

Python level-I

Table of Contents

- [Python level-I](#)
 - [Table of Contents](#)
 - [Familiar with your keyboard](#)
 - [Using Markdown](#)
 - [Getting start](#)
 - [playground and help document](#)
 - [print](#)
 - [Data Type](#)
 - [Operators](#)
 - [If-Else Execution Control](#)
 - [Loop](#)
 - [turtle](#)
 - [draw snow man](#)
 - [Function](#)
 - [ball game](#)
 - [Simple math](#)
 - [Dice](#)
 - [Prime](#)
 - [File access](#)
 - [plot](#)
 - [Python class](#)
 - [OOP](#)
 - [install npm](#)
 - [ReactJS](#)
 - [App server](#)
 - [Mongo DB](#)
 - [Terminal Games](#)
 - [Review](#)
 - [install and using QuickType](#)
 - [Sqlite](#)
 - [tkinter GUI](#)
 - [OpenGL](#)

[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()

Key	Name
#	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
```

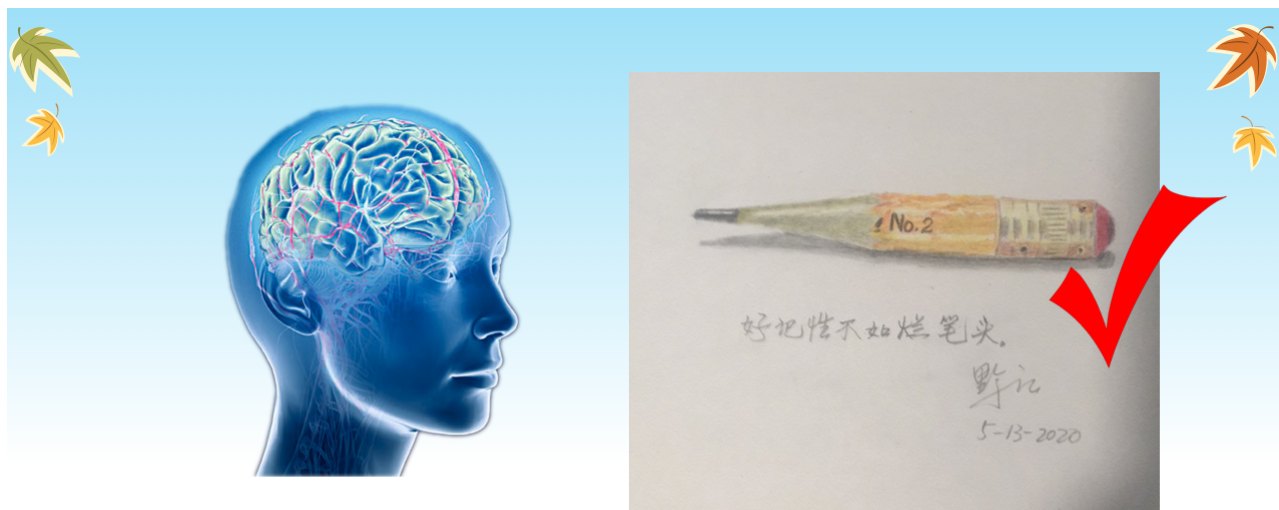
- Hight light block of code
- Ctrl+c: copy
- Ctrl+v: paste

- 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



Good memory cannot be as good as Markdown

-
- 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

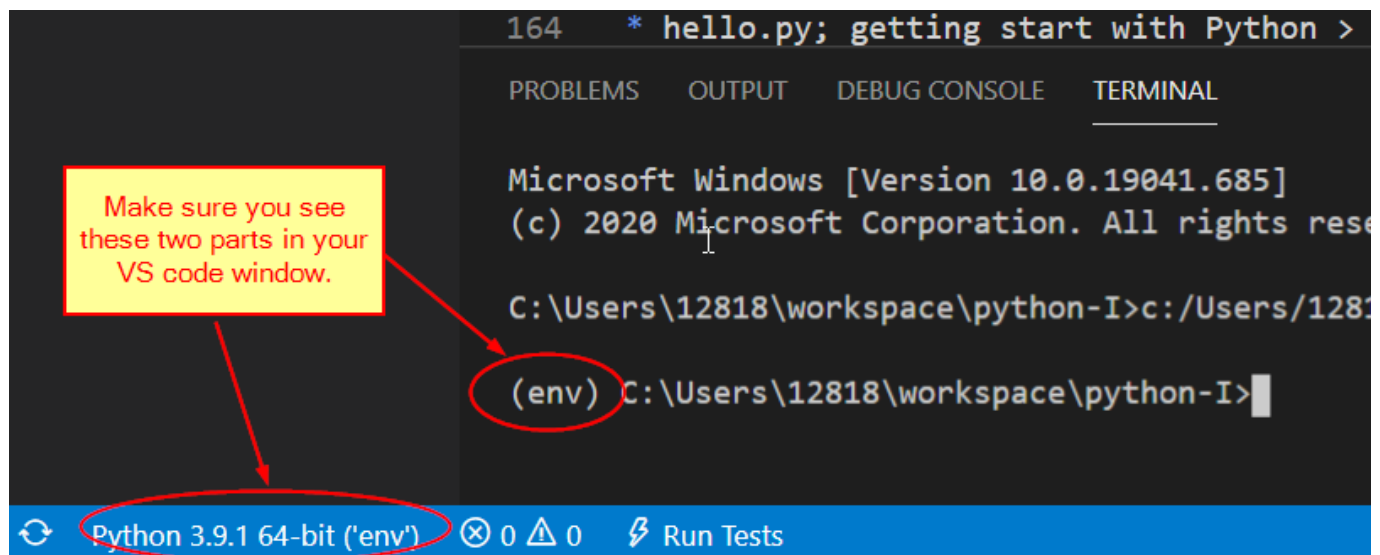
```
print("Hello, world!")  
a = 4  
b = 5  
print(a+b)
```

save to first.py

```
python first.py
```

- build virtual environment

```
python -m venv env
```



- familiar with VSCode. [VS code](#)
- convert python script to exe

```
pip install pyinstaller  
pyinstaller --onefile -w 'filename.py'
```

[Table of Contents](#)

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](#)

print

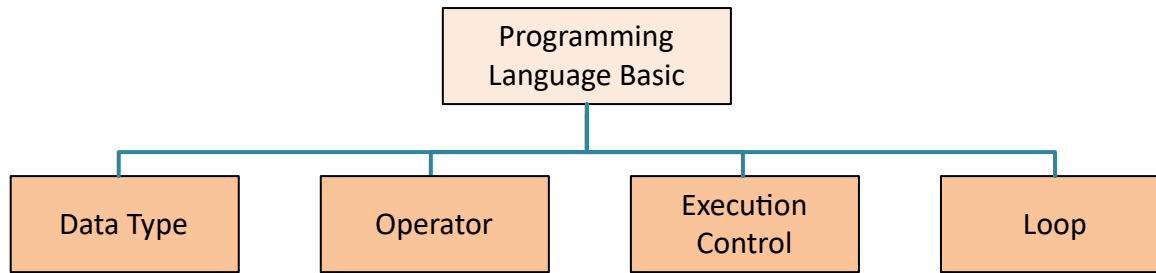
- hello.py; getting start with Python > hello.ReadMe.md
- print.py; hello/print.py
- helloHim.py; intruduce input() function
- print-string.py;
- input.py
- guessNumber.py
- dice.py; introduce random module, dice/dice1.py
- dice2.py; figure out possibility, understand how computer do things

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2210	\sqcup	Σ	$-$	\mp	\dagger	$/$	\diagdown	$*$	\circ	\cdot	$\sqrt{}$	$\sqrt[3]{}$	$\sqrt[4]{}$	∞	∞	\angle
2220	\angle	\triangleleft	\triangleleft	$ $	\dagger	\parallel	\nparallel	\wedge	\vee	\cap	\cup	\int	\iint	\iiint	\oint	\Re
2230	\oint	f	\oint	\oint	\therefore	\because	$:$	$::$	\div	\div	\div	\sim	\sim	\approx	\approx	\approx
2240	\wr	\approx	\approx	\approx	\neq	\equiv	\equiv	\equiv	\approx	\neq	\equiv	\equiv	\equiv	\neq	\subsetneq	\supsetneq
2250	\doteq	\div	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq	\doteq
2260	\neq	\equiv	\equiv	\equiv	\leq	\geq	\leq	\geq	\leq	\geq	\ll	\gg	\oslash	\neq	\neq	\neq
2270	\leq	\geq	\leq	\geq	\neq	\neq	\leq	\geq	\neq	\neq	$<$	$>$	\leq	\geq	\leq	\geq
2280	\neq	\neq	\subset	\supset	\subsetneq	\supsetneq	\subset	\supset	\subsetneq	\supsetneq	\subset	\supset	\subset	\supset	\subset	\supset
2290	\sqsubset	\sqsupset	\sqsubset	\sqsupset	\sqcup	\oplus	\ominus	\otimes	\oslash	\odot	\odot	\otimes	\ominus	\ominus	\boxplus	\boxminus
22A0	\boxtimes	\boxdot	\vdash	\dashv	\top	\bot	\vdash	\vdash	\vdash	\vdash	\vdash	\vdash	\vdash	\vdash	\vdash	\vdash
22B0	\curvearrowright	\curvearrowleft	\triangleleft	\triangleright	\triangleleft	\triangleright	\hookrightarrow	\hookrightarrow	\hookrightarrow	\div	\top	\vee	\wedge	∇	\triangleleft	\triangleleft
22C0	\wedge	\vee	\cap	\cup	\circ	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot
22D0	\subset	\supset	\subset	\supset	\subset	\supset	\subset	\supset	\subset	\supset	\subset	\supset	\subset	\supset	\subset	\supset
22E0	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq	\neq
22F0	\therefore	\because	\in	\in	\in	\in	\in	\in	\in	\in	\in	\in	\in	\in	\in	\in
2300	\emptyset	\emptyset	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft	\triangleleft
2310	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil
2320	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil
2330	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil	\lceil	\rceil

Math Symbols

[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

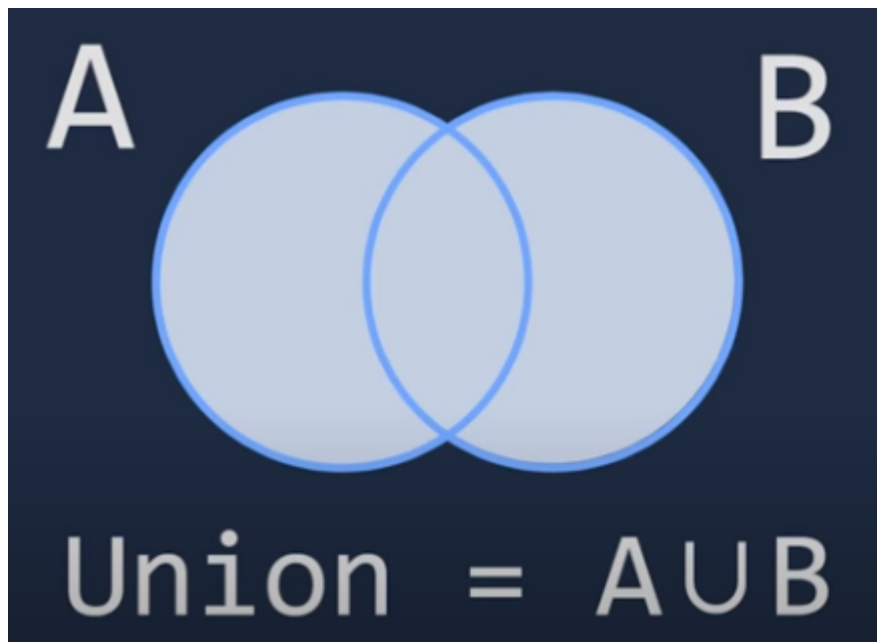
list	tuple
add data	cannot be changed

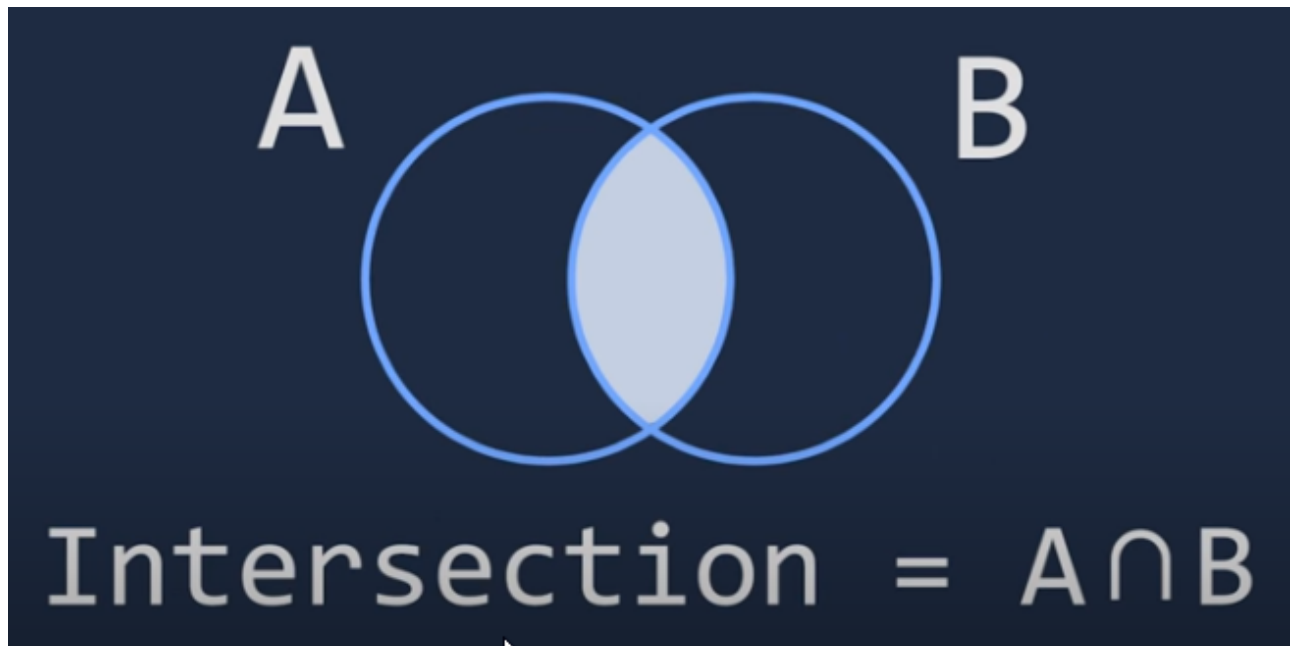
list	tuple
remove data	cannot be changed
change data	immutable
slow operation	made quickly

- tupleList.py
 - create a list [expr for val in collection] [expr for val in collection if]
- 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; difference between two dates
- strftime() and strptime() Format Codes
- datetime3.py; convert string to date
- datetime4.py; rocket launch date time.

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)
%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

Directive	Meaning	Example
%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 \pm 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
%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

Operators

./operator

- Arithmetic Operators (+, -, *, /, %, **, //)
-

Arithmetic Operator

Operator	Name	Example
+	Addition	a + b
-	Subtraction	a - b
*	Multiplication	a * b
/	Division	a / b
%	Modulus	a % b
**	Exponentiation	a ** b
//	Floor division	a // b

- Bitwise Operators

bitwise sample code

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off

- Assignment Operators

Assignment sample code

Operator	Example	Same As
----------	---------	---------

Operator	Example	Same As
=	x = 5	x=5
+=	x += 3	x=x+3
-=	x -= 4	x=x-4
*=	x *= 5	x=x*5
/=	x /= 4	x=x/4
%=	x %= 5	x=x%5
//=	x //= 3	x=x//3
=	x **= 2	x=x2
&=	x &= 4	x=x&4
=	x = 3	x=x 3
^=	x ^= 4	x=x^4
>>=	x >>=3	x=x>>3
<<=	x <<=2	x=x<<2

- Comparison Operators

Comparison Sample code

Operator	Name	Example
==	Equal	a == b
!=	Not Equal	a != b
<	Less than	a < b
>	Greater than	a > b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b

- Logical Operators

Logical Sample code

Operator	Description	Example
and	Returns True if both statements are true	a<5 and a>10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the returns	not (x<10)

- Identity Operators

Identity Sample Code

Operator	Description	Example
is	Returns True if both variables are the same object	a is b
is not	Returns True if both variables are not the same object	a is not b

- Membership Operators

Membership Sample Code

Operator	Description	Example
in	Returns True if a sequence with the specified value is present in the object	a in y
not in	Returns True if a sequence with the specified value is not present in the object	x not in y

- Ternary Operators

Ternary Sample Code

if-else Statement structure:

```
value_if_true if condition else value_if_false
```

Statement example:

```
a = 5
s = "Hello" if a==5 else "A is not 5."
```

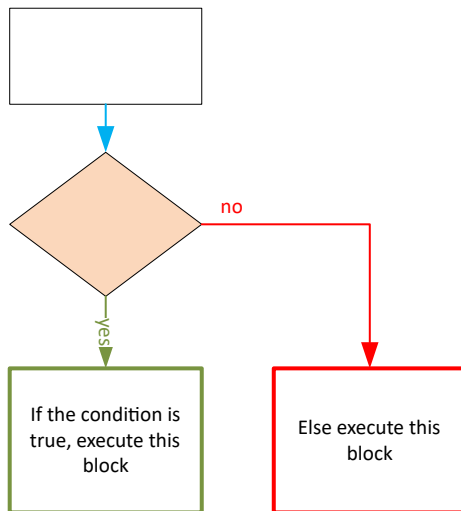
logical Statement:

```
a=5
b=11
min1 = a<b and a or b
print(min1)
```

tuple statement

```
min1 = (b, a)[a < b]
```

If-Else Execution Control



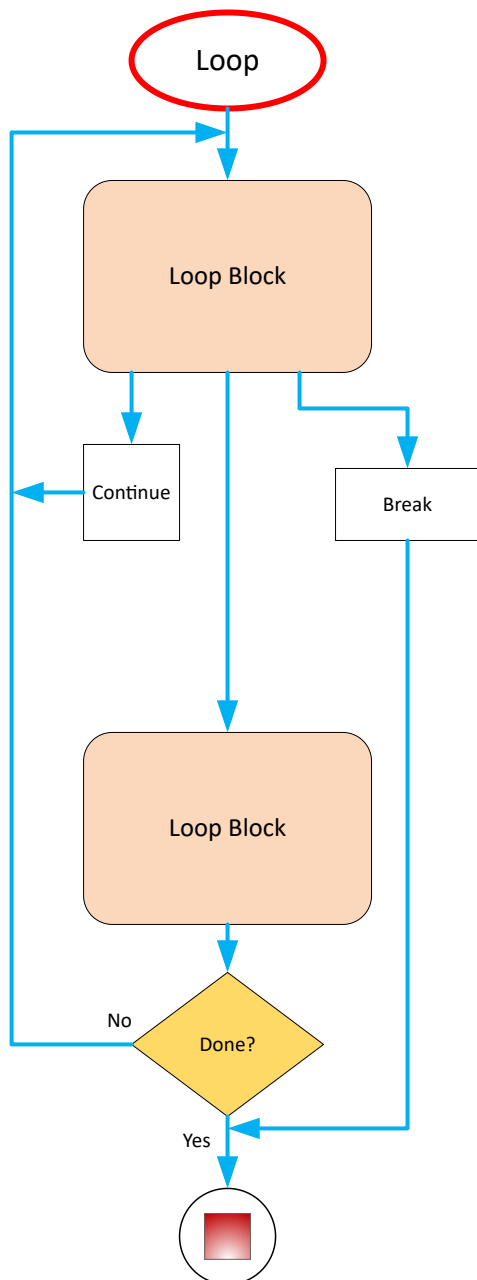
./if-else

- [Sample 1](#)
- [Sample 2](#)
- [Sample 3](#)
 - Infinit loop while True: > input("Continue? (y/n)")
 - [Practice:](#)

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

[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
```

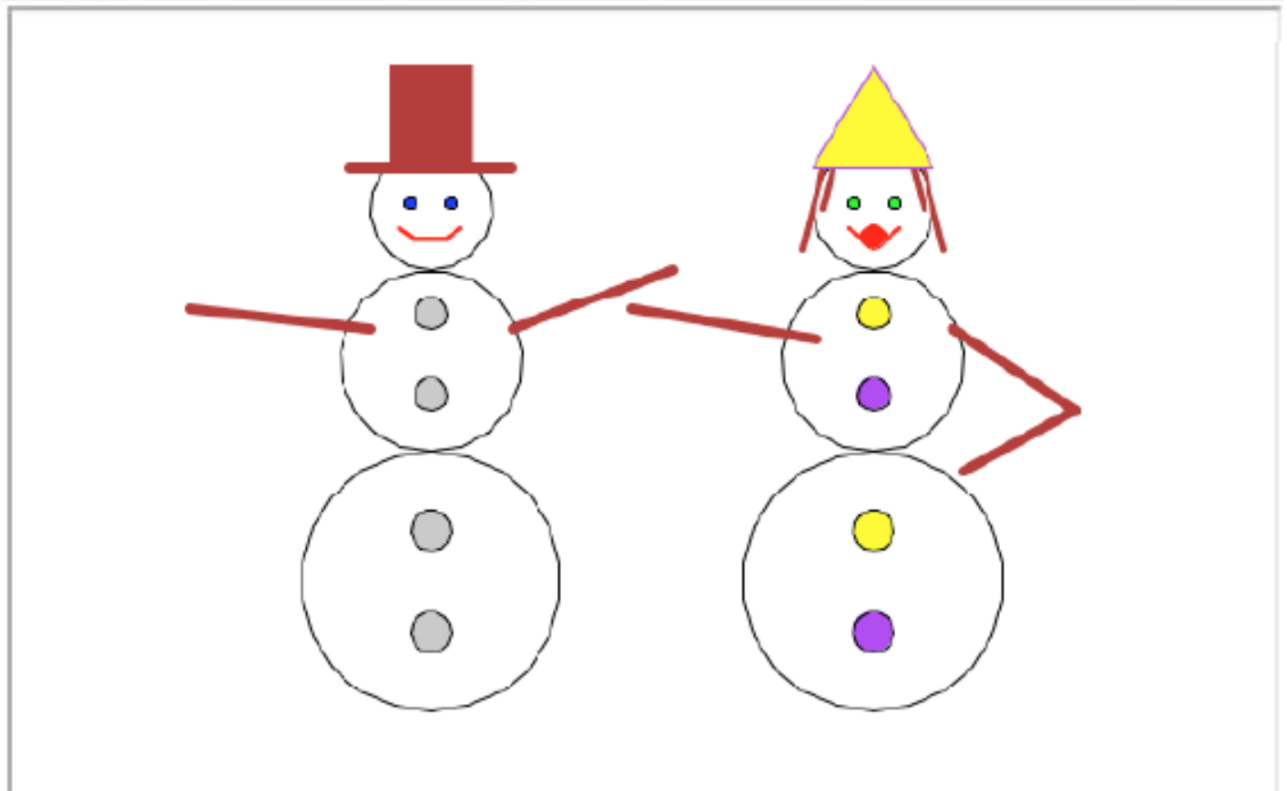
- loop string
 - whileElse.py
 - guessNumber.py
 - assign homework to modify guessNumber.py for two players
-

[Table of Contents](#)

turtle

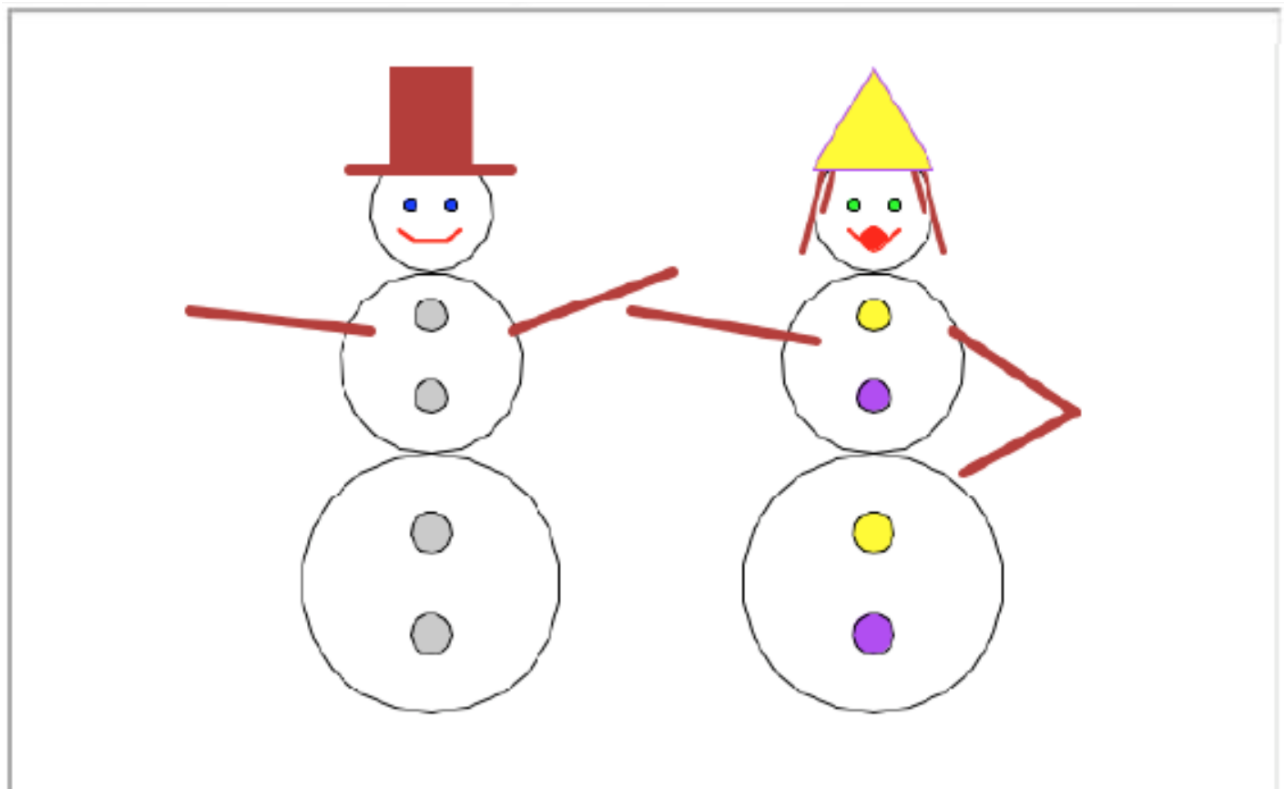
```
python -m turtledemo
```

- 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](#)

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](#)

Function

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

- define a function `def {keyword} {function_name} {arguments}`
- type following code in python playground.

```
def circleArea(radius):
    return 3.141593 * radius**2

def f():
    pass

dir()

f()

f
```

`area = pi * r^2`

if you don't return value from function, you will get None when you assign the value to a variable.

- math1.py (circle area, rectangle area, triangle area)
- defineFunction.py (help(sum))

- collision.py; use / to avoid collision,
 - keywordArgs.py; positional arguments first
 - 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](#)

ball game

- ball1.py [Display a ball at center of the screen.]
 - ball2.py [Display a ball at the screen bottom]
 - ball3.py [Use left/right arrow key to control the ball]
 - ball4.py [Use space key to fire the ball]
 - ball5.py [Move the ball gradually on fire.]
 - ball6.py [put ball back when it moves out of the screen]
 - ball7.py [Display bird and background.]
 - ball8.py [Move the bird from right to left.]
 - ball9.py [Move the bird back to start position once it is hitted.]
 - ball10.py [Final version of ball game.]
-

[Table of Contents](#)

Simple math

./mymath

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

- `math0.py`/ built-in functions, `abs()`, `pow()`, `sum()`, `max()`, `min()`, `round()`

Square Root $x = \sqrt{x^2} = x^{\frac{1}{2}}$ $4 = \sqrt{16} = 16^{\frac{1}{2}}$

- [math1.py](#); functions defined in math module, `sqrt()`, `ceil()`, `floor()`, `sin()`, `cos()`
- [math2.py](#)
- [math2.py](#)
- [solution.py](#)

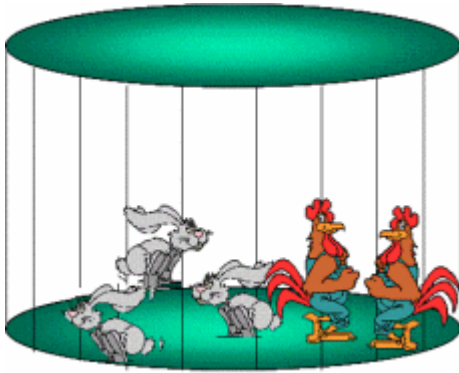
$A = \pi r^2$, where **A** is area of a circle, **r** is radius of the circle.

- `circle.py`



几个老头儿去赶集，上街买了一堆梨，一人一个多一个，一人两梨少两梨。问：几个老头儿几个梨？

- linear1.py; 老头儿们买梨。



- linear2.py
- linear3.py
- linear4.py
- linear5.py
- perfectNumber1.py
- Volumn of Sphere $V = \frac{4}{3} \pi r^3$
- Volumn of Cylinder $V = r^2 \pi \cdot h$
- Triangle area $area = \frac{1}{2} (b \cdot h)$

\cdots

- 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 n th **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

More Exercises: [Exercises and Solutions](#)

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

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

File access

Text Files**Binary Files**

Plain Text, XML, HTML, JSON, Source Code, CSV

Compiled code, App data, Media files, images, audio, video

- 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

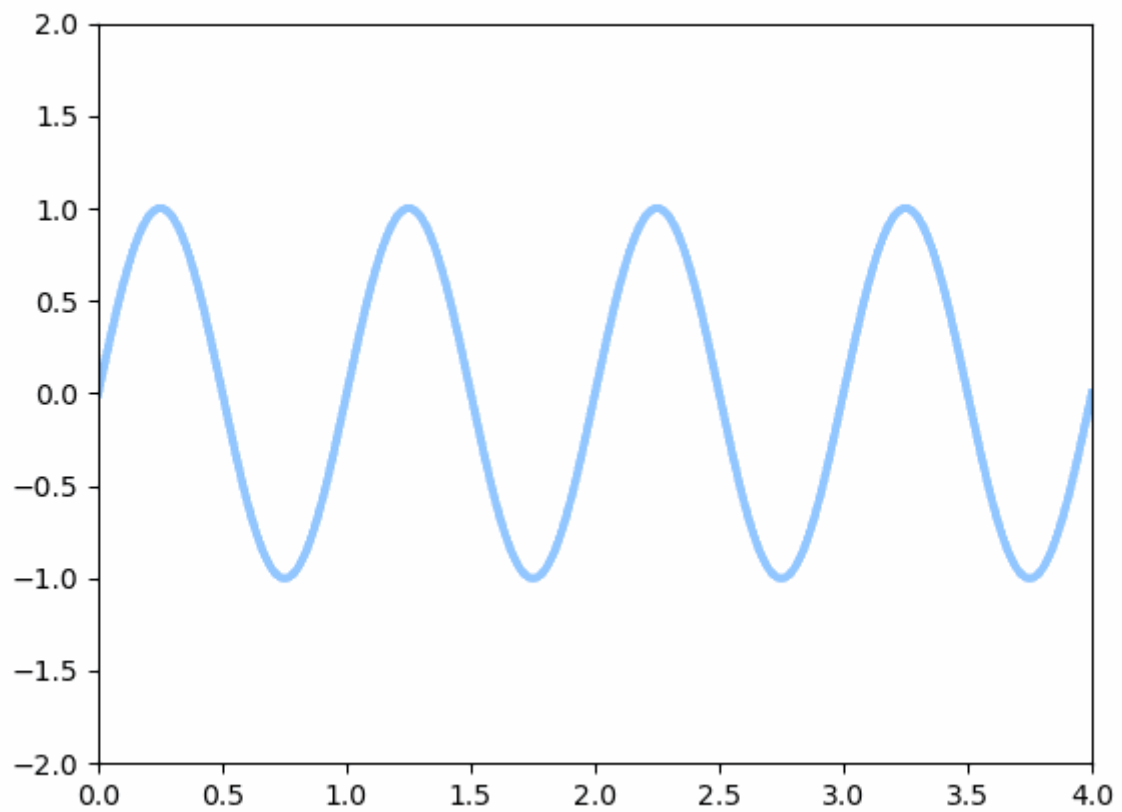
[Table of Contents](#)

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

[Animations with Matplotlib](#)

- movingSinWave.py



- growingCoil.py

Creating a growing coil with matplotlib!

- stockDynamic.py & stock.txt
- 3dAnimation.py

[Download ImageMagick](#) Install Magic by running file:

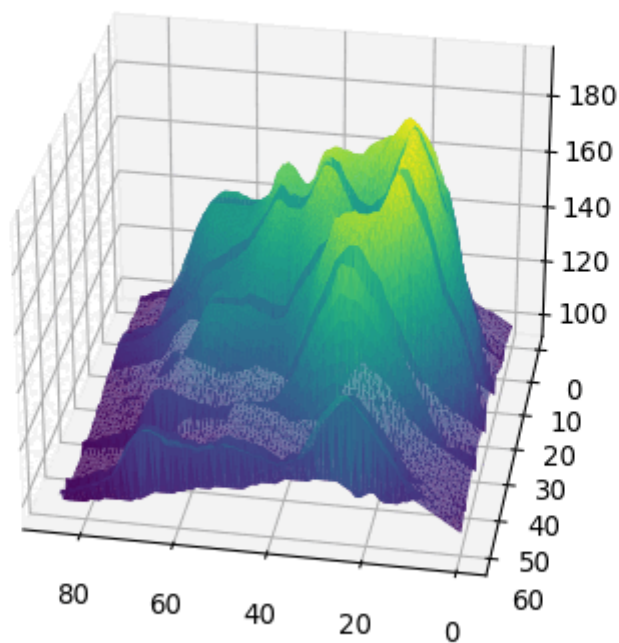
ImageMagick-x86_64-pc-windows.exe

after installation finished check the installation:

```
magick logo: logo.gif
magick identify logo.gif
magick logo.gif win:

magick -delay 10 Volcano_*.png volcano.gif
```

The animated gif file will be created.



[Matplotlib Exercises and Solution](#)

[Table of Contents](#)

Python class

```
class User:
    pass
```

- assign fields to an instance of User

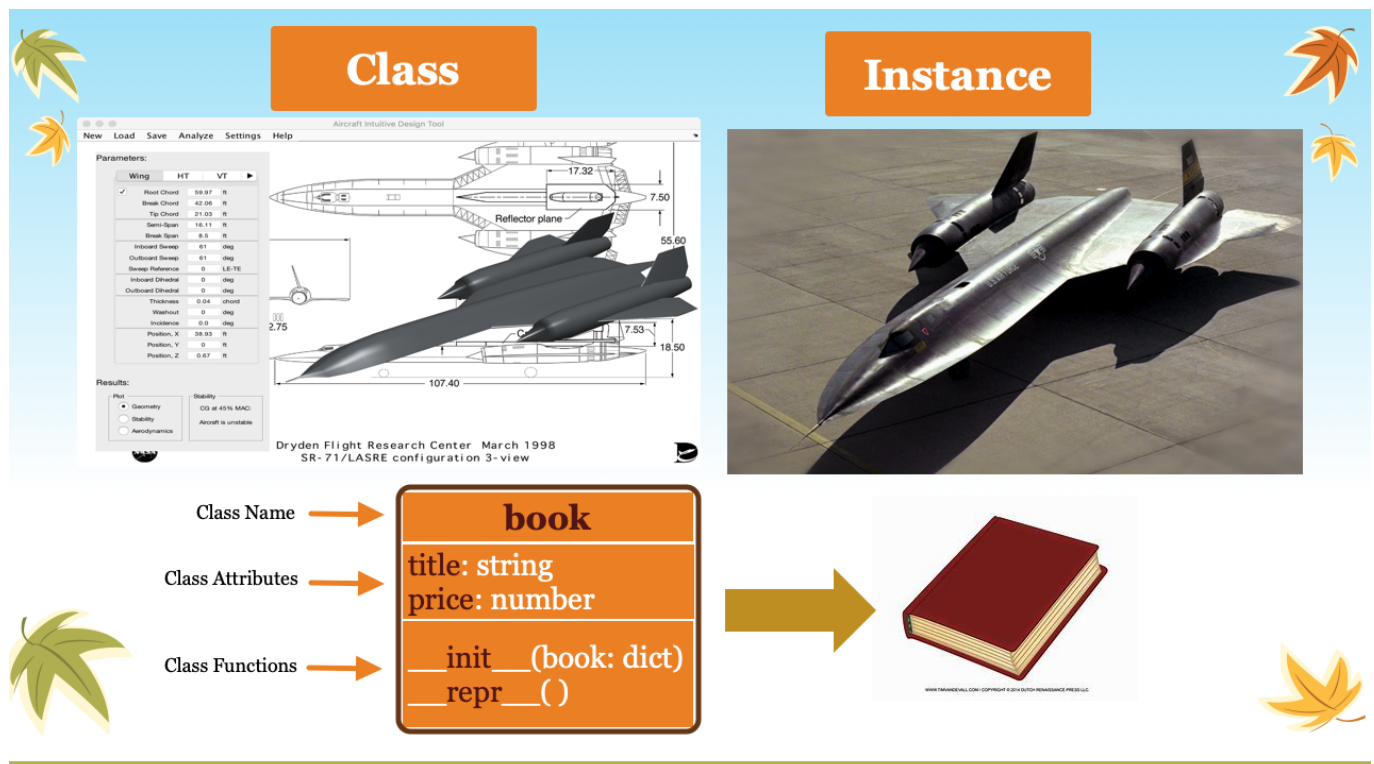
Python Classes

- 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
- ☐ class8.py; inherit from Enum
- ☐ class9.py; difference between **init** and **new**
- ☐ class10.py; override **new**() constructor
- ☐ class11.py; issue caused by class level variables
- ☐ class12.py; different object with different attributes
- ☐ class13.py; class level variables
- ☐ class14.py; define internal function by outside function, the benefit is use this outside function by more than one class.
- ☐ class15.py; function call function
- ☐ class16.py; multiple inheritance
- ☐ class17.py; reverse given string
- ☐ class18.py; define Website class

- ☐ 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
- ☐ snowman.py & drawSnowMan.py; draw snowman with class

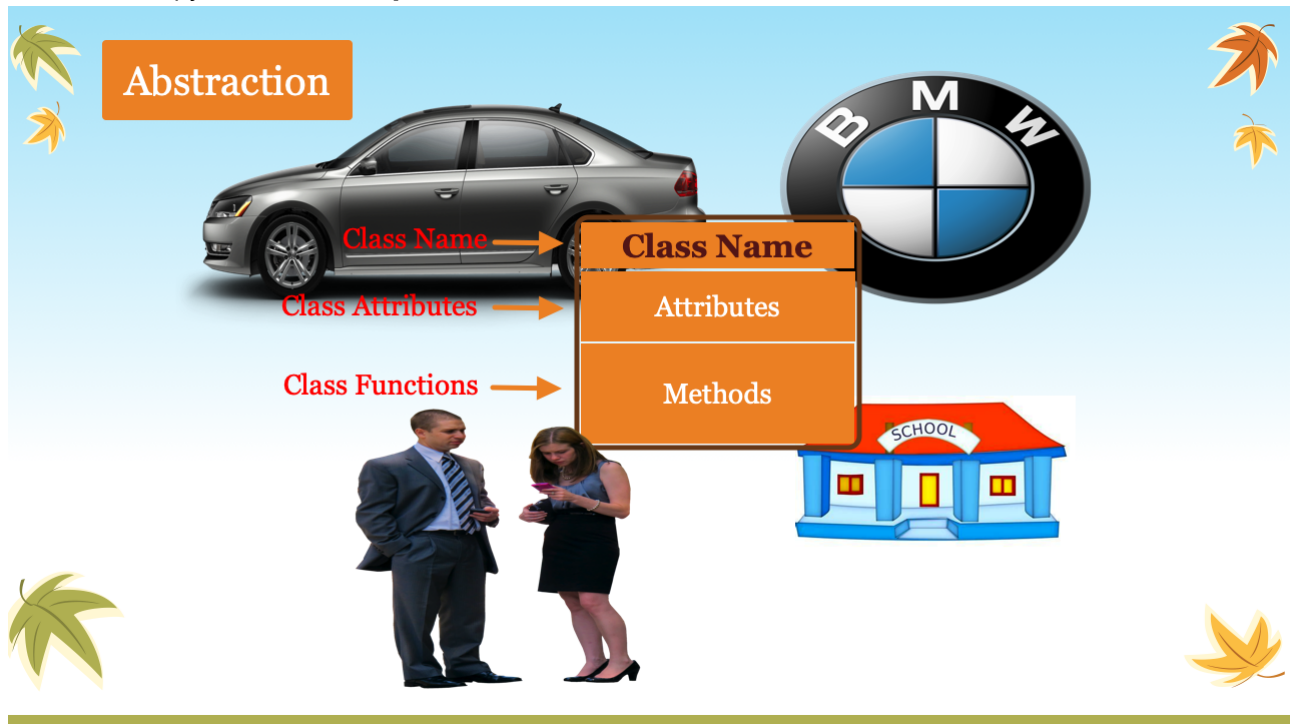
Table of Contents

OOP

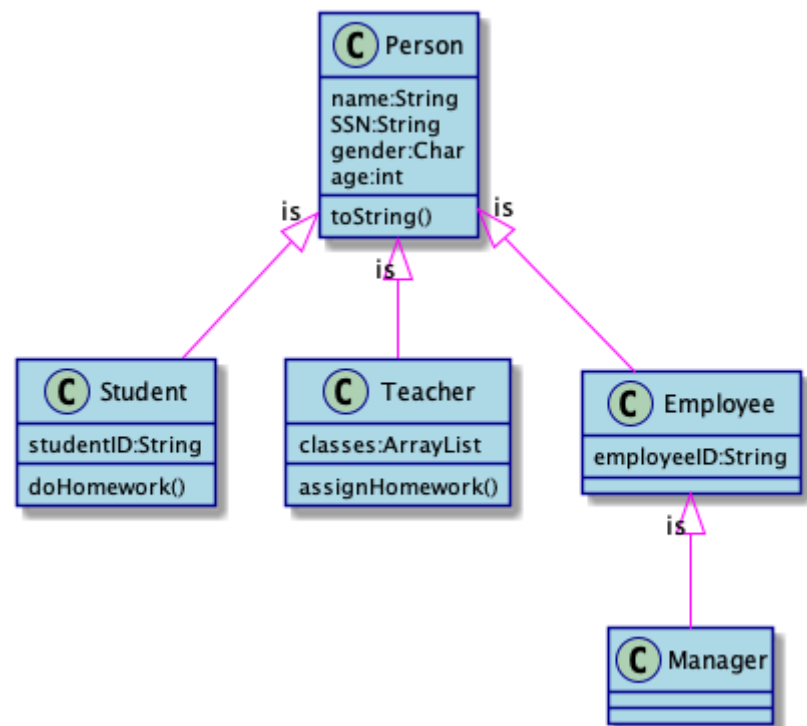


- class book, **init**, **repr**

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



- user.py, User, SubUser inheritance testUser



- person, teacher, student inheritance
[YouTube Classes Python Classes](#)

[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

Windows Installer
node-v12.18.1-x86.msi

Current
Latest Features

macOS Installer
node-v12.18.1.pkg

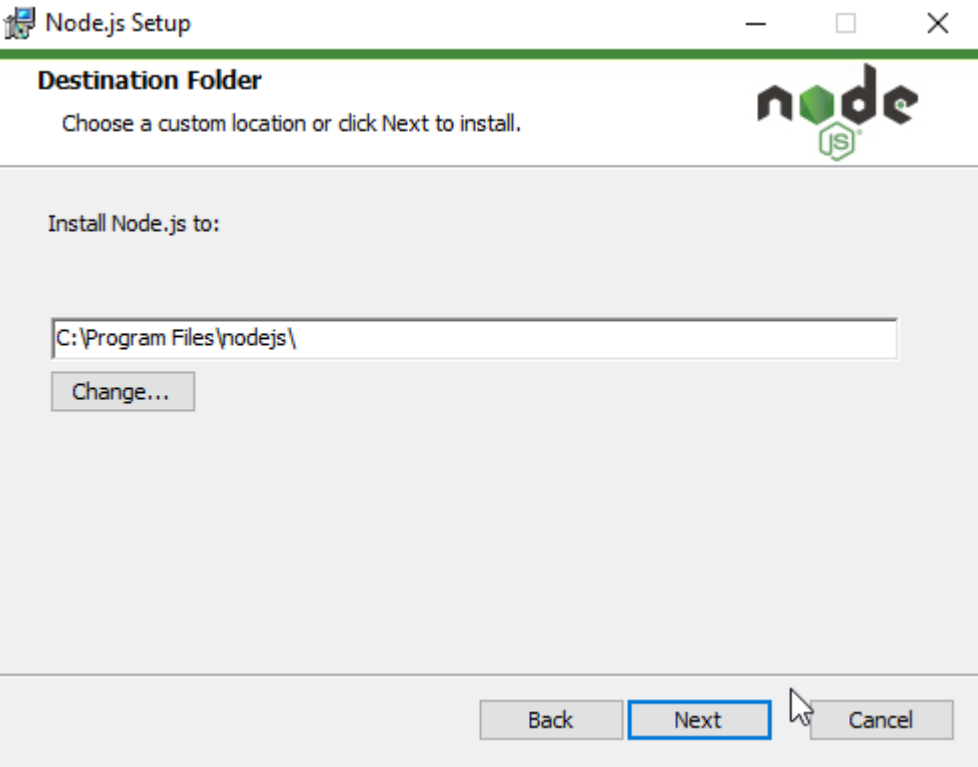
Source Code
node-v12.18.1.tar.gz

Windows Installer (.msi)
Windows Binary (.zip)
macOS Installer (.pkg)
macOS Binary (.tar.gz)
Linux Binaries (x64)
Linux Binaries (ARM)
Source Code

32-bit	64-bit
32-bit	64-bit
64-bit	
64-bit	
64-bit	
ARMv7	ARMv8
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/



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

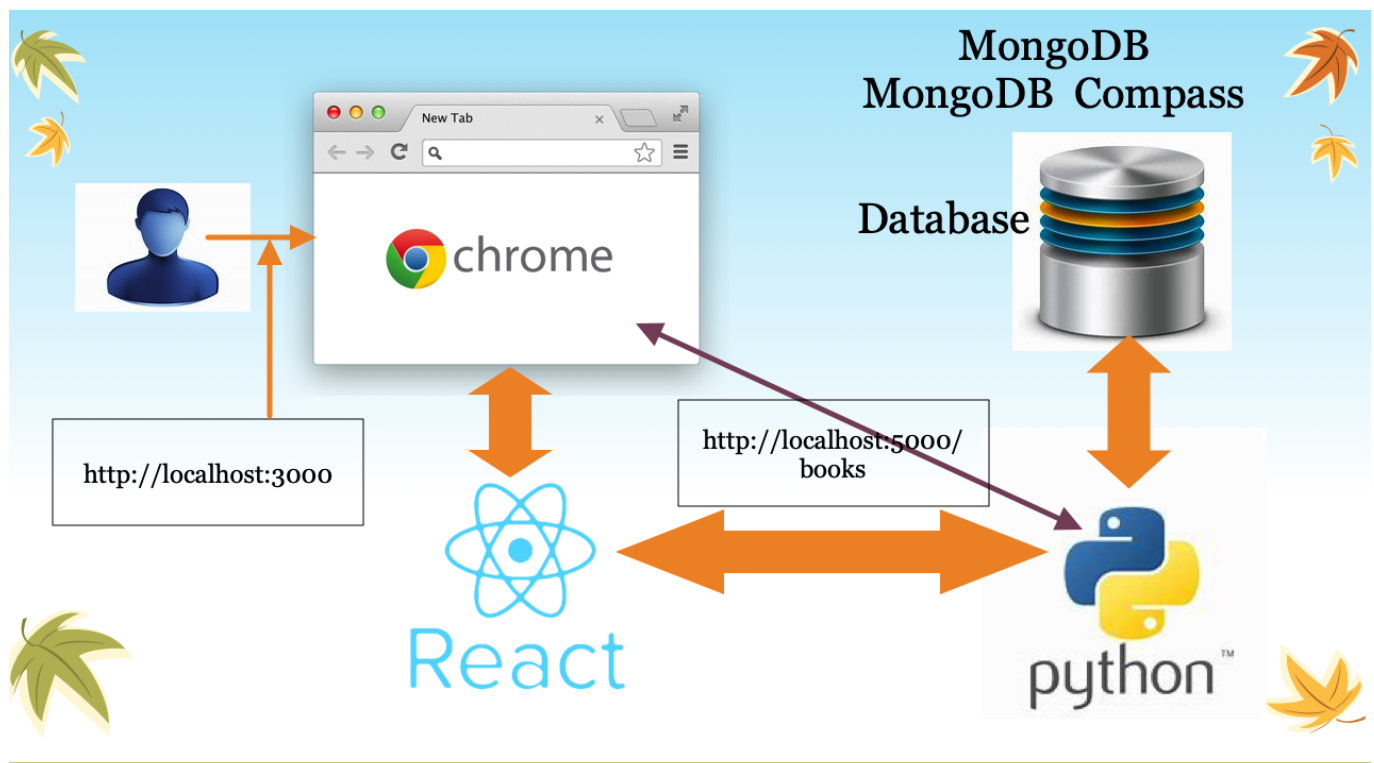
```
sudo npx create-react-app wang-app  
sudo chown -R wangqianjiang wang-app  
cd wang-app  
npm start
```

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

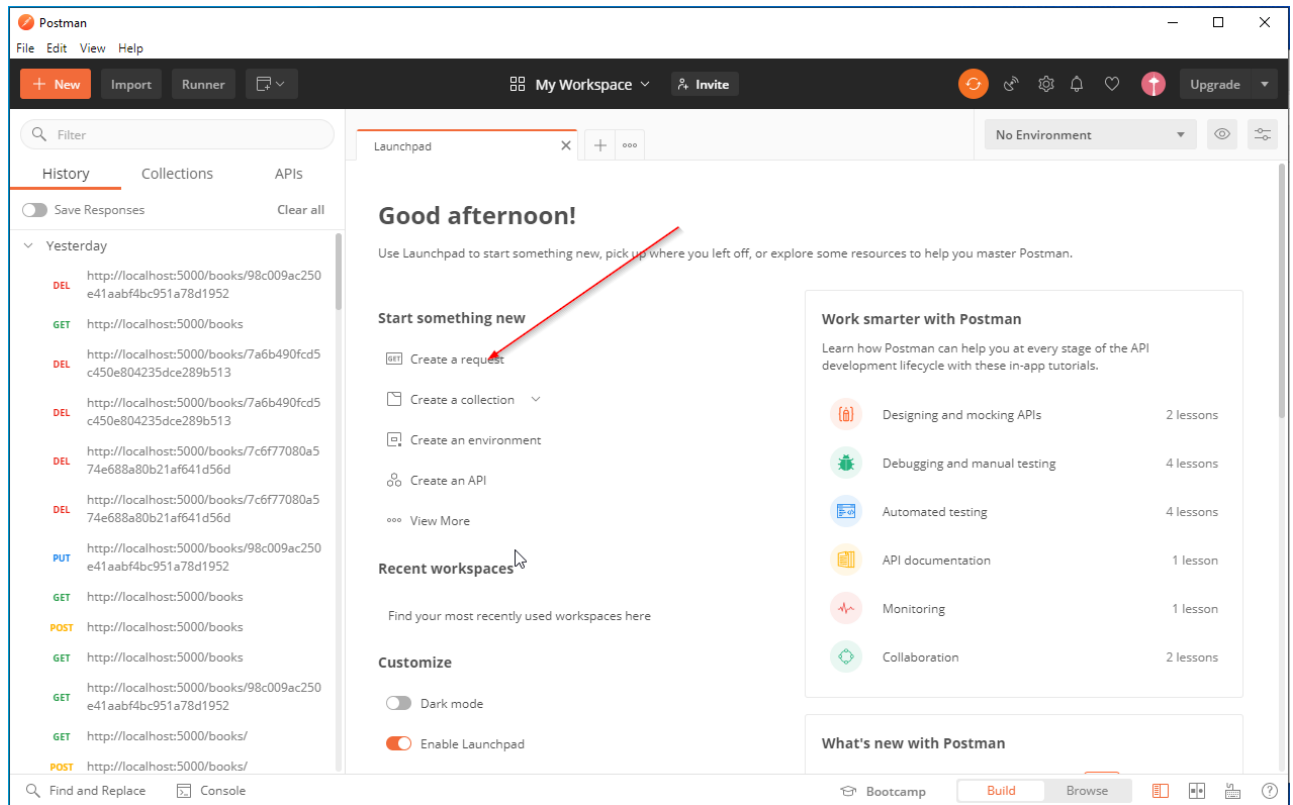
```
cd server
python app.py
cd ../book-app
npm start
```

[Table of Contents](#)

App server

- URL: Uniform Resource Locator
 - <https://www.google.com>
 - Protocol: 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](#)



- 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](#)

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](#)

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 install -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 inheritance testUser
- person, teacher, student inheritance

[YouTube Classes](#)

classes are fundamental tools to any object oriented programming language, think of class as template for creating object and related data and functions that do interesting things with that data. Python makes 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

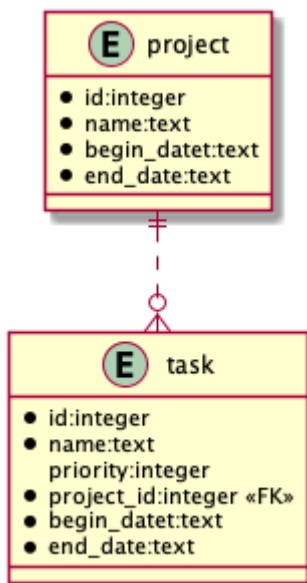
The screenshot shows the DB Browser for SQLite application. The main window displays the results of a SQL query: `SELECT * from tasks`. The results are shown in a table with 9 rows and 7 columns: `id`, `name`, `priority`, `project_id`, `begin_date`, and `end_date`.

	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

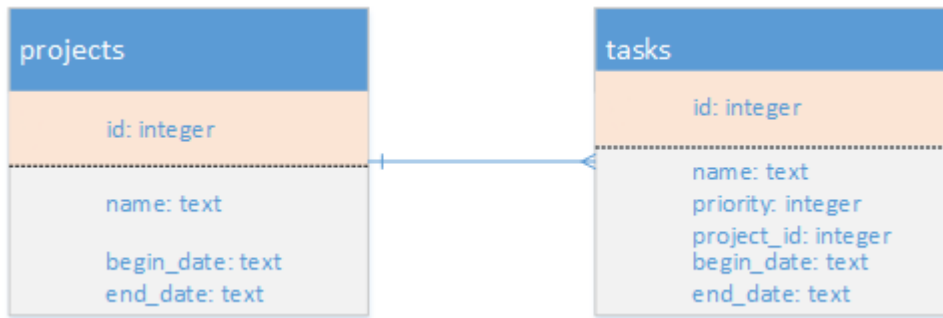
Below the table, the execution status is shown: "Execution finished without errors. Result: 9 rows returned in 15ms. At line 1: SELECT * from tasks".

The right sidebar shows the "Edit Database Cell" panel with the selected cell containing the date "2020-03-28". Below it, the "Remote" panel shows connection options for DBHub.io, Local, and Current Database.

- 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

tkinter GUI

[tkinter documentation](#) [tkinter course](#) [tkinter sample web](#)

Basic Widgets

Buttons

Labels

Frames

CheckBoxes

RadioButtons

Entry

ComboBox

- ☐ tkinter1.py; tkinter.TK() Open window
- ☐ tkinter2.py; add Label
- ☐ tkinter3.py; disable resize window
- ☐ tkinter4.py; Exit Button
- ☐ tkinter5.py; add line to Canvas
- ☐ tkinter6.py; Button with command function
- ☐ tkinter7.py; Combobox
- ☐ tkinter8.py; Checkbutton
- ☐ tkinter9.py; Spinbox
- ☐ tkinter10.py; Text
- ☐ tkinter11.py; Entry
- ☐ tkinter12.py; Radiobutton
- ☐ tkinter13.py; Radiobutton
- ☐ tkinter14.py; ScrolledText

- ☐ tkinter15.py; Progressbar
- ☐ tkinter16.py; Disable button
- ☐ tkinter17.py; Listbox
- ☐ tkinter18.py; Listbox
- ☐ tkinter19.py; Simpliedialog
- ☐ tkinter20.py; Inheritance from tkinter.Frame
- ☐ tkinter21.py; Display resized image in frame
- ☐ tkinter22.py; Select file from file explore
- ☐ tkinter23.py; Read file and display contents in Scrolledtext
- ☐ tkinter24.py; Read file and save contents in Scrolledtext to a file
- ☐ tkinter25.py; Menu
- ☐ tkinter26.py; Layout management: pack()
- ☐ tkinter27.py; Layout management: pack(fill=tk.X)
- ☐ tkinter28.py; Layout management: pack(fill=tk.X, padx=10)
- ☐ tkinter29.py; Layout management: pack(fill=tk.X, pady=10)
- ☐ tkinter30.py; Layout management: pack(fill=tk.X, pady=10, side=tk.LEFT)
- ☐ tkinter31.py; Layout management: place(x, y, width, height)
- ☐ tkinter32.py; Layout management: grid(row, column)
- ☐ tkinter33.py; Button.bind and event [Bind & event](#)
- ☐ tkinter34.py; Mouse position (event.x, event.y)
- ☐ tkinter35.py; Table
- ☐ tkinter36.py; Pie chart
- ☐ tkinter37.py; page switch and animated chart
- ☐ tkinter38.py; Notebook (tabs)
- ☐ tkinter39.py; LabelFrame (titled frame)

Event	Description
Button	A mouse button is pressed with the mouse pointer over the widget. The detail part specifies which button, e.g. The left mouse button is defined by the event "Button-1" the middle button by 'Button-2', and the rightmost mouse button by 'Button-3'. 'Button-4' defines the scroll up event on mice with wheel support and and 'Button-5' the scroll down. If you press down a mouse button over a widget and keep it pressed, Tkinter will automatically "grab" the mouse pointer. Further mouse events like Motion and Release events will be sent to the current widget, even if the mouse is moved outside the current widget. The current position, relative to the widget, of the mouse pointer is provided in the x and y members of the event object passed to the callback. You can use ButtonPress instead of Button, or even leave it out completely: , , and '1' are all synonyms.
Motion	The mouse is moved with a mouse button being held down. To specify the left, middle or right mouse button use , , and respectively. The current position of the mouse pointer is provided in the x and y members of the event object passed to the callback, i.e. event.x, event.y
ButtonRelease	Event, if a button is released. To specify the left, middle or right mouse button use 'ButtonRelease-1', 'ButtonRelease-2', and 'ButtonRelease-3' respectively. The current position of the mouse pointer is provided in the x and y members of the event object passed to the callback, i.e. event.x, event.y

Double-Button	Similar to the Button event, see above, but the button is double clicked instead of a single click. To specify the left, middle or right mouse button use 'Double-Button-1', 'Double-Button-2', and 'Double-Button-3' respectively. You can use Double or Triple as prefixes. Note that if you bind to both a single click ('Button-1') and a double click ('Double-Button-1'), both bindings will be called.
Enter	The mouse pointer entered the widget. Attention: This doesn't mean that the user pressed the Enter key!. is used for this purpose.
Leave	The mouse pointer left the widget.
FocusIn	Keyboard focus was moved to this widget, or to a child of this widget.
FocusOut	Keyboard focus was moved from this widget to another widget.
Return	The user pressed the Enter key. You can bind to virtually all keys on the keyboard: The special keys are Cancel (the Break key), BackSpace, Tab, Return(the Enter key), Shift_L (any Shift key), Control_L (any Control key), Alt_L (any Alt key), Pause, Caps_Lock, Escape, Prior (Page Up), Next (Page Down), End, Home, Left, Up, Right, Down, Print, Insert, Delete, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Num_Lock, and Scroll_Lock.
Return	The user pressed the Enter key. You can bind to virtually all keys on the keyboard: The special keys are Cancel (the Break key), BackSpace, Tab, Return(the Enter key), Shift_L (any Shift key), Control_L (any Control key), Alt_L (any Alt key), Pause, Caps_Lock, Escape, Prior (Page Up), Next (Page Down), End, Home, Left, Up, Right, Down, Print, Insert, Delete, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Num_Lock, and Scroll_Lock.
Key	The user pressed any key. The key is provided in the char member of the event object passed to the callback (this is an empty string for special keys).
a	The user typed an "a" key. Most printable characters can be used as is. The exceptions are space ('space') and less than ('less'). Note that 1 is a keyboard binding, while '1' is a button binding.
Shift-Up	The user pressed the Up arrow, while holding the Shift key pressed. You can use prefixes like Alt, Shift, and Control.
Configure	The size of the widget changed. The new size is provided in the width and height attributes of the event object passed to the callback. On some platforms, it can mean that the location changed.

OpenGL

Install PyOpenGL modules

```
pip install PyOpenGL PyOpenGL_accelerate
```

OpenGL Tutorial

-  opengl1.py; rotate Cube with edge lines only

- ☐ opengl2.py; rotate Cube with colored surface and lines
- ☐ opengl3.py; move Cube by arrow keys or mouse wheel
- ☐ opengl4.py; move Cube by arrow keys or mouse wheel
- ☐ opengl5.py; move Cube by arrow keys or mouse wheel
- ☐ opengl6.py; move Cube by arrow keys or mouse wheel