华夏中文学校 Python level-I

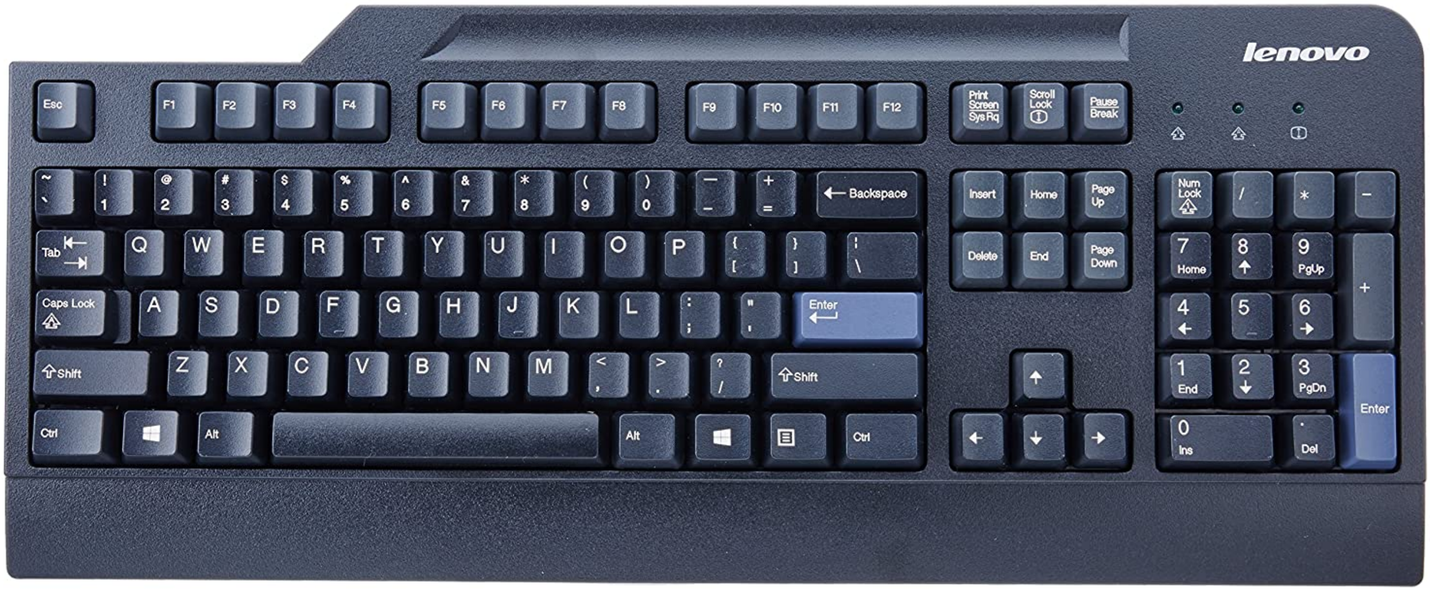
开课前要把所有用到的程序运行一遍

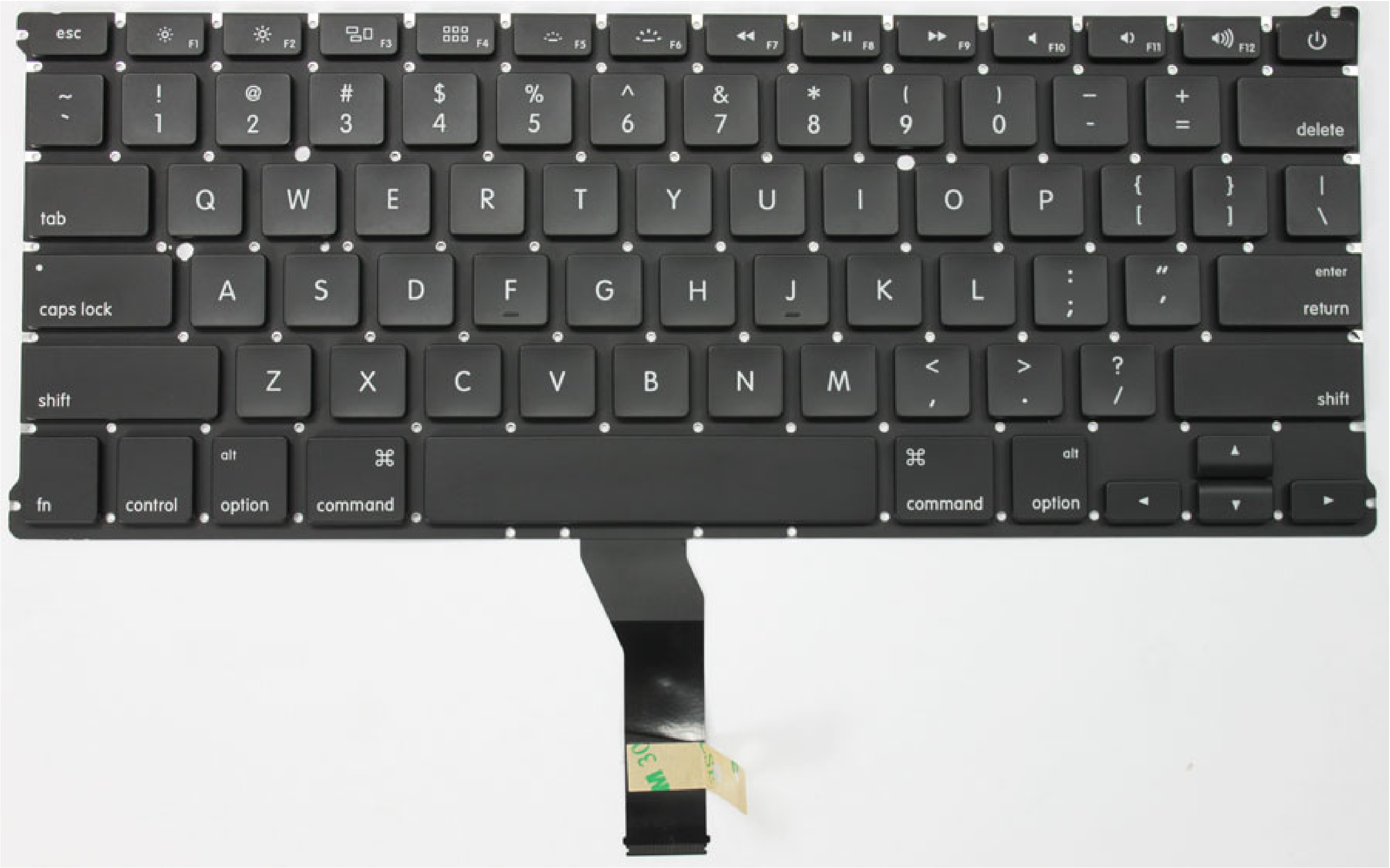
Table of Contents

√

Table of Contents

Familiar with your keyboard





Share Keyboard document

1. Key name

Key	Name
space	space, empty space in editor
Enter	return, enter, new line in editor
:	colon, key:value separator in dict
,	comma, list or tuple item separator, delimiter in csv file
.	dot, period, instance function call()
#	pound, hashtag, number, hold shift key click number 3, comments the line
`	back quote, grave accent, command block in markdown
*	asterisk, star, bullet point in markdown, math multiply operator
()	parenthesis, tuple, function definition and call
-	dash, hyphen, minus math operator, command option python --version
_	underscore, dunder function or variable, private or protected variables
{}	curly bracket, dict or set
[]	bracket, square bracket, list
\	back slash, line continue, escape sequence
/	forward slash, file name path fold dilimiter
|	pipe, virtical bar, bitwise OR operator
&	ampersand, and simple, bitwise AND operator
^	caret, circumflex, bitwise XOR operator
?	question mark, space holder in sqlit
$	dollar sign
;	semicolon

- combination keys

```
ctrl+c
Ctrl+v
ctrl+/
shift+downarrow
tab
shift+tab
```

- Command line arrow key usage

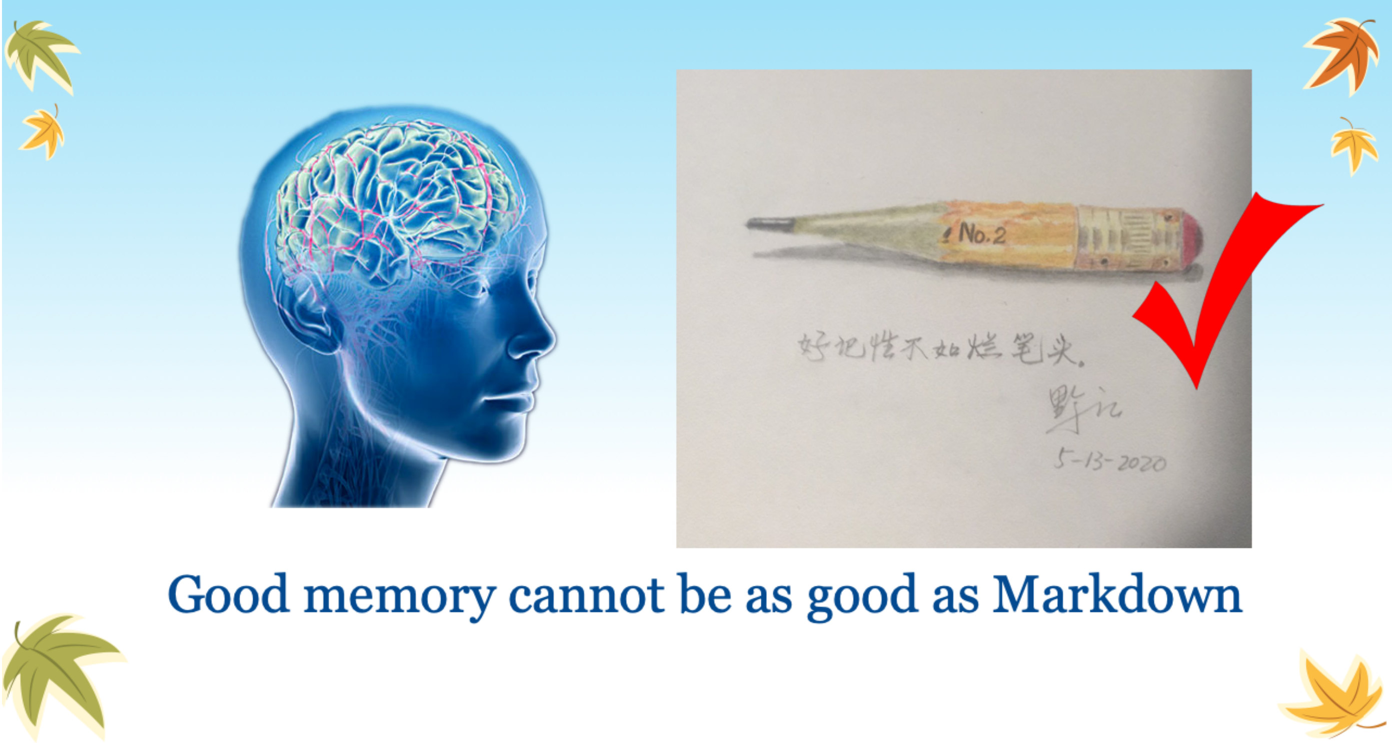
```
upArrow: bring previous command back
downArrow: bring next command back
leftArrow: move cursor to left in DOS window
rightArrow: move cursor to right in DOS window
```

- Hight light block of code
- Ctrl+c: copy
- Ctrl+v: paste
- Ctrl+/: toggle comments

Table of Contents

Using Markdown

- turn in homework to GitHub
- VS Code Extension
  - Markdown All in One
  - Markdown Preview Enhanced
  - Unicode LaTex
- √ Markdown md文件的制作，制作课堂笔记



  - add Markdown Extension
  - 显示标题，子标题 #, ##
  - 显示 bullet point *， 1
  - 显示命令行
  - 显示图形
  - 显示链接

- Markdown Cheat Sheet
- Reference to pythonInstall.md
- Install Greenshot

installation file name: Greenshot-INSTALLER-1.2.10.6-RELEASE.exe

- Basic operation

Table of Contents

Getting start

- install softwares needed

refer to python installation file.

- check installation
- installation check

```
python --version
git --version
code --version
```

- build working folders

```
mkdir workspace
cd workspace
mkdir python1
```

use text editor: NotePad.exe

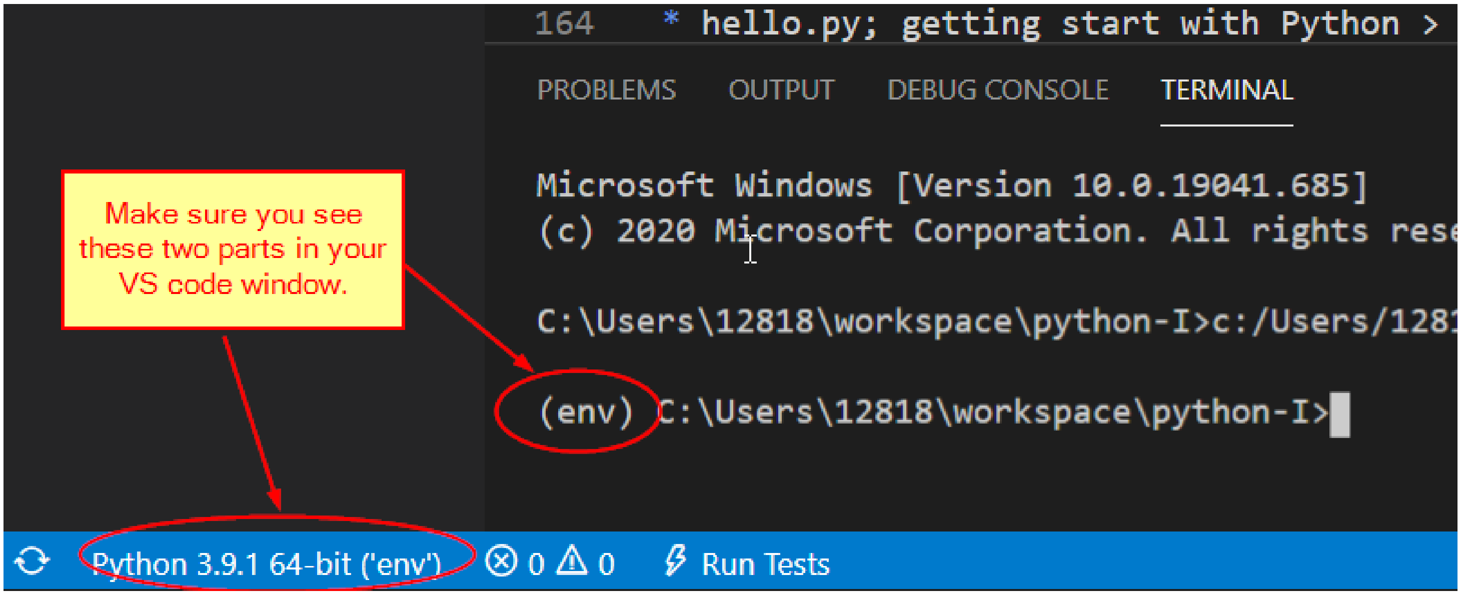
```
print("Hello, world!")
a = 4
b = 5
print(a+b)
```

save to first.py

```
python first.py
```

- build virtual environment

```
python -m venv env
```



```
* hello.py; getting start with Python >
```

Make sure you see these two parts in your VS code window.

```
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. All rights rese

C:\Users\12818\workspace\python-I>c:/Users/1281

(env) C:\Users\12818\workspace\python-I>
```

Python 3.9.1 64-bit ('env')

- familiar with VSCode.
  VS code

Table of Contents

print

- hello.py; getting start with Python > hello.ReadMe.md
- print.py; hello/print.py
- helloHim.py; intruduce input() function
- print-string.py;
- input.py
- guessNumber.py
- dice.py; introduce random module, dice/dice1.py
- dice2.py; figure out possibility, understand how computer do things

Table of Contents

Dice

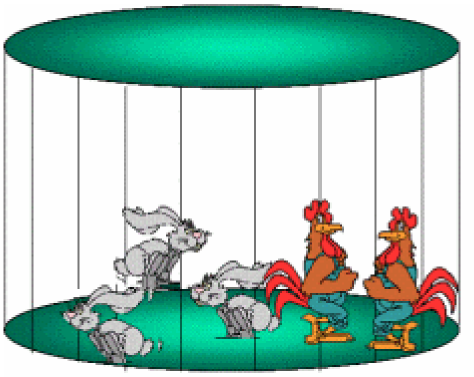
- dice.py; introduce random module, dice/dice1.py
- dice2.py; figure out possibility, understand how computer do things

Table of Contents

Simple math

./mymath

- math0.py
- math1.py
- math2.py
- math10.py
- solution.py
- circle.py
- linear1.py



- linear2.py
- linear3.py
- linear4.py
- linear5.py

$$area = r^2\pi$$

- perfactNumber1.py
- Volumn of Sphere

$$V = \frac{4}{3}\pi r^3$$

- Volumn of Cylinder

$$V = r^2\pi \cdot h$$

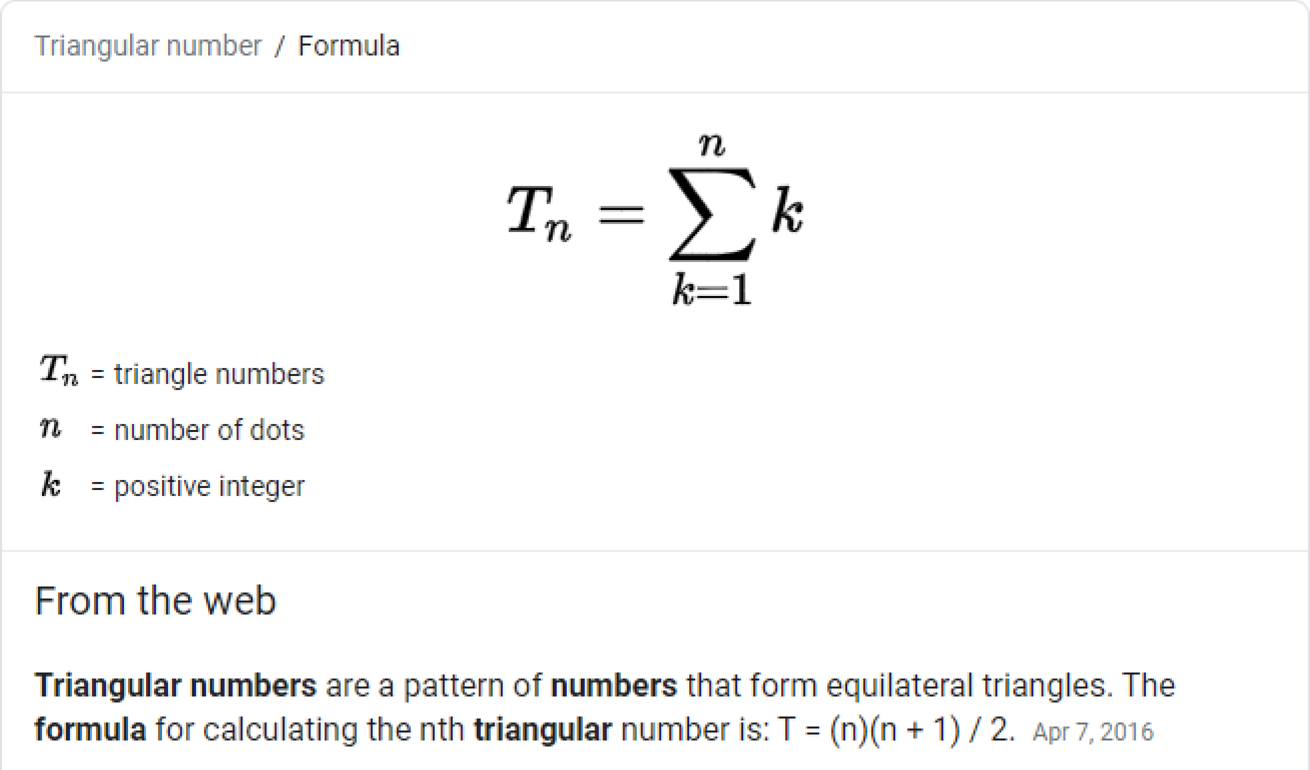
- Triangle area

$$area = \frac{1}{2}(b \cdot h)$$

- Triangular Number

$$T_n = \sum_{k=1}^{n} k$$

$T(n) = \frac{n(n+1)}{2}$



Triangular number / Formula

$$T_n = \sum_{k=1}^{n} k$$

T_n = triangle numbers

n = number of dots

k = positive integer

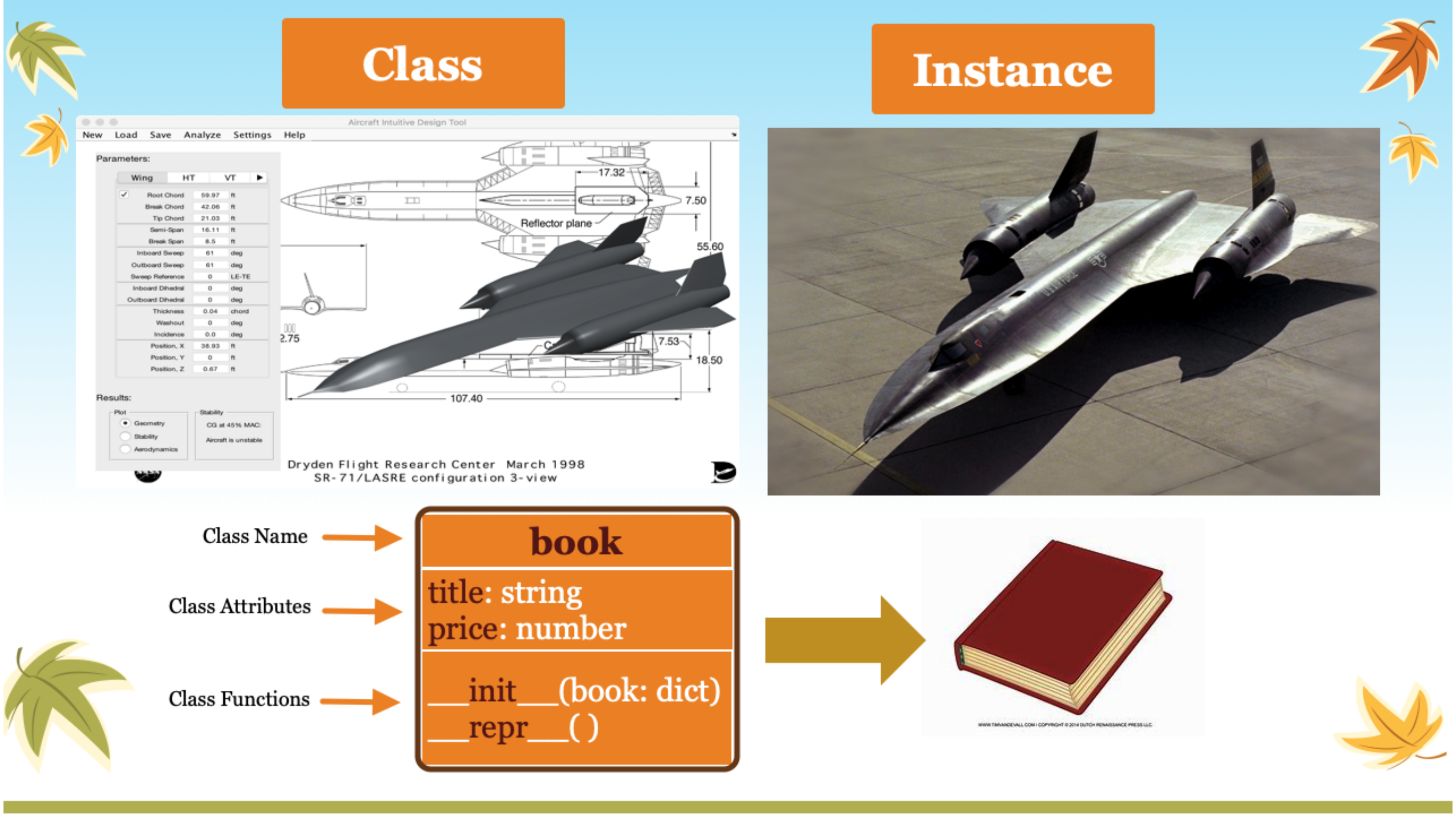
From the web

Triangular numbers are a pattern of **numbers** that form equilateral triangles. The **formula** for calculating the nth **triangular** number is: T = (n)(n + 1) / 2. Apr 7, 2016

- solution1.py
- circle.py
- prime1.py; ./prime/prime1.py
- prime1.py ~ prim7.py; treat computer as humanbeen, do it right

Table of Contents

OOP



Class

Instance

Aircraft Intuitive Design Tool

New Load Save Analyze Settings Help

Parameters:

Wing | HT | VT | ▶

☑	Root Chord	59.97	ft
	Break Chord	42.06	ft
	Tip Chord	21.03	ft
	Semi-Span	16.11	ft
	Break Span	8.5	ft
	Inboard Sweep	61	deg
	Outboard Sweep	61	deg
	Sweep Reference	0	LE-TE
	Inboard Dihedral	0	deg
	Outboard Dihedral	0	deg
	Thickness	0.04	chord
	Washout	0	deg
	Incidence	0.0	deg
	Position, X	38.93	ft
	Position, Y	0	ft
	Position, Z	0.67	ft

Results:

Plot

● Geometry

○ Stability

○ Aerodynamics

Stability

CG at 45% MAC:

Aircraft is unstable

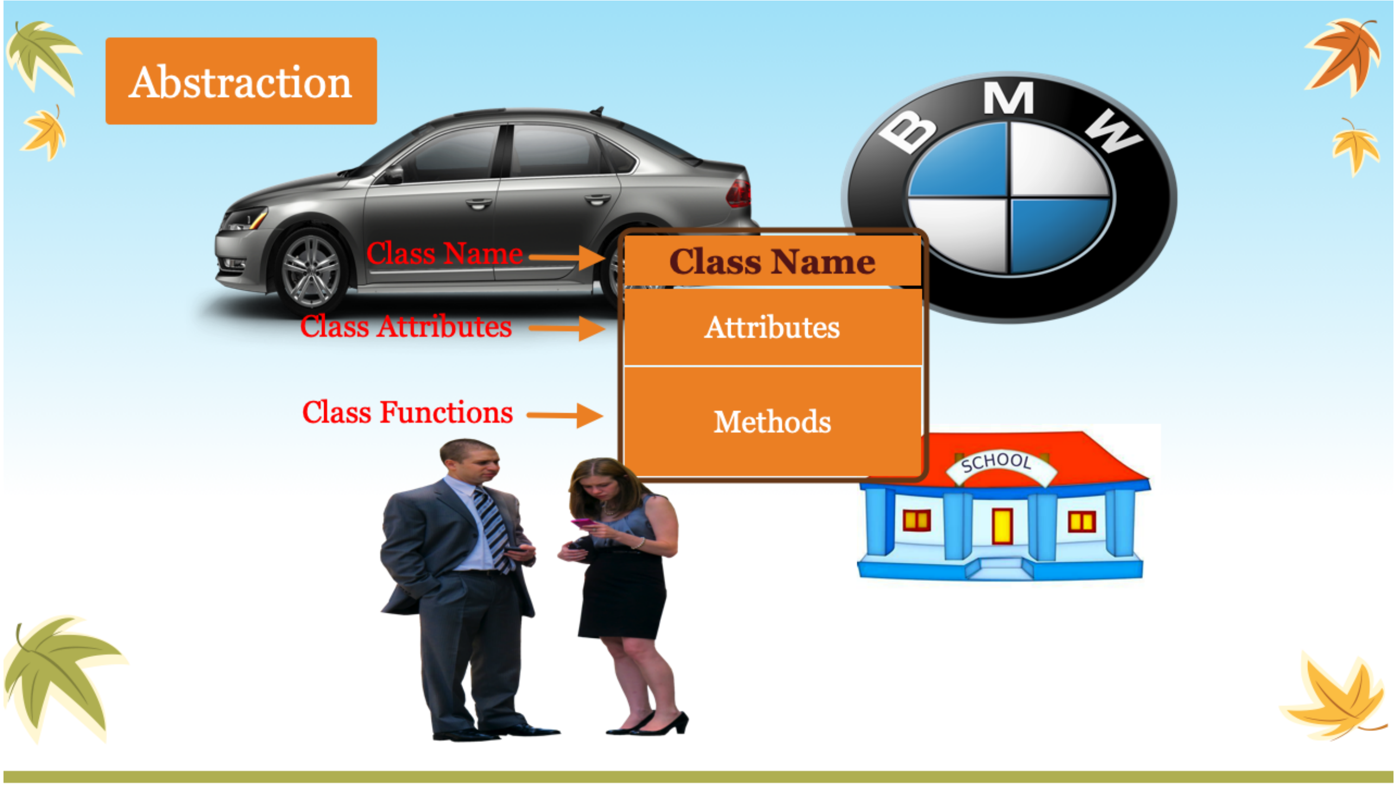
17.32
7.50
Reflector plane
55.60
2.75
7.53
18.50
107.40
Dryden Flight Research Center March 1998
SR-71/LASRE configuration 3-view

Class Name → book

Class Attributes → title: string
price: number

Class Functions → __init__(book: dict)
__repr__( )

- class book, **init**, **repr**
- class student.py constructor, **repr** abstraction



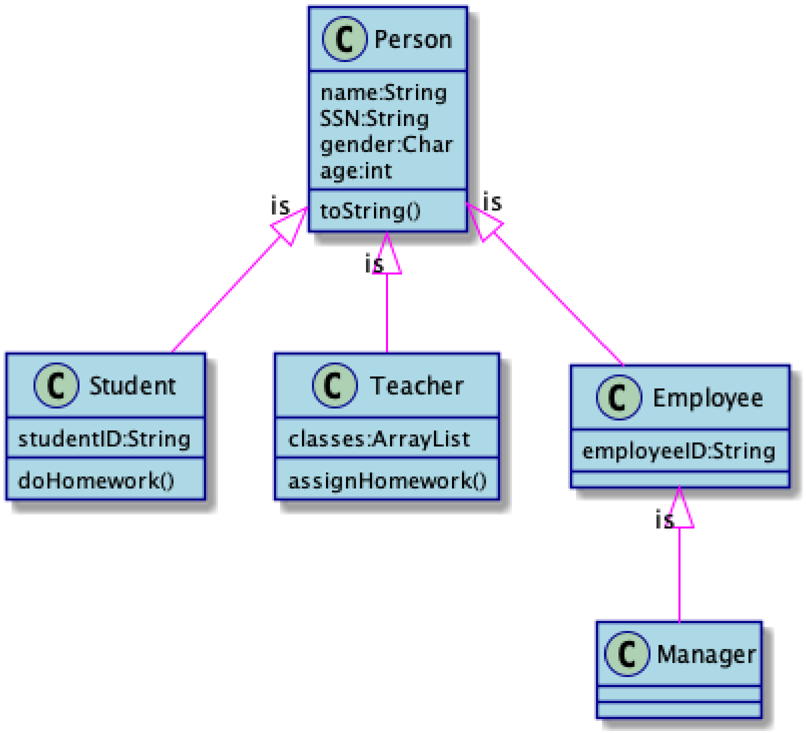
Abstraction

Class Name → Class Name

Class Attributes → Attributes

Class Functions → Methods

- user.py, User, SubUser inheritence testUser
- person, teacher, student inheritence



YouTube Classes

Python Classes

Table of Contents

Python class

```
class User:
    pass
```

- assign fields to an instance of User

Python Classes

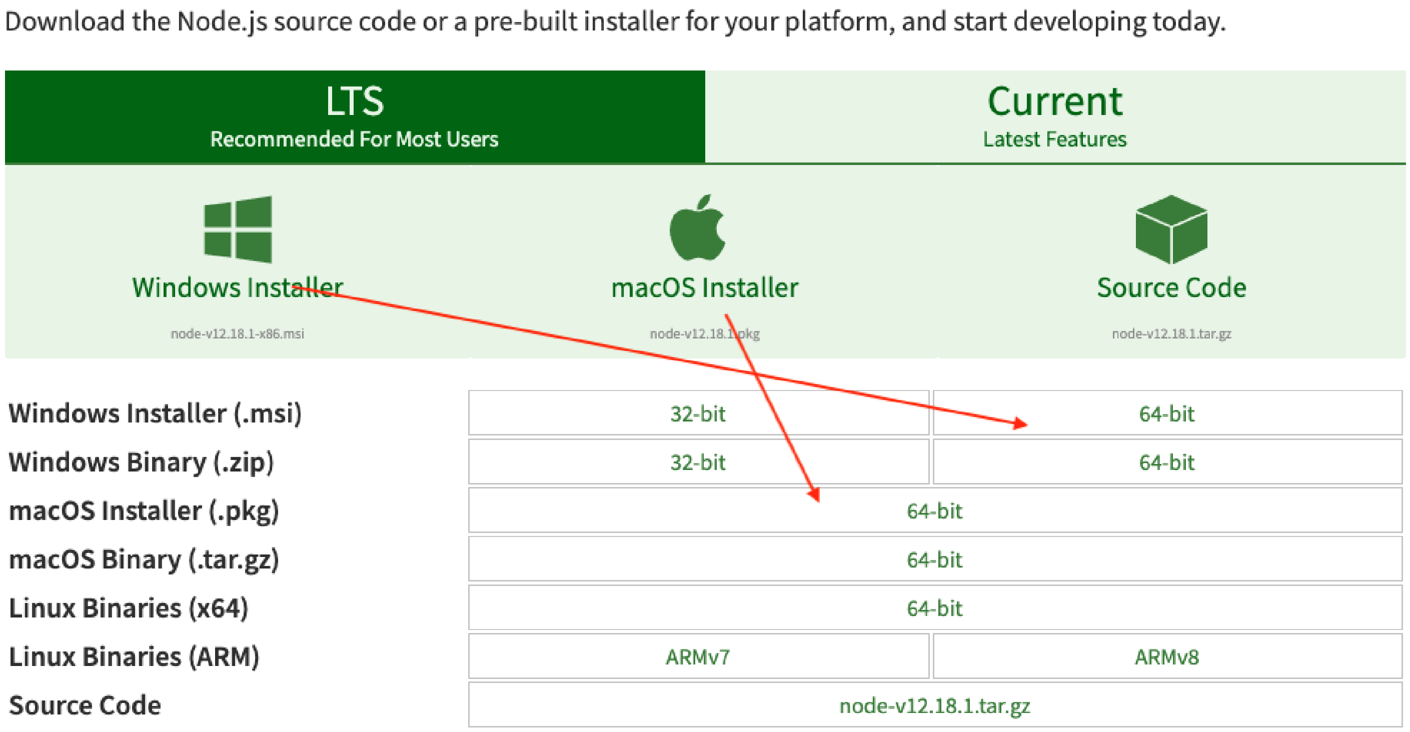
- person.py
- bookdb.py
- create a class snowman.py > drawSnowMan.py > shapes.py
- class0.py - pass class, instance and class level attributes
- create a class snowman.py > drawSnowMan.py > shapes.py
- class1.py > dynamically assign instance attribute and access it from outside function
- √ class2.py > define internal function

- √ class3.py > **init**(self) and internal function
- √ class4.py > use keyword argument in **init**(self)
- √ class5.py > understand **str**, **repr**, and **len**()
- √ class6.py > protected attribute and private attribute
- √ class7.py > getter/setter
- √ personInheritance.py > inheritance
- personTest.py > understand class name <module_name>.<class_name>
- √ bookdb.py > used in app4.py
- polygon.py; ask student implement **repr**(self)
- student.py; using class level method
-

Table of Contents

install npm

Download and install node



Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS
Recommended For Most Users

Current
Latest Features

Windows Installer
node-v12.18.1-x86.msi

macOS Installer
node-v12.18.1.pkg

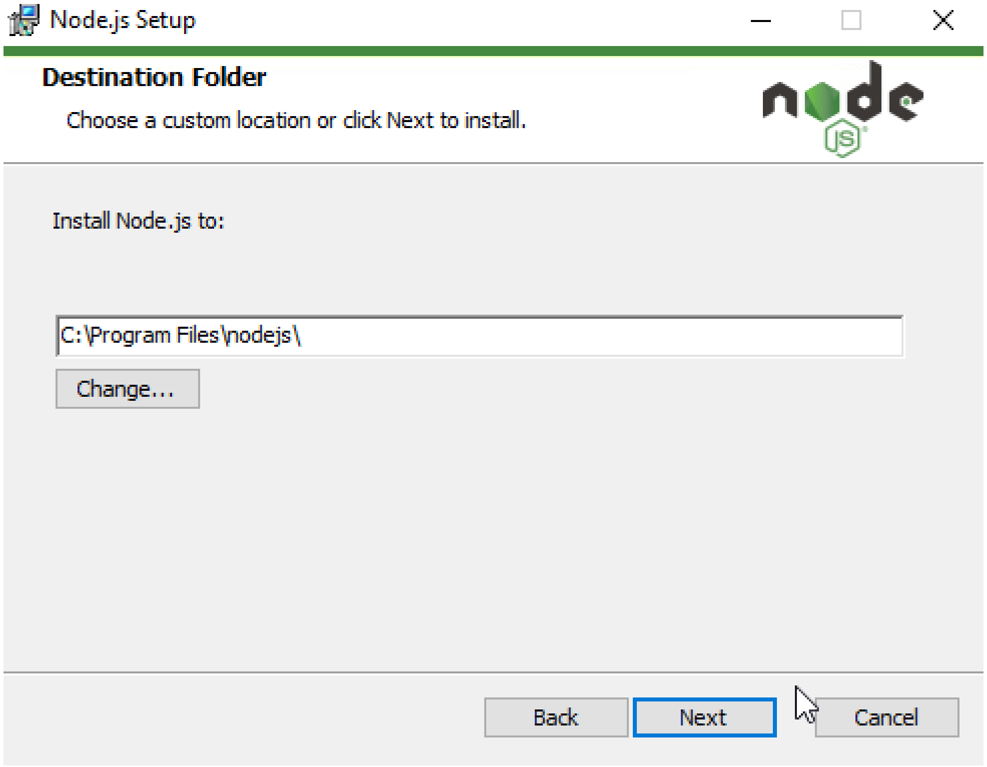
Source Code
node-v12.18.1.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.18.1.tar.gz	

- windows File: node-v12.18.3-x64.msi
- macos File:

- √ Install NodeJS & npm on windows 10

  nodejs.org/en/

  Node.js Setup

  Destination Folder

  Choose a custom location or click Next to install.

  Install Node.js to:

  C:\Program Files\nodejs\

  Google Search: install reactjs on windows 10

  Step by step option 2

```
node --version
npm --version
```

- create react js application

```
npm install -g create-react-app

create-react-app --version

create-react-app reactproject2
```

- Install ReactJs on MacOS

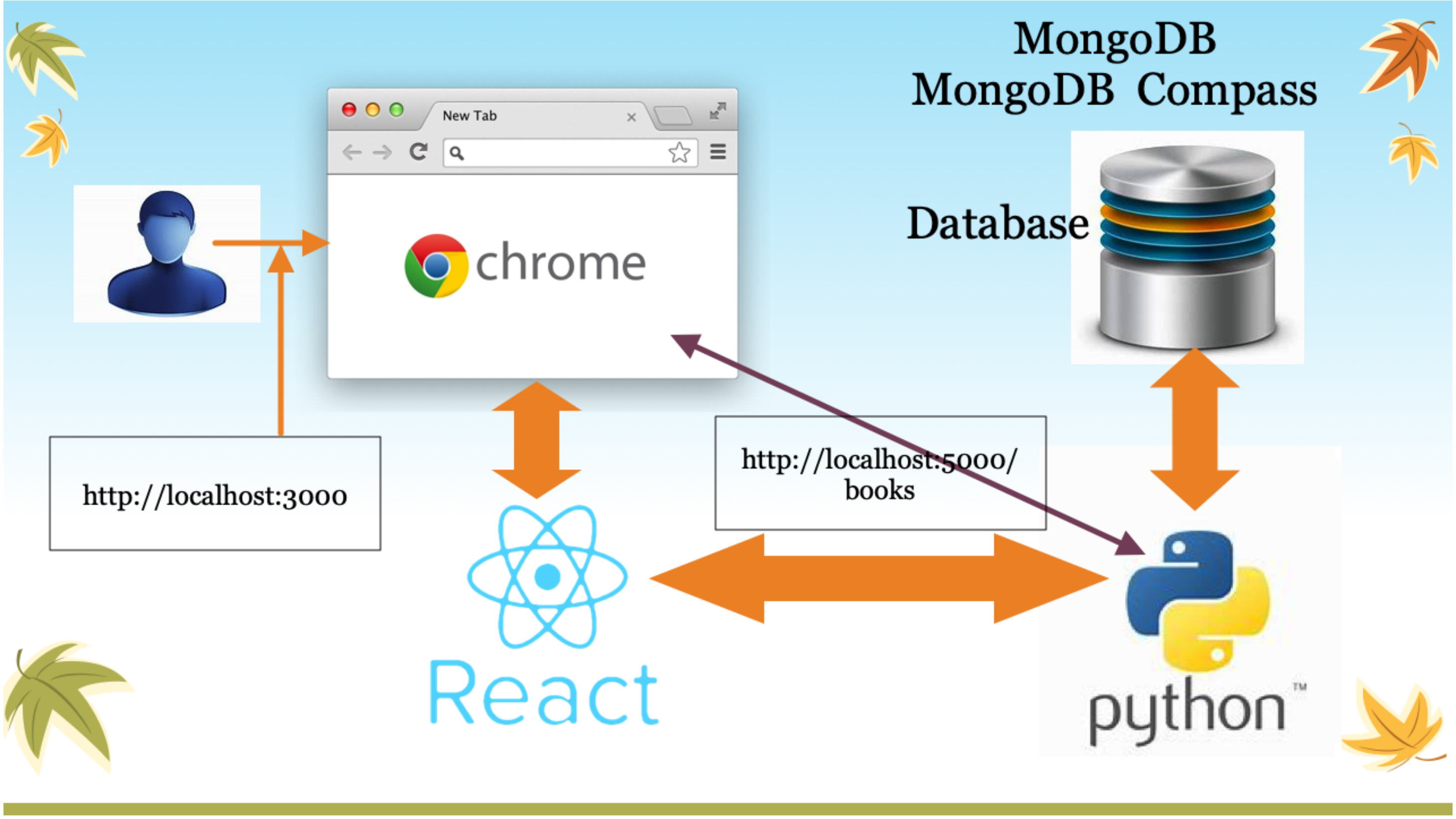
```
sudo npx create-react-app wang-app
sudo chown -R wangqianjiang wang-app
cd wang-app
npm start
```

Table of Contents

ReactJS

- web application vs. window application
  open new VSCode window > python-gui (demo on window's machine.)

```
python calculator2.py
```



- get reactjs project from github

```
git clone https://github.com/jwang1122/reactjs.git
```

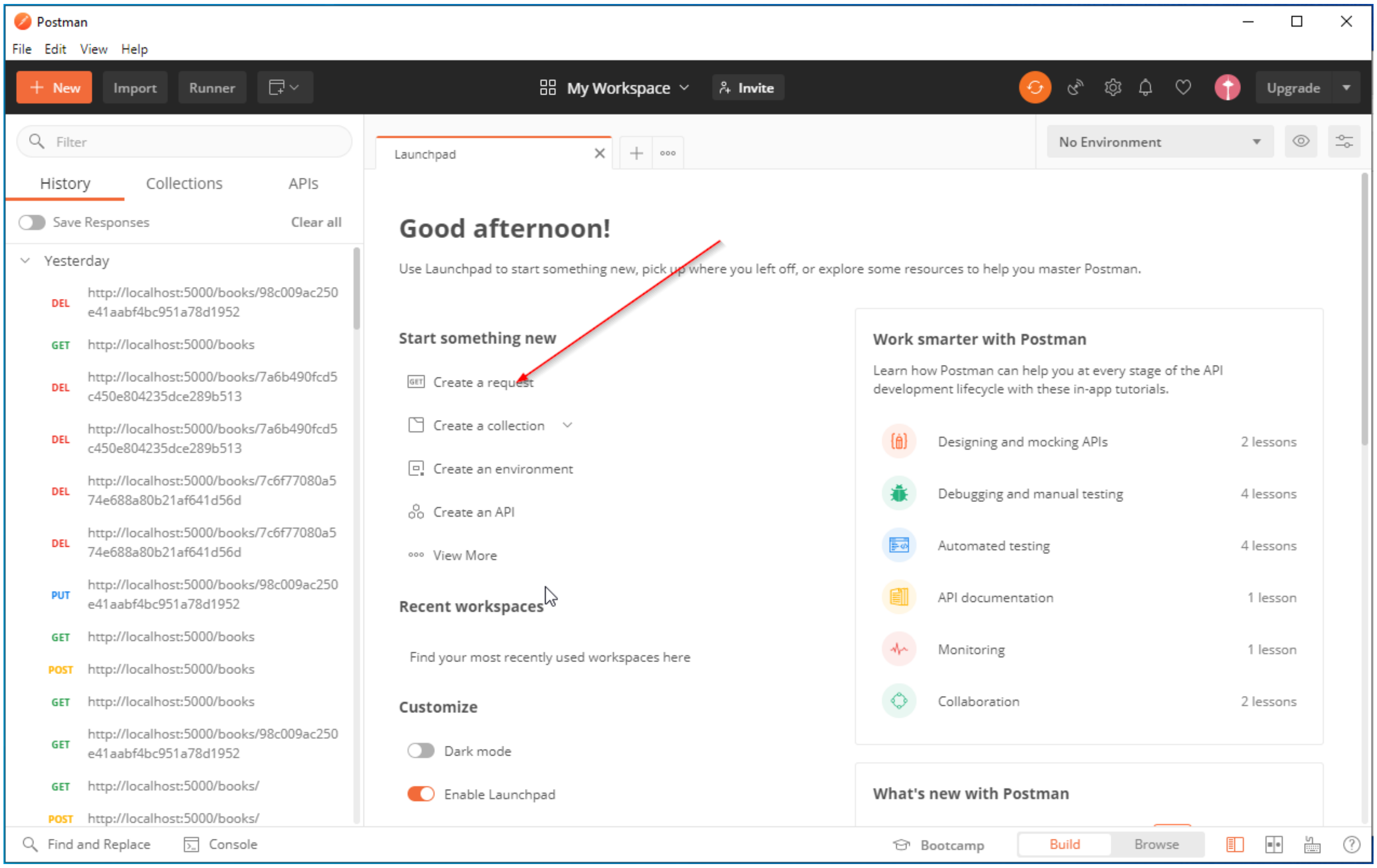
- start the application
  open new VSCode > ~/workspace/reactjs

```
cd server
python app.py
cd ../book-app
npm start
```

App server

- URL: Uniform Resource Locator
  - https://www.google.com
  - Protocal: http, https, ftp ...
  - Host: www.google.com
  - Port: number followed by :, default 80 for http, 443 for https
  - Path:
  - Querystring: text after ?, key=value pair separated by &
  - Fragment: text after #(hashtag), jump to certain section in the document
- app1.py > ping-pong
- app2.py >
- app3.py > display hardcoded books
- app4.py > display books from mongodb, postman > test service
- getJson.py > load books from given website url
- bookdb.py
- Install Postman

  Download Website



DEL http://localhost:5000/books/98c009ac250e41aabf4bc951a78d1952
GET http://localhost:5000/books
DEL http://localhost:5000/books/7a6b490fcd5c450e804235dce289b513
DEL http://localhost:5000/books/7a6b490fcd5c450e804235dce289b513
DEL http://localhost:5000/books/7c6f77080a574e688a80b21af641d56d
DEL http://localhost:5000/books/7c6f77080a574e688a80b21af641d56d
PUT http://localhost:5000/books/98c009ac250e41aabf4bc951a78d1952
GET http://localhost:5000/books
POST http://localhost:5000/books
GET http://localhost:5000/books
GET http://localhost:5000/books/98c009ac250e41aabf4bc951a78d1952
GET http://localhost:5000/books/
POST http://localhost:5000/books/

Good afternoon!

Use Launchpad to start something new, pick up where you left off, or explore some resources to help you master Postman.

Learn how Postman can help you at every stage of the API development lifecycle with these in-app tutorials.

Find your most recently used workspaces here

- start app4.py, test POST, UPDATE, DELETE methods

Table of Contents

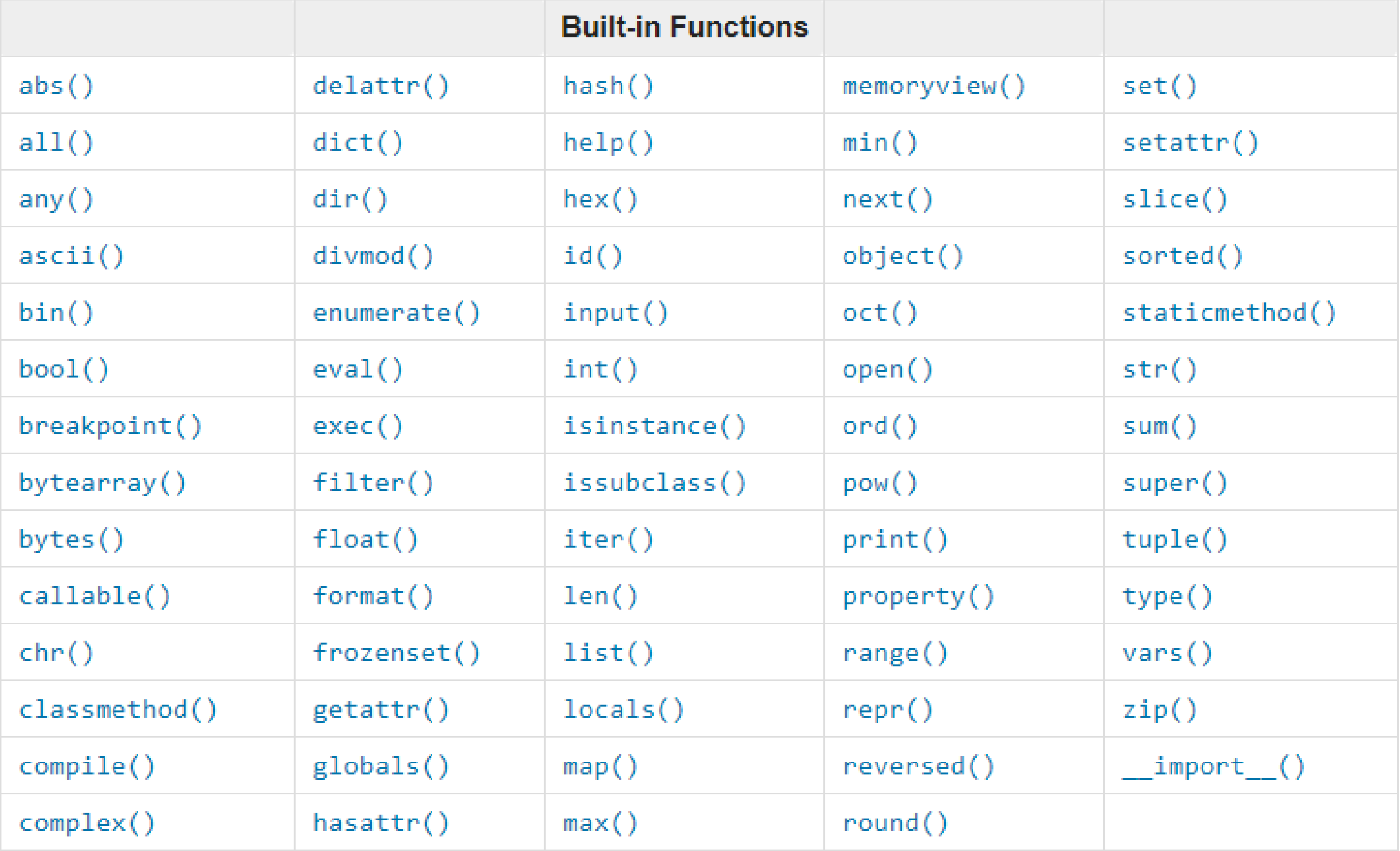
Mongo DB

Install MongoDB

- NoSQL - MongoDB ->
  - collection
- SQL: Structured Query Language
  What is SQL?
- create0.py > create book and save it to mongodb
- create1.py > create more than one document at once
- retrieve0.py > retrieve one book from mongodb
- retrieve1.py > retrieve all books from mongodb
- retrieve2.py > retrieve some books based on condition from mongodb
- update.py > update one document
- delete.py > delete one document
- bookdb.py > create a class include all CRUD process.

Table of Contents

Function



		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

- define a function

$$\underbrace{def}_{keyword}\ \underbrace{circle\_area}_{function\ name}\left(\underbrace{a, b, c...}_{arguments}\right)\underbrace{:}_{eol}$$

- type following code in python playground.

```
def f():
    pass

dir()

f()

f
```

$$area = \pi * r^2$$

- math1.py (circle area, rectangule area, triangle area)
- defineFunction.py (help(sum))
- collision.py; use / to avoid collision
- keywordArgs.py

- practice: define a function with keyword arguments
    - (createList.py parseString(str, sep=','))
- defaultValue.py
- annotation1.py; wonderful use of keyword arguments
- annotation2.py; long large function
- ask.pys
- attribute.py
- optionalPositionalArgs.py
- innerFunction0.py
- innerFUnction1.py
- homework1
- homework2

Table of Contents

Prime

- prime0.py > straight forward, define function
- prime1.py > optimized by half
- prime2.py > define function isPrime()
- prime3.py > calculate range(40-50)
- prime4.py > define function rangePrime(x,y)

Table of Contents

plot

- plot0.py
- plot1.py
- plot2.py
- plot3.py
- plot-student-csv.py
- Practice: plot sin(x) and cos(x) in the same chart > plot4.py
- China-vs-USA.py

    - Online data
    - Online data
    - [Homework] Choose different two states, plot the data
- covid-19/covid0.py
- covid-19/covid1.py
- covid-19/covid2.py

Table of Contents

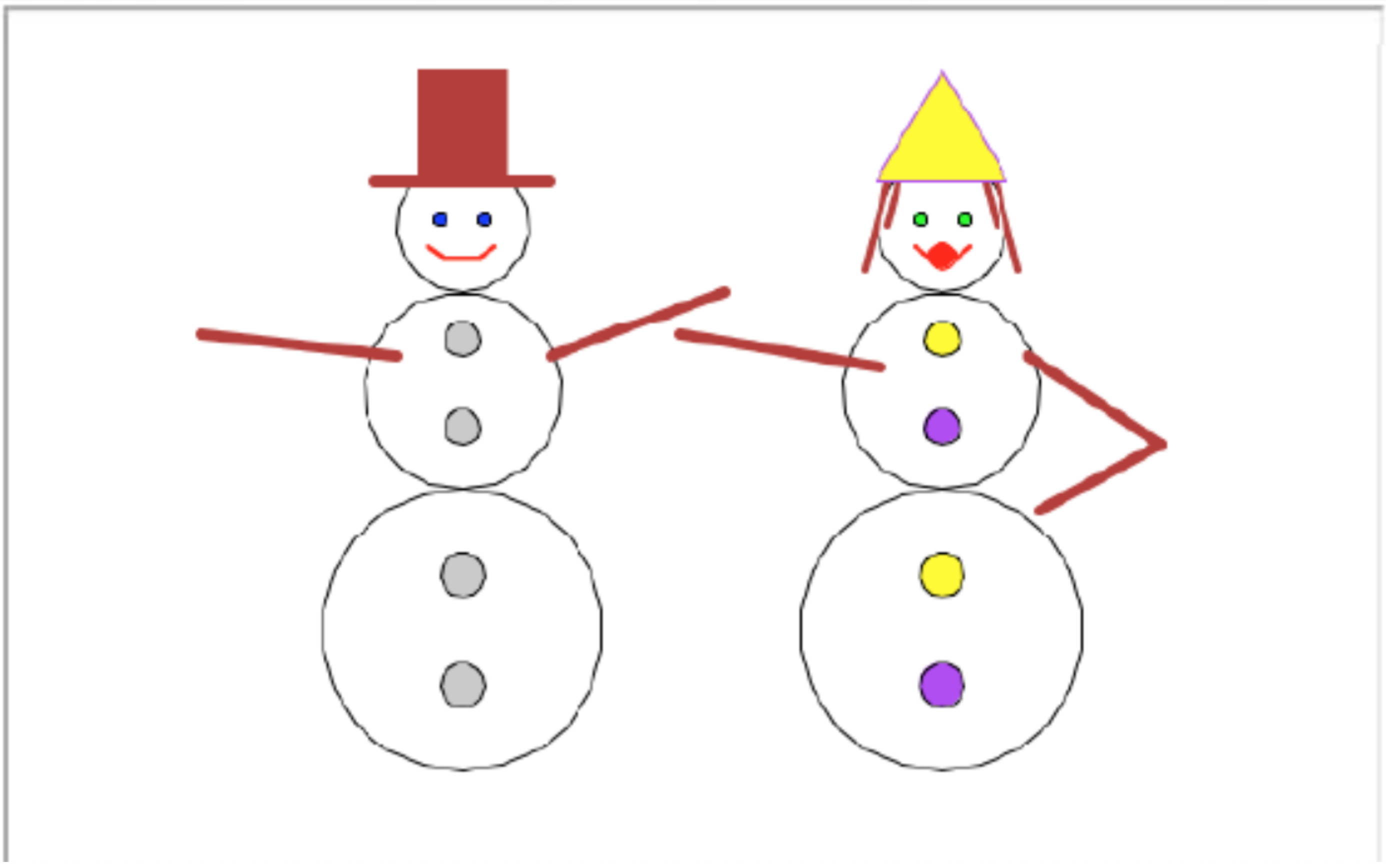
Terminal Games

- Check homework
- roll dice
    - dice.py
    - Practice: add total value of 2 dices
    - Practice: circle.py > circle_area(r)
- guess number
    - guessNumber.py
- ball game
  ball10.py

Table of Contents

draw snow man

-



- demo draw_snowman.py
- shapes.py
- testShapes.py
- drawSun.py; add snow man in the picture.
- homework> draw snowcouple

Table of Contents

turtle

```
python -m turtledemo
```

- turtle1.py; display turtle pen
- turtle2.py; basic turtle move
- turtle3.py; mouse click on turtle
- turtle4.py; random move on click
- turtle5.py; avoid turtle move out of window
- turtle6.py; avoid turtle move out of window
- turtle7.py; display card on turtle screen

- turtle8.py; draw star
- turtle9.py; draw half circle
- shapes.py; triangle, rectangle, line, circle
- testShapes.py; test all functions defined in shapes.py
- drawSun.py; drawing a sun and house by using shapes.py
  - assign homework draw snow couple

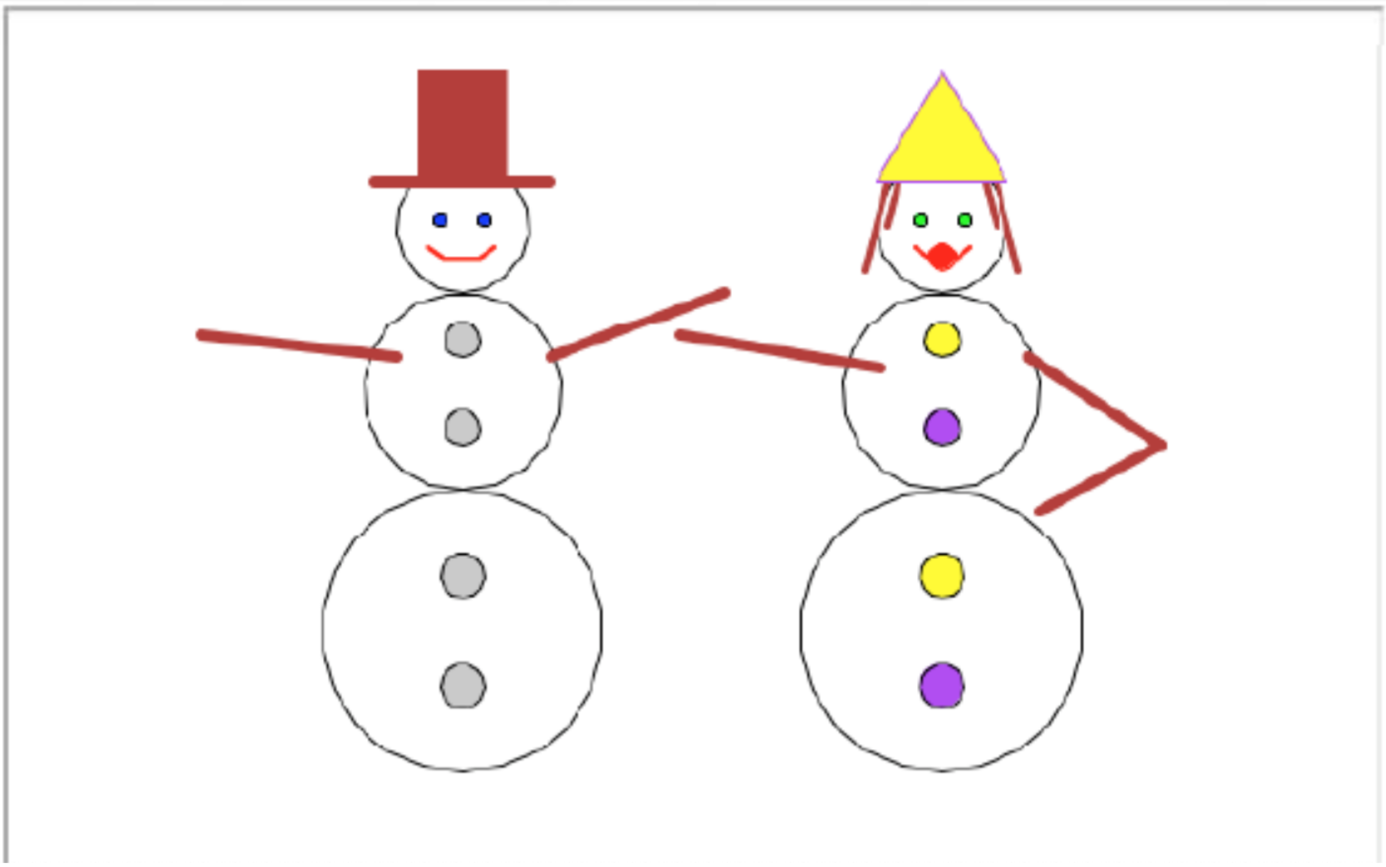


Table of Contents

ball game

- ball1.py [Display a ball at center of the screen.]
- ball2.py []
- ball3.py
- ball4.py
- ball5.py
- ball6.py
- ball7.py
- ball8.py

- ball9.py
- ball10.py [Final version of ball game.]

Table of Contents

Loop

- forLoop1.py
- forBreak.py
- forContinue.py
- forNested1.py; print right triangle
- forNested2.py; print Equilatera triangle
- forNested3.py; print diamond
- forNested4.py; define function for n
- forElse.py
- for1.py; generator
- for2.py; more generator
- while.py
    - Practice:

      ```
      We're on time 0
      We're on time 1
      We're on time 2
      We're on time 3
      ```

    - loop string
- whileElse.py
- guessNumber.py
    - assign homework to modify guessNumber.py for two players

Table of Contents

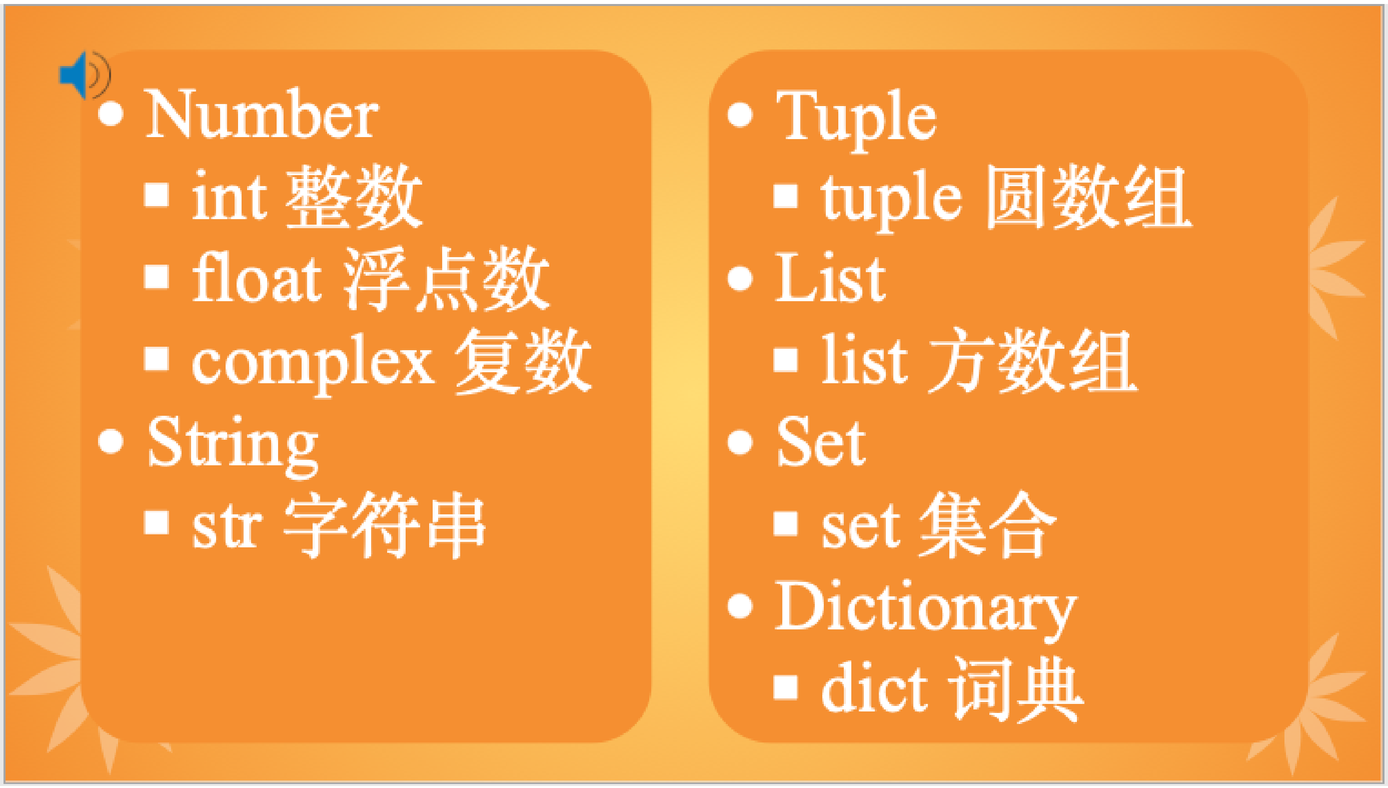
If-Else

```
- ifelse.py
- if-else.py
- Infinit loop while True: > input("Continue? (y/n)")
- [Practice]:
```

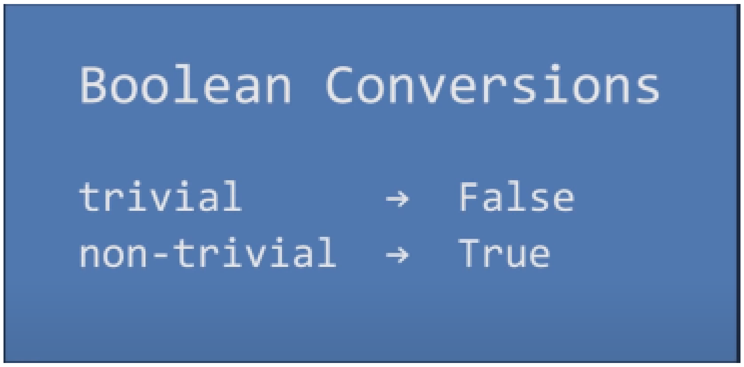
```
2, 4, 6, 8, 10
1, 3, 5, 7, 9
```

Table of Contents

Data Type

- 

  - Number
    - int 整数
    - float 浮点数
    - complex 复数
  - String
    - str 字符串
  - Tuple
    - tuple 圆数组
  - List
    - list 方数组
  - Set
    - set 集合
  - Dictionary
    - dict 词典
- python terminal
- simpleDataType.py; simple datatype, number, string, boolean



Boolean Conversions

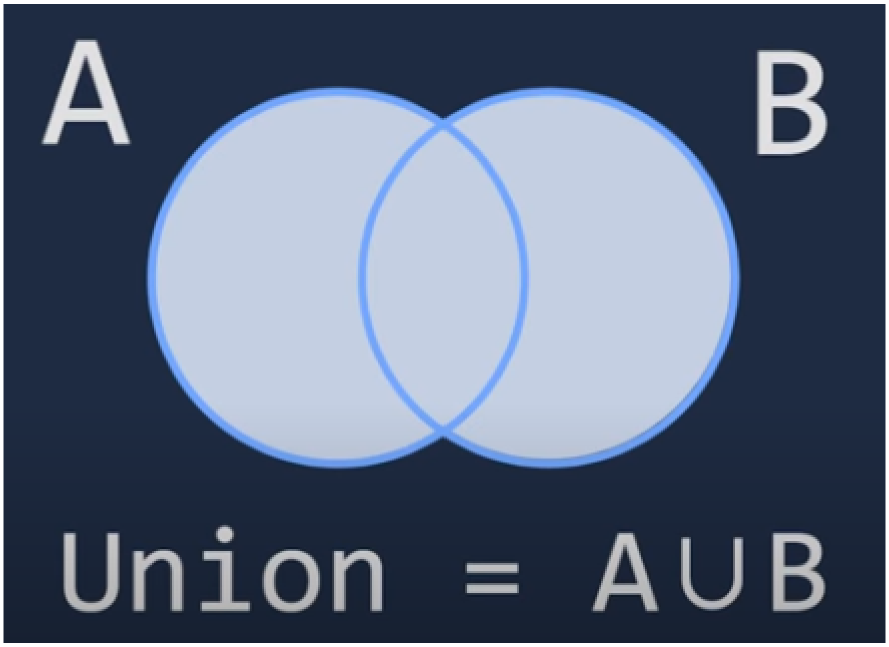
```
trivial      →  False
non-trivial  →  True
```

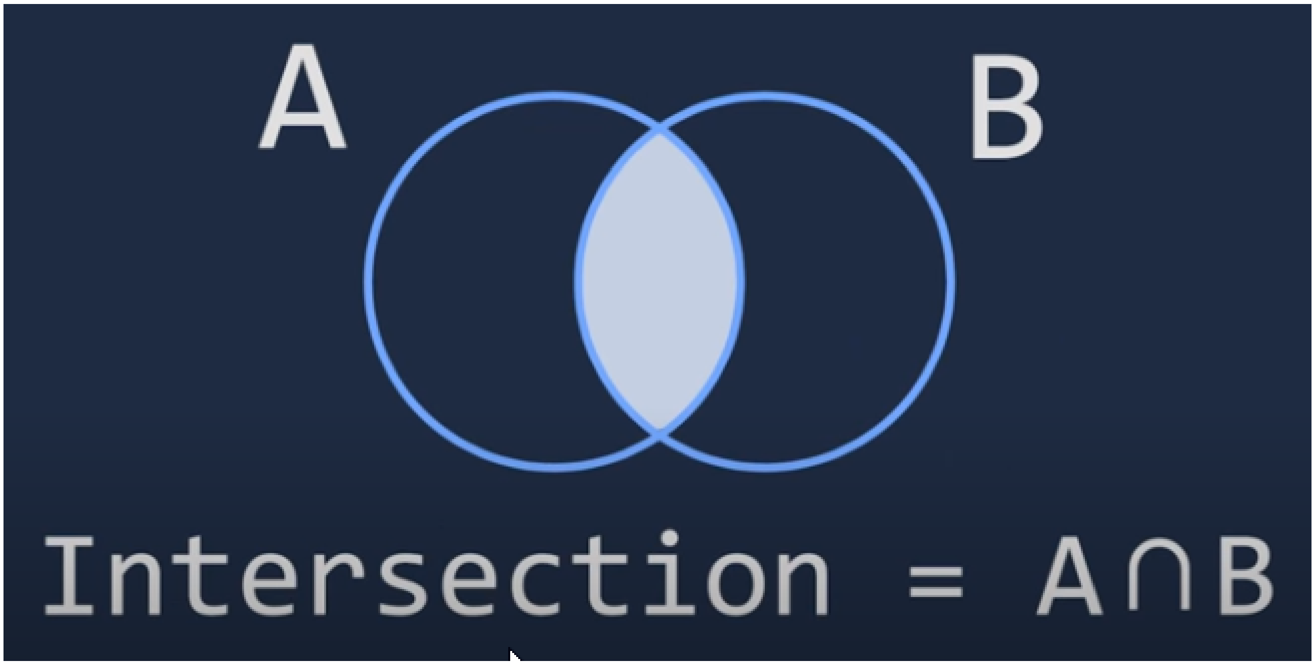
- int, float, complex > floatTest.py
- str > strTest.py; operation on string
- tuple > tupleTest.py
- list > listTest.py
- set > setTest.py

  we use set when the order and frequency of data is not matter

```
python >>>
myset = set()
dir(myset)
help(myset.add)
myset.add(1)
myset.add("hello")
myset.add(1)
```

second time use myset.add(1) will be ignored.
set do not contain duplicated element.





- dict > dictTest.py

Basic date and time types

- datetime1.py; other data type (datetime.date)
- datetime2.py;
- strftime() and strptime() Format Codes

Directive	Meaning	Example
%a	Weekday as locale's abbreviated name.	Sun, Mon, …, Sat (en_US);So, Mo, …, Sa (de_DE)
%A	Weekday as locale's full name.	Sunday, Monday, …, Saturday (en_US); Sonntag, Montag, …, Samstag (de_DE)
%w	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	0, 1, …, 6
%d	Day of the month as a zero-padded decimal number.	01, 02, …, 31
%b	Month as locale's abbreviated name.	Jan, Feb, …, Dec (en_US); Jan, Feb, …, Dez (de_DE)

Directive	Meaning	Example
%B	Month as locale’s full name.	January, February, …, December (en_US); Januar, Februar, …, Dezember (de_DE)
%m	Month as a zero-padded decimal number.	01, 02, …, 12
%y	Year without century as a zero-padded decimal number.	00, 01, …, 99
%Y	Year with century as a decimal number.	0001, 0002, …, 2013, 2014, …, 9998, 9999
%H	Hour (24-hour clock) as a zero-padded decimal number.	00, 01, …, 23
%I	Hour (12-hour clock) as a zero-padded decimal number.	01, 02, …, 12
%p	Locale’s equivalent of either AM or PM.	AM, PM (en_US); am, pm (de_DE)
%M	Minute as a zero-padded decimal number.	00, 01, …, 59
%S	Second as a zero-padded decimal number.	00, 01, …, 59
%f	Microsecond as a decimal number, zero-padded on the left.	000000, 000001, …, 999999
%z	UTC offset in the form ±HHMM[SS[.ffffff]] (empty string if the object is naive).	(empty), +0000, -0400, +1030, +063415, -030712.345216
%Z	Time zone name (empty string if the object is naive).	(empty), UTC, GMT
%j	Day of the year as a zero-padded decimal number.	001, 002, …, 366
%U	Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0.	00, 01, …, 53
%W	Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0.	00, 01, …, 53

Directive	Meaning	Example
%c	Locale’s appropriate date and time representation.	Tue Aug 16 21:30:00 1988 (en_US); Di 16 Aug 21:30:00 1988 (de_DE)
%x	Locale’s appropriate date representation.	08/16/88 (None); 08/16/1988 (en_US); 16.08.1988 (de_DE)
%X	Locale’s appropriate time representation.	21:30:00 (en_US); 21:30:00 (de_DE)
%%	A literal '%' character.	%
%G	ISO 8601 year with century representing the year that contains the greater part of the ISO week (%V).	0001, 0002, …, 2013, 2014, …, 9998, 9999
%u	ISO 8601 weekday as a decimal number where 1 is Monday.	1, 2, …, 7
%V	ISO 8601 week as a decimal number with Monday as the first day of the week. Week 01 is the week containing Jan 4.	01, 02, …, 53

- datetime3.py; convert string to datetime by strptime(string, format)
- datetime4.py; differences between datetime, date, time

Table of Contents

Python playground and help document

- python >>> help(print) (positional arguments, keyword arguments)
- Practice: different print statements
- hello/print.py
- hello/print-string.py
-

Table of Contents

File access

- file0.py (write to file)
- file1.py (read and write to existing file)
- file2.py (with open, auto close)
- file3.py (dump json, write to json file)
- file3a.py (read json from file)
- file3b.py (read json from string)
- file4.py (pandas read csv)
- file5.py (read csv file, and plot the data)
- file6.py (write dict to csv file)
- readJson.py
- csvReader.py

Table of Contents

Review

- Markdown document
- ball game
- draw snowman
- file access (read/write plain text, csv, json)
- plot
- covid_19
- debug python code
- database access (CRUD)
- Postman to test web service
- application web server
- react JS front end GUI server

Table of Contents

install and using QuickType

QuickType website

- QuickType Installation

```
npm install -g quicktype
quicktype --version
```

- Python code generation

```
quicktype ./data/student.json -o student.py
```

- install

```
npm intall -g quicktype
quicktype --version
```

- generate python code based on Json

```
quicktype ./data/student.json -o student.py
```

- book.py > **init**, **str**
- student.py constructor, **repr** abstraction
- user.py, User, SubUser inheritence testUser
- person, teacher, student inheritence

YouTube Classes

classes are foundmantal tools to any object oriented programing language, think of class as template for creating object and related data and functions that do interesting things with that data. Python make it easy to create classes

Table of Contents

Sqlite

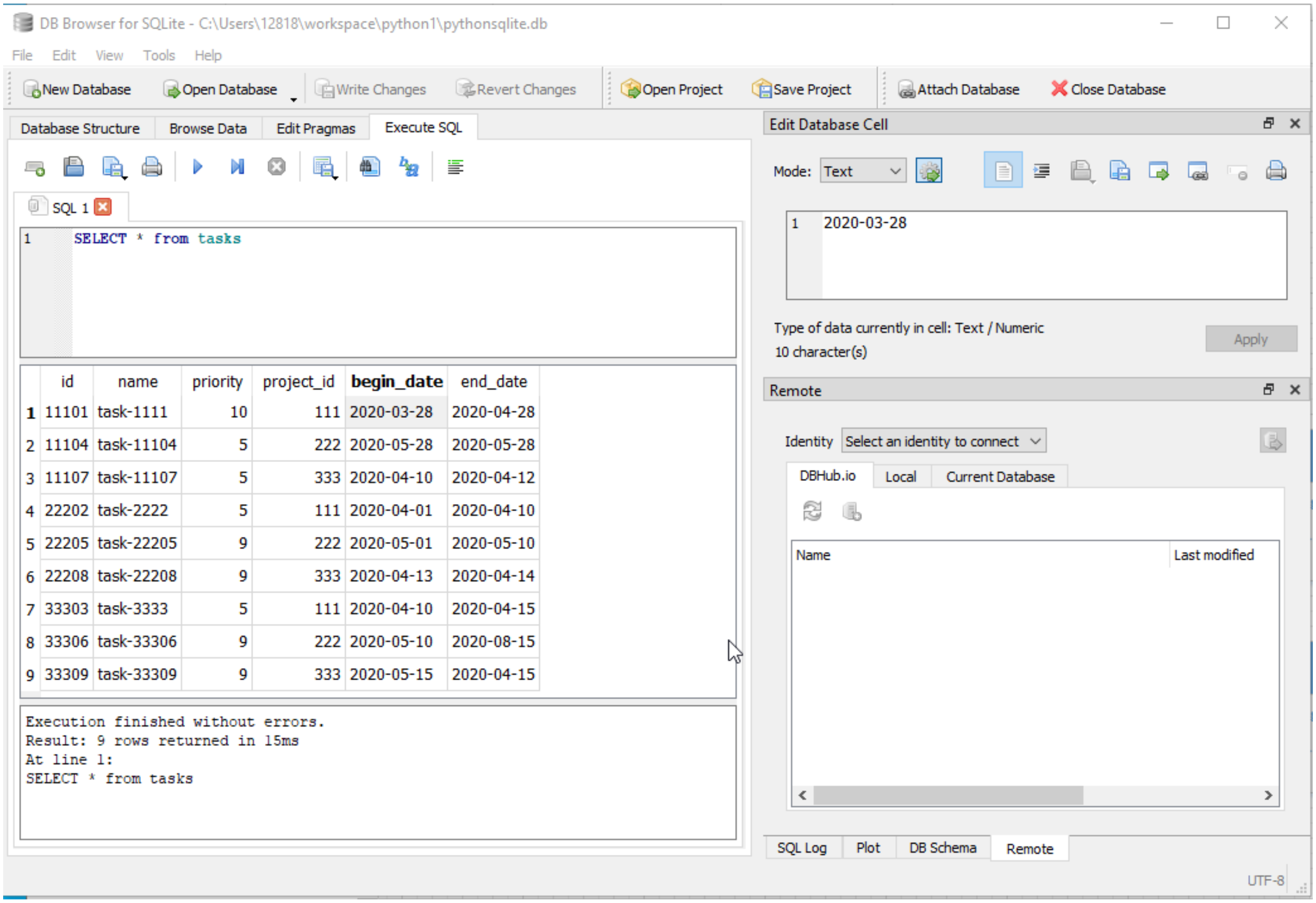
- sqlite0.py > create connection
- sqlite1.py
- sqlite2.py
- install DB browser for SQLite

Google Search: DB Browser for Sqlite

SQLite GUI Download Website

SQLite Browser for MacOS

File: DB.Browser.for.SQLite-3.12.1-win64-v2.msi



```
SELECT * from tasks
```

	id	name	priority	project_id	begin_date	end_date
1	11101	task-1111	10	111	2020-03-28	2020-04-28
2	11104	task-11104	5	222	2020-05-28	2020-05-28
3	11107	task-11107	5	333	2020-04-10	2020-04-12
4	22202	task-2222	5	111	2020-04-01	2020-04-10
5	22205	task-22205	9	222	2020-05-01	2020-05-10
6	22208	task-22208	9	333	2020-04-13	2020-04-14
7	33303	task-3333	5	111	2020-04-10	2020-04-15
8	33306	task-33306	9	222	2020-05-10	2020-08-15
9	33309	task-33309	9	333	2020-05-15	2020-04-15

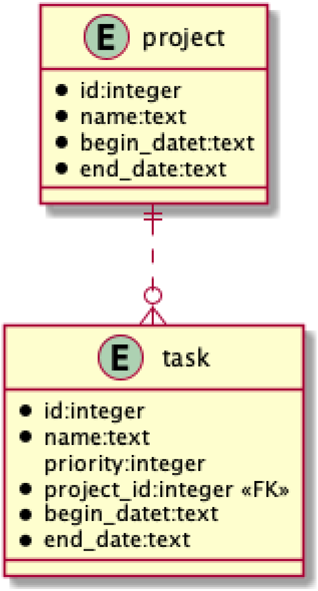
```
Execution finished without errors.
Result: 9 rows returned in 15ms
At line 1:
SELECT * from tasks
```

Mode: Text

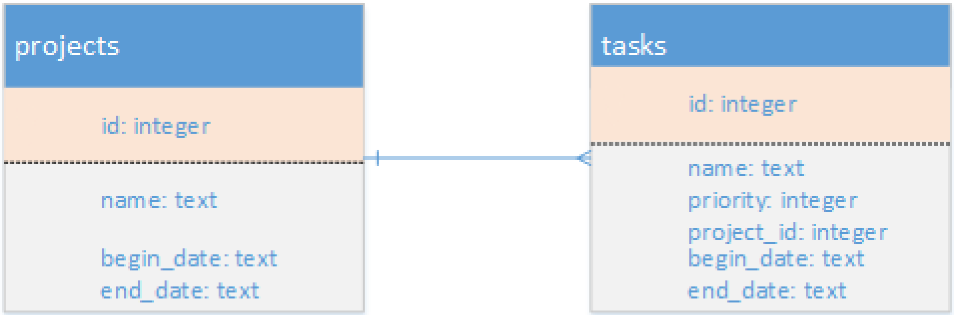
```
1  2020-03-28
```

Type of data currently in cell: Text / Numeric
10 character(s)

- sqlite4.py
- sqlite5.py
- sqlite6.py



- sqlite7.py > build relational data
- sqlite8.py > show relation between project and task



- review bookdb.py
- sqlite9.py > create books table
- sqlite10.py > insert data into books table
- sqlitebookdb.py > build CRUD
- app5.py > use sqlitebookdb.py to provide service
  use Postman to check the service.

Table of Contents