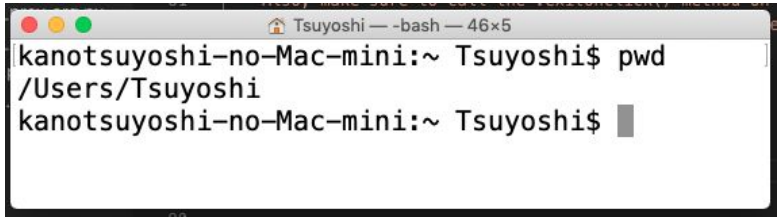
B = vehicle(...)
body color: Blue
tire: 2
type: bicycle
owner: Tom

C = vehicle(...)
body color: Green
tire: 2
type: Scooter
owner: John

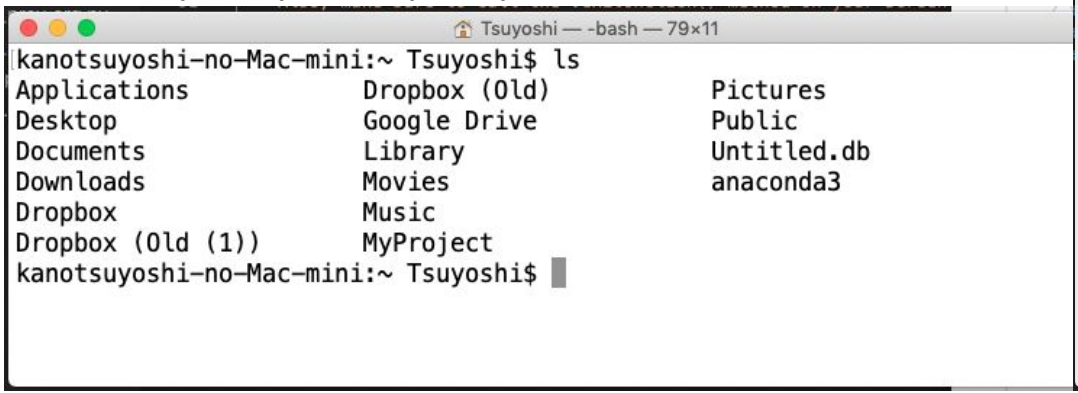
Instance

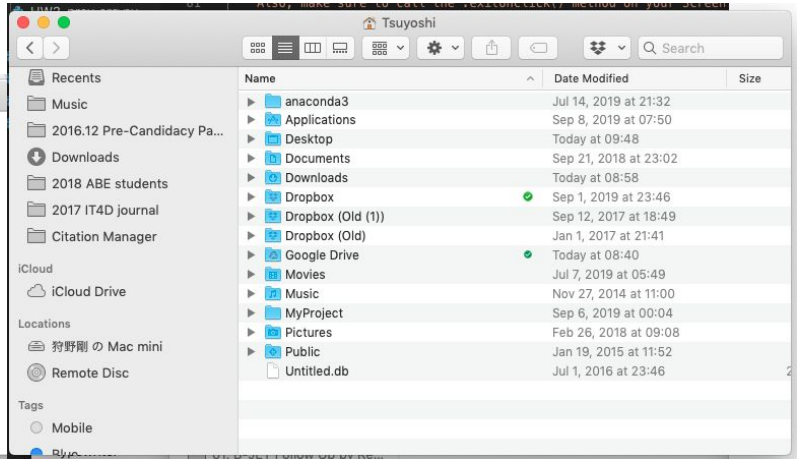
Appendix: Basic Linux / MS-DOS commands 1

1. pwd (Mac) / chdir (Win) : Display your location in the file system



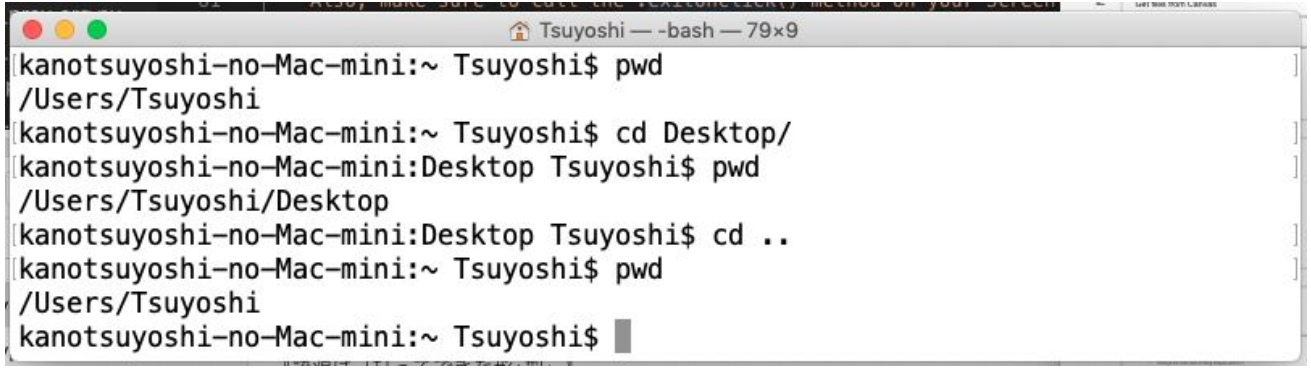
1. ls (Mac) / dir (Win) : Lists files





Appendix: Basic Linux / MS-DOS commands 2

1. cd *pathname*: Changes directories



1. python *filename.py* / python3 *finename.py* : Run a python file

Command's Purpose	MS-DOS	Linux	Basic Linux Example
Copies files	`copy`	`cp`	`cp thisfile.txt /home/thisdirectory`
Moves files	`move`	`mv`	`mv thisfile.txt /home/thisdirectory`
Lists files	`dir`	`ls`	`ls`
Clears screen	`cls`	`clear`	`clear`
Closes shell prompt	`exit`	`exit`	`exit`
Displays or sets date	`date`	`date`	`date`
Deletes files	`del`	`rm`	`rm thisfile.txt`
"Echoes" output to the screen	`echo`	`echo`	`echo this message`
Edits text files	`edit`	`gedit`([a])	`gedit thisfile.txt`
Compares the contents of files	`fc`	`diff`	`diff file1 file2`
Finds a string of text in a file	`find`	`grep`	`grep word or phrase thisfile.txt`

Command's Purpose	MS-DOS	Linux	Basic Linux Example
Formats a diskette	`format a:` (if diskette is in `A:`)	`mke2fs`	`/sbin/mke2fs /dev/fd0` (`/dev/fd0` is the Linux equivalent of `A:`)
Displays command help	*command* `/?`	`man` or `info`	`man` *command*
Creates a directory	`mkdir`	`mkdir`	`mkdir` *directory*
Views contents of a file	`more`	`less`([b])	`less` *thisfile.txt*
Renames a file	`ren`	`mv`([c])	`mv` *thisfile.txt thatfile.txt*
Displays your location in the file system	`chdir`	`pwd`	`pwd`
Changes directories with a specified path (*absolute path*)	`cd` *pathname*	`cd` *pathname*	`cd` */directory/directory*
Changes directories with a *relative path*	`cd..`	`cd ..`	`cd ..`
Displays the time	`time`	`date`	`date`
Shows amount of RAM in use	`mem`	`free`	`free`