Scientific Programming and Visualization

(MSDM 5002-Fall 2023)

Instructor: Junwei Liu

Office: 4474 Tel: 2358 7971

Email: liuj@ust.hk

Lec: Fri 07:00pm - 09:50pm

Room 2464

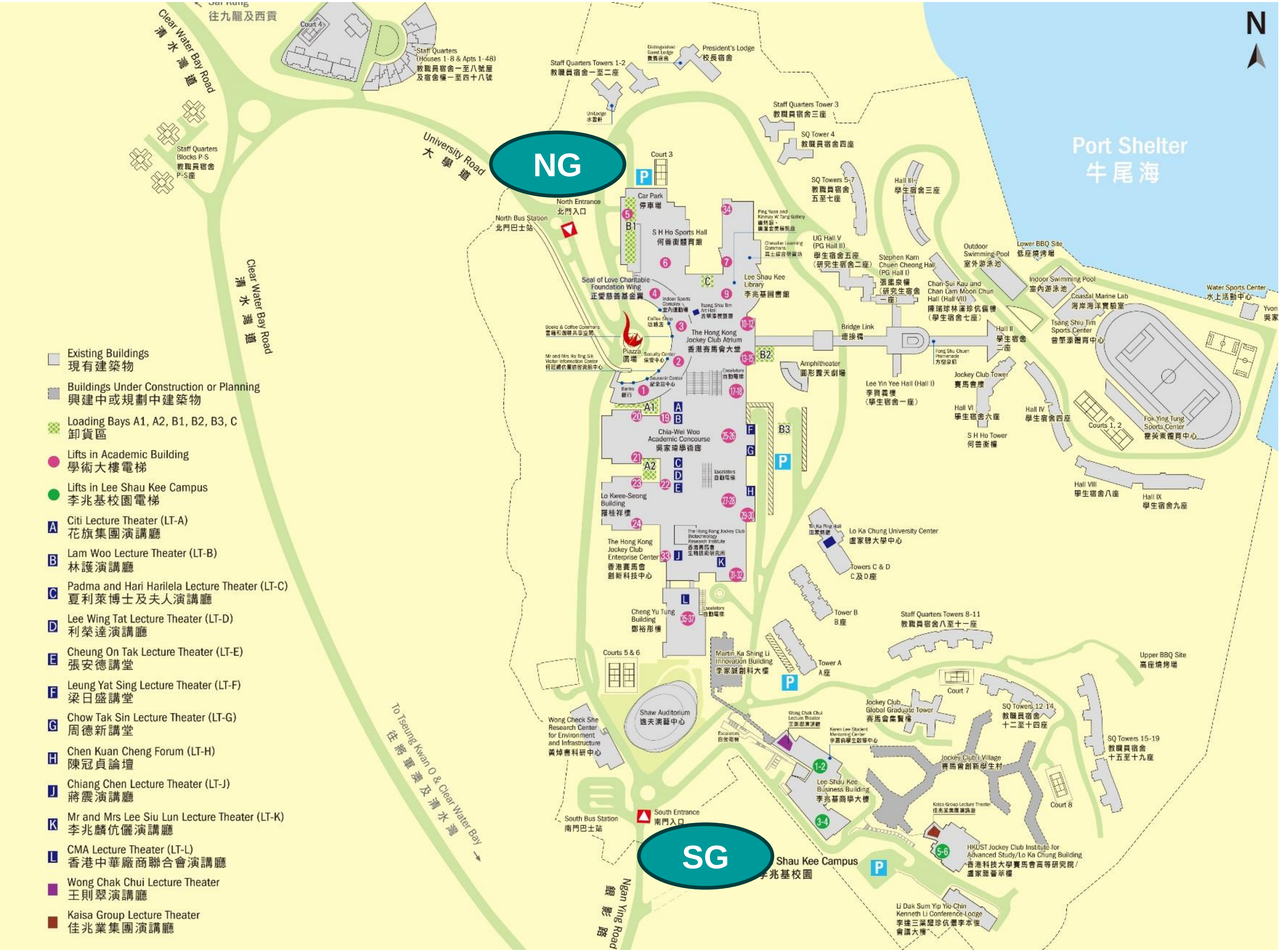
Sep 5, 2023

Instructional Assistant

SHING Ming Tony

Email: **shingm@ust.hk**

Office Hour: Meeting upon appointment



□ Existing Buildings
現有建築物

▨ Buildings Under Construction or Planning
興建中或規劃中建築物

▨ Loading Bays A1, A2, B1, B2, B3, C
卸貨區

● Lifts in Academic Building
學術大樓電梯

● Lifts in Lee Shau Kee Campus
李兆基校園電梯

A Citi Lecture Theater (LT-A)
花旗集團演講廳

B Lam Woo Lecture Theater (LT-B)
林護演講廳

C Padma and Hari Harilela Lecture Theater (LT-C)
夏利萊博士及夫人演講廳

D Lee Wing Tat Lecture Theater (LT-D)
利榮達演講廳

E Cheung On Tak Lecture Theater (LT-E)
張安德講堂

F Leung Yat Sing Lecture Theater (LT-F)
梁日盛講堂

G Chow Tak Sin Lecture Theater (LT-G)
周德新講堂

H Chen Kuan Cheng Forum (LT-H)
陳冠貞論壇

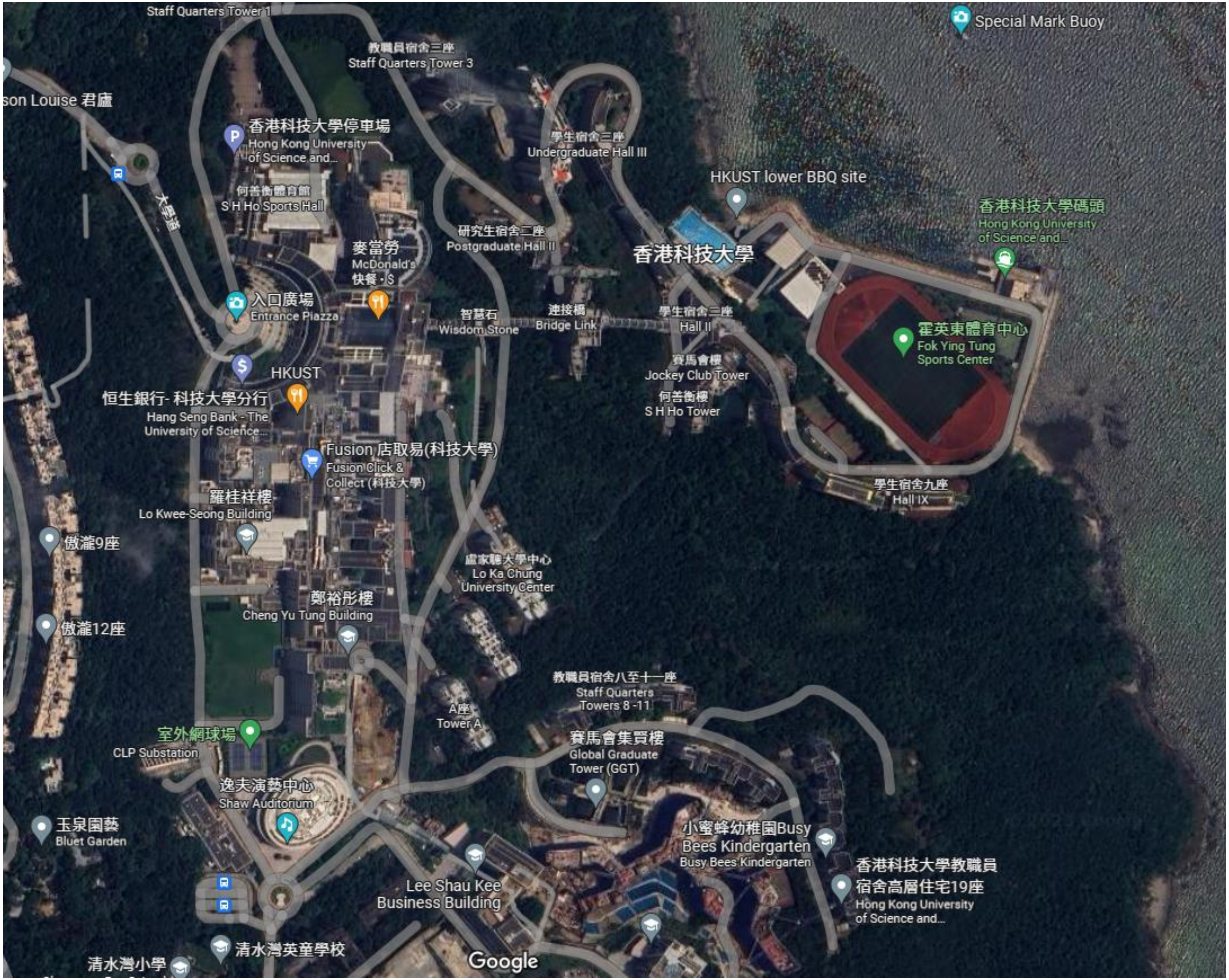
J Chiang Chen Lecture Theater (LT-J)
蔣震演講廳

K Mr and Mrs Lee Siu Lun Lecture Theater (LT-K)
李兆麟伉儷演講廳

L CMA Lecture Theater (LT-L)
香港中華廠商聯合會演講廳

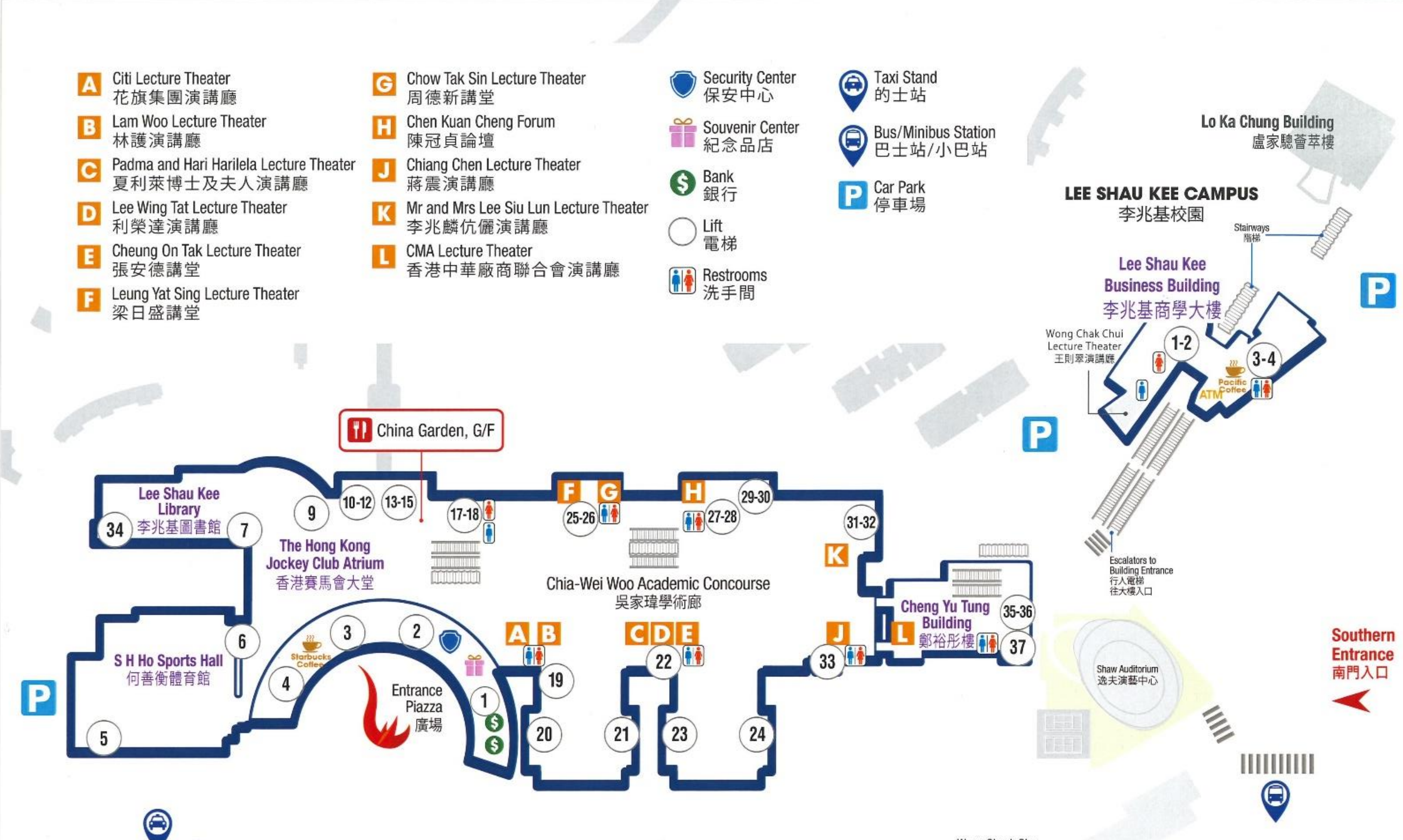
■ Wong Chak Chui Lecture Theater
王則翠演講廳

■ Kaisa Group Lecture Theater
佳兆業集團演講廳


Staff Quarters Tower 1
教職員宿舍三座
Staff Quarters Tower 3
Special Mark Buoy
son Louise 君廬
香港科技大學停車場
Hong Kong University of Science and...
學生宿舍三座
Undergraduate Hall III
HKUST lower BBQ site
大學道
何善衡體育館
S H Ho Sports Hall
研究生宿舍二座
Postgraduate Hall II
香港科技大學
香港科技大學碼頭
Hong Kong University of Science and...
麥當勞
McDonald's
快餐 · $
入口廣場
Entrance Piazza
智慧石
Wisdom Stone
連接橋
Bridge Link
學生宿舍二座
Hall II
霍英東體育中心
Fok Ying Tung Sports Center
HKUST
賽馬會樓
Jockey Club Tower
恒生銀行- 科技大學分行
Hang Seng Bank - The University of Science...
何善衡樓
S H Ho Tower
Fusion 店取易(科技大學)
Fusion Click & Collect (科技大學)
學生宿舍九座
Hall IX
羅桂祥樓
Lo Kwee-Seong Building
傲瀧9座
盧家驄大學中心
Lo Ka Chung University Center
鄭裕彤樓
Cheng Yu Tung Building
傲瀧12座
教職員宿舍八至十一座
Staff Quarters Towers 8 -11
A座
Tower A
室外網球場
賽馬會集賢樓
Global Graduate Tower (GGT)
CLP Substation
逸夫演藝中心
Shaw Auditorium
小蜜蜂幼稚園Busy Bees Kindergarten
Busy Bees Kindergarten
玉泉園藝
Bluet Garden
香港科技大學教職員宿舍高層住宅19座
Hong Kong University of Science and...
Lee Shau Kee Business Building
清水灣英童學校
清水灣小學
Google

A	Citi Lecture Theater 花旗集團演講廳
B	Lam Woo Lecture Theater 林護演講廳
C	Padma and Hari Harilela Lecture Theater 夏利萊博士及夫人演講廳
D	Lee Wing Tat Lecture Theater 利榮達演講廳
E	Cheung On Tak Lecture Theater 張安德講堂
F	Leung Yat Sing Lecture Theater 梁日盛講堂
G	Chow Tak Sin Lecture Theater 周德新講堂
H	Chen Kuan Cheng Forum 陳冠貞論壇
J	Chiang Chen Lecture Theater 蔣震演講廳
K	Mr and Mrs Lee Siu Lun Lecture Theater 李兆麟伉儷演講廳
L	CMA Lecture Theater 香港中華廠商聯合會演講廳

🛡	Security Center 保安中心
🎁	Souvenir Center 紀念品店
$	Bank 銀行
○	Lift 電梯
🚻	Restrooms 洗手間
🚕	Taxi Stand 的士站
🚌	Bus/Minibus Station 巴士站/小巴站
P	Car Park 停車場



Wong Check She
Research Center for
Environment and
Infrastructure
黃焯書科研中心

Bus 巴士：

91M	.... Po Lam 寶林
91	.... Clear Water Bay 清水灣
792M	..... Tseung Kwan O / Sai Kung 將軍澳 / 西貢

Minibus 小巴：

11/11M	... Hang Hau 坑口
12	.... Po Lam / Sai Kung 寶林 / 西貢

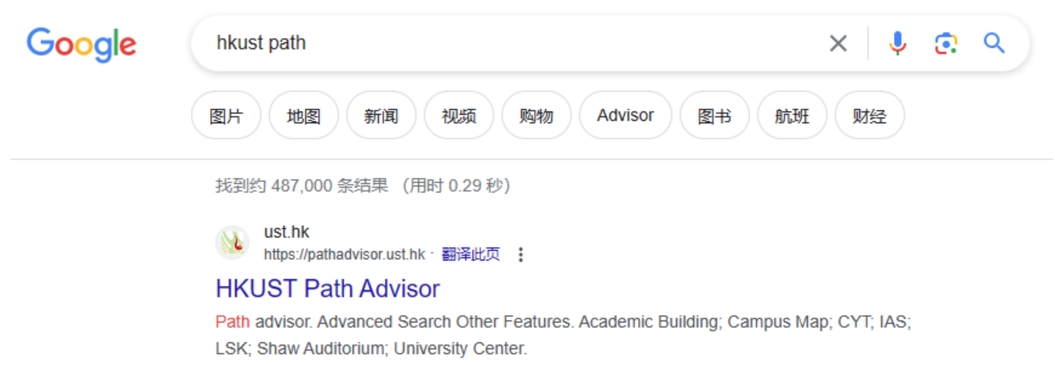
Northern Entrance
北門入口

Bus 巴士：

91/91M	... Diamond Hill 鑽石山
91P	.... Choi Hung 彩虹

Minibus 小巴：

11	.... Choi Hung 彩虹
104	.... Ngau Tau Kok 牛頭角



hkust path

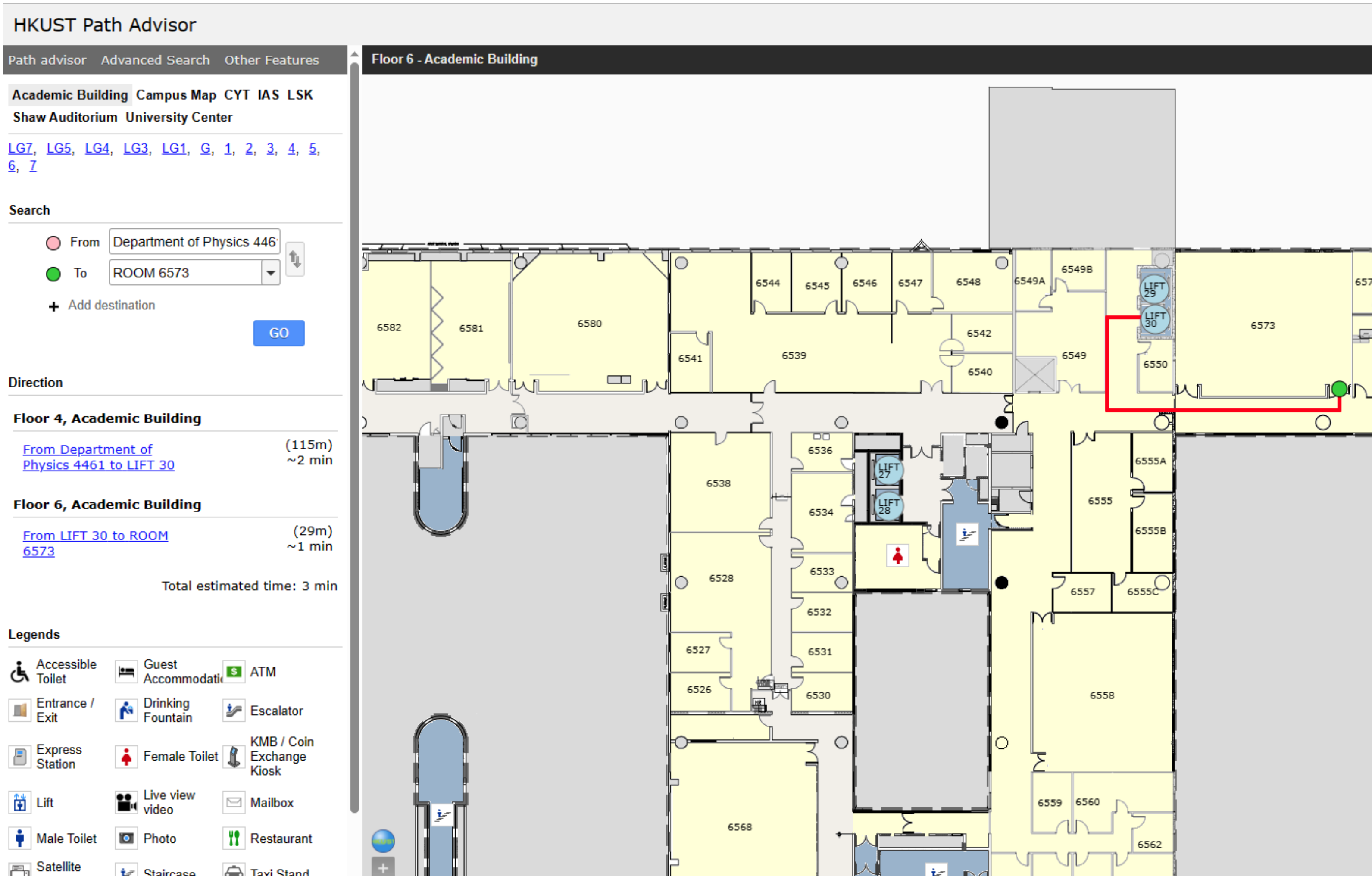
图片 地图 新闻 视频 购物 Advisor 图书 航班 财经

找到约 487,000 条结果 （用时 0.29 秒）

ust.hk
https://pathadvisor.ust.hk · 翻译此页

HKUST Path Advisor

Path advisor. Advanced Search Other Features. Academic Building; Campus Map; CYT; IAS; LSK; Shaw Auditorium; University Center.



HKUST Path Advisor

Path advisor Advanced Search Other Features

Academic Building Campus Map CYT IAS LSK
Shaw Auditorium University Center

LG7, LG5, LG4, LG3, LG1, G, 1, 2, 3, 4, 5,
6, 7

Search

From: Department of Physics 446

To: ROOM 6573

+ Add destination

GO

Direction

Floor 4, Academic Building

From Department of Physics 4461 to LIFT 30 (115m) ~2 min

Floor 6, Academic Building

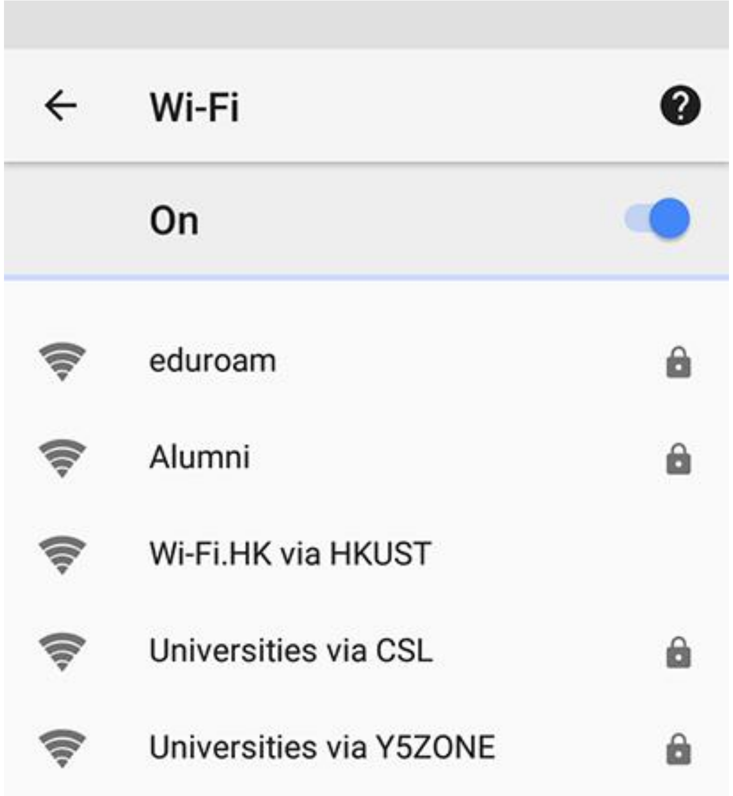
From LIFT 30 to ROOM 6573 (29m) ~1 min

Total estimated time: 3 min

Legends

Accessible Toilet
Guest Accommodati
ATM
Entrance / Exit
Drinking Fountain
Escalator
Express Station
Female Toilet
KMB / Coin Exchange Kiosk
Lift
Live view video
Mailbox
Male Toilet
Photo
Restaurant
Satellite
Staircase
Taxi Stand

Floor 6 - Academic Building



Wi-Fi

On

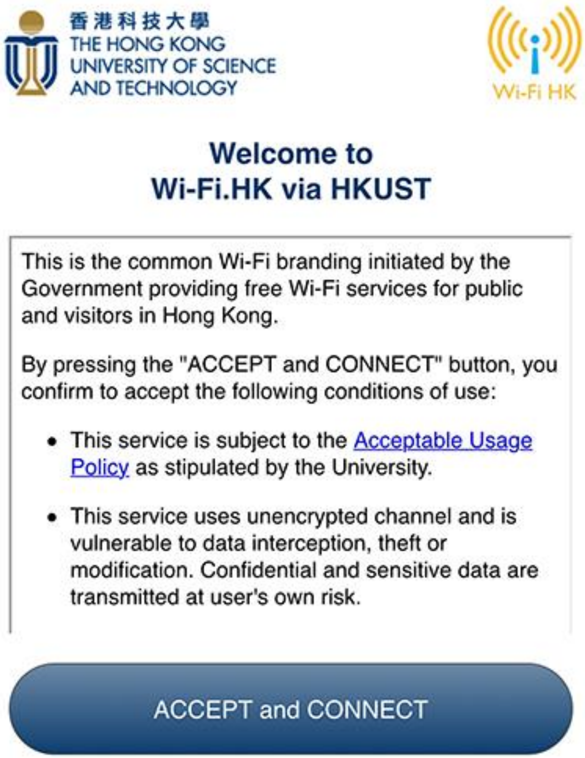
eduroam

Alumni

Wi-Fi.HK via HKUST

Universities via CSL

Universities via Y5ZONE



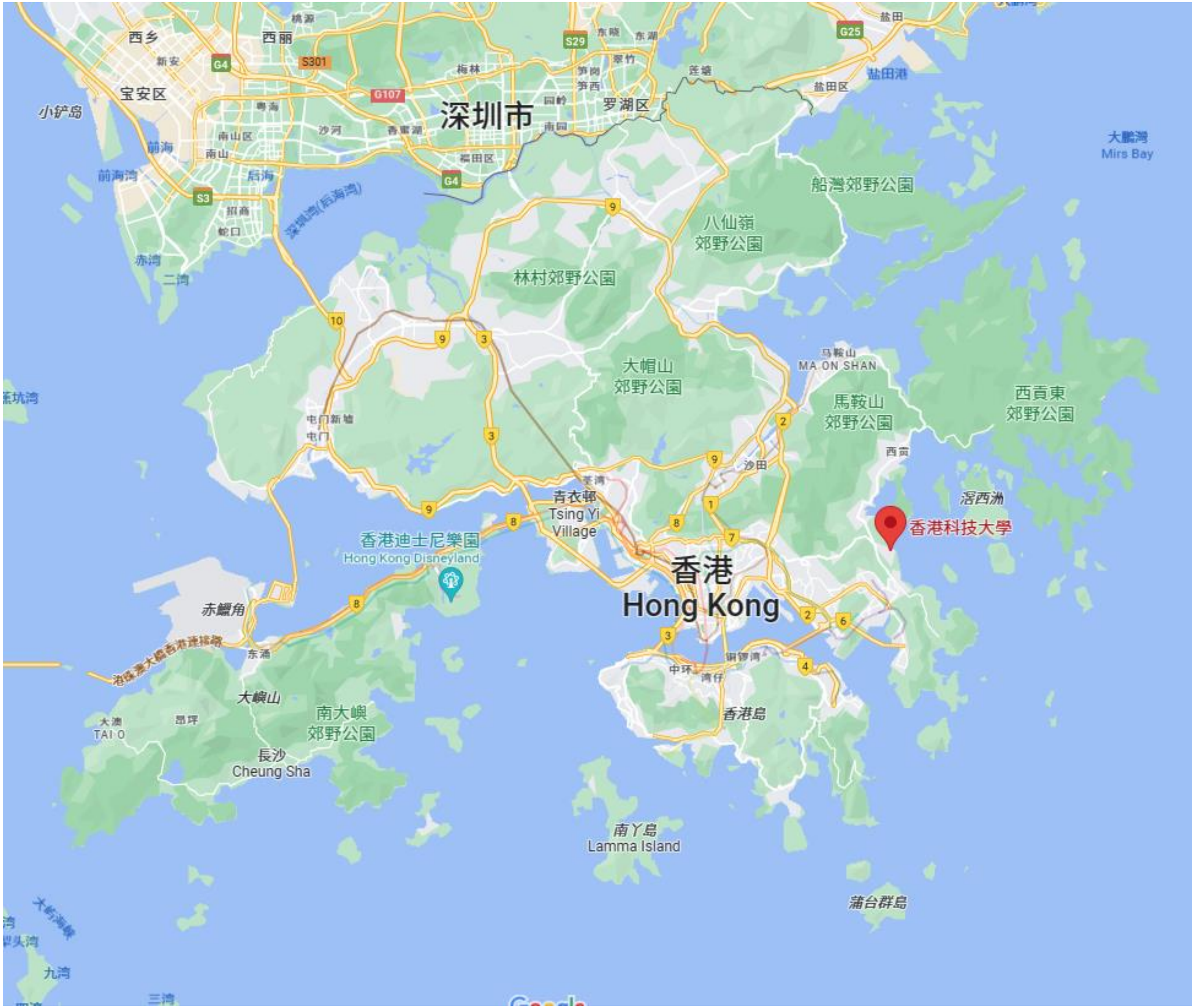
Welcome to
Wi-Fi.HK via HKUST

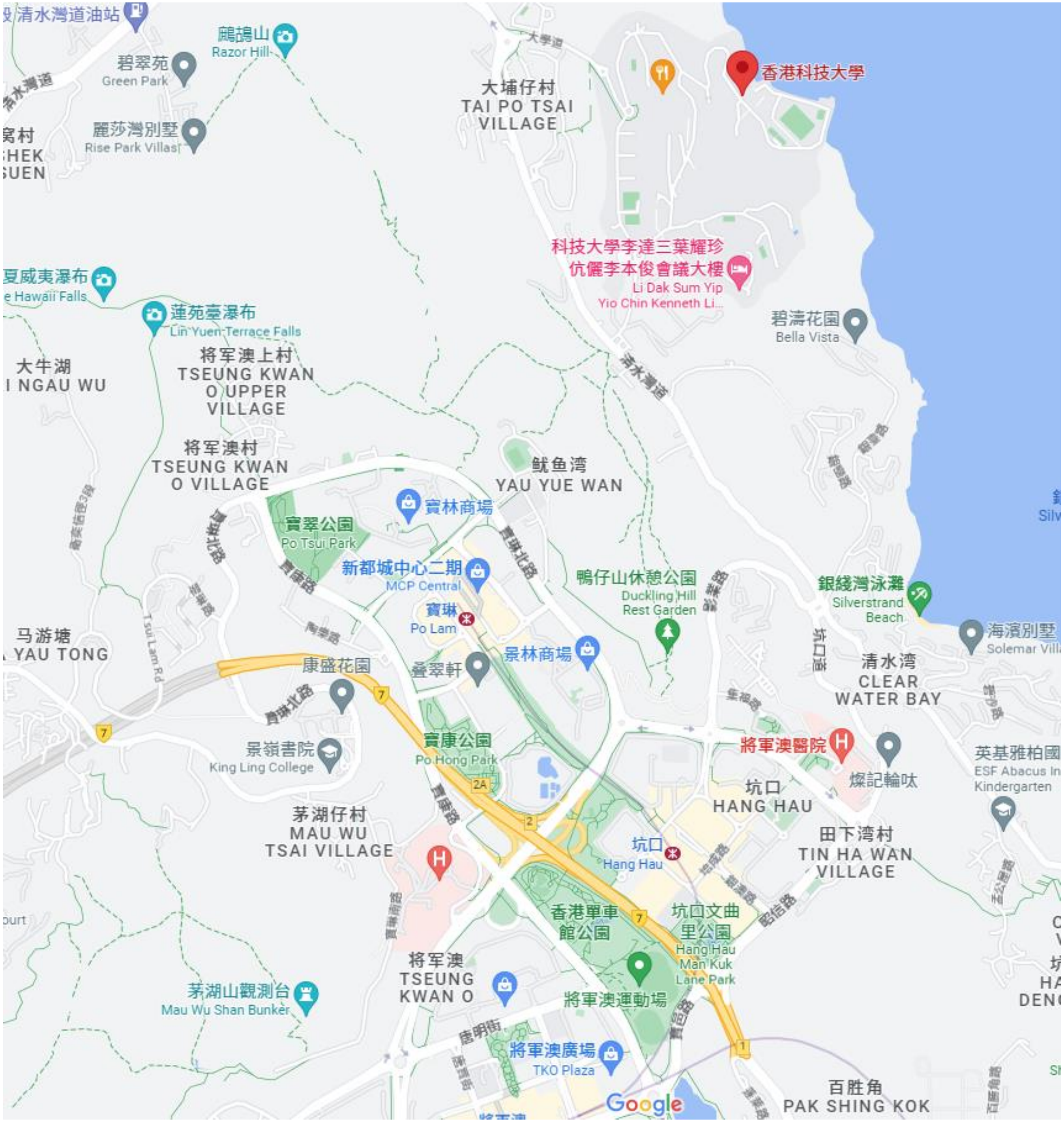
This is the common Wi-Fi branding initiated by the Government providing free Wi-Fi services for public and visitors in Hong Kong.

By pressing the "ACCEPT and CONNECT" button, you confirm to accept the following conditions of use:

- This service is subject to the Acceptable Usage Policy as stipulated by the University.
- This service uses unencrypted channel and is vulnerable to data interception, theft or modification. Confidential and sensitive data are transmitted at user's own risk.

ACCEPT and CONNECT


深圳市
西乡
西丽
宝安区
新安
小铲岛
南山区
南山
前海
前海湾
后海
招商
蛇口
赤湾
二湾
深圳湾(后海湾)
粤海
沙河
香蜜湖
福田区
梅林
罗湖区
笋岗
笋西
翠竹
东晓
东湖
莲塘
盐田
盐田区
盐田港
大鹏湾
Mirs Bay
船灣郊野公園
八仙嶺
郊野公園
林村郊野公園
大帽山
郊野公園
马鞍山
MA ON SHAN
馬鞍山
郊野公園
西貢東
郊野公園
西貢
沙田
滘西洲
香港科技大學
荃湾
青衣邨
Tsing Yi
Village
屯门新墟
屯门
香港迪士尼樂園
Hong Kong Disneyland
香港
Hong Kong
赤鱲角
港珠澳大橋香港連接路
东涌
大嶼山
大澳
TAI O
昂坪
南大嶼
郊野公園
長沙
Cheung Sha
中环
湾仔
铜锣湾
香港島
南丫島
Lamma Island
蒲台群島
九湾
三湾
G4
G25
S29
S301
G107
S3
10
9
8
3
2
1
7
6
4
Google



- North gate: 11 to Hang Hau (坑口) and subway
- South gate: 11, 91 etc to Choi Hung (彩虹) and subway
- Subway
- Octopus card

Course description

Brief Information/synopsis:

In this course, the students will learn Python and are required to use Python to solve some scientific and real-life problems. Meanwhile, the students will learn different packages and tools to deal with different types of data. By doing this, they are expected to know how to choose the suitable tools for different types of problems. In addition, the students will know how to visualize the data generated by scientific computing or the data in daily life in suitable ways.

Key Topics:

- Basic programming skill in Python
- Basic data visualization methods
- Typical packages and tools dealing with data

Intended Learning Outcomes

Upon successful completion of this course, students should be able to:

1. Master Python
2. Collect, clean and organize data in different format
3. Choose suitable methods (tables, figures or animations) to represent data clearly
4. Make animation to show the evolution of data

Assessment Scheme

Assessment	Assessing Course ILOs
5% by attendance	
55% by assignments including quizzes	1-4
40% by the final project	1-4

Learning Resources

A. Adopted References and Textbook:

- https://www.python.org/doc/
- https://leetcode.com/
- Google
- many other websites

B. Lecture Notes and Course Homepage - http://canvas.ust.hk

Rough Course Schedule

- Lecture 01-02: Python basics: strings, numbers, operators, variables, program structure, control flow, data structure, etc.
- Lecture 03: Python: functions, modules and packages
- Lecture 04: Accuracy and Speed (Usage of Array)
- Lecture 05: Simple applications
- Lecture 06-09: Graphics: graphs, scatter plots, density plots, 3D graphics and Animation, etc
- Lecture 10: Web scraping
- Lecture 11-12: Pandas
- Lecture 13 (optional): Overall review of the course + Projects

General principle

The Simpler The Better

Keep Improving

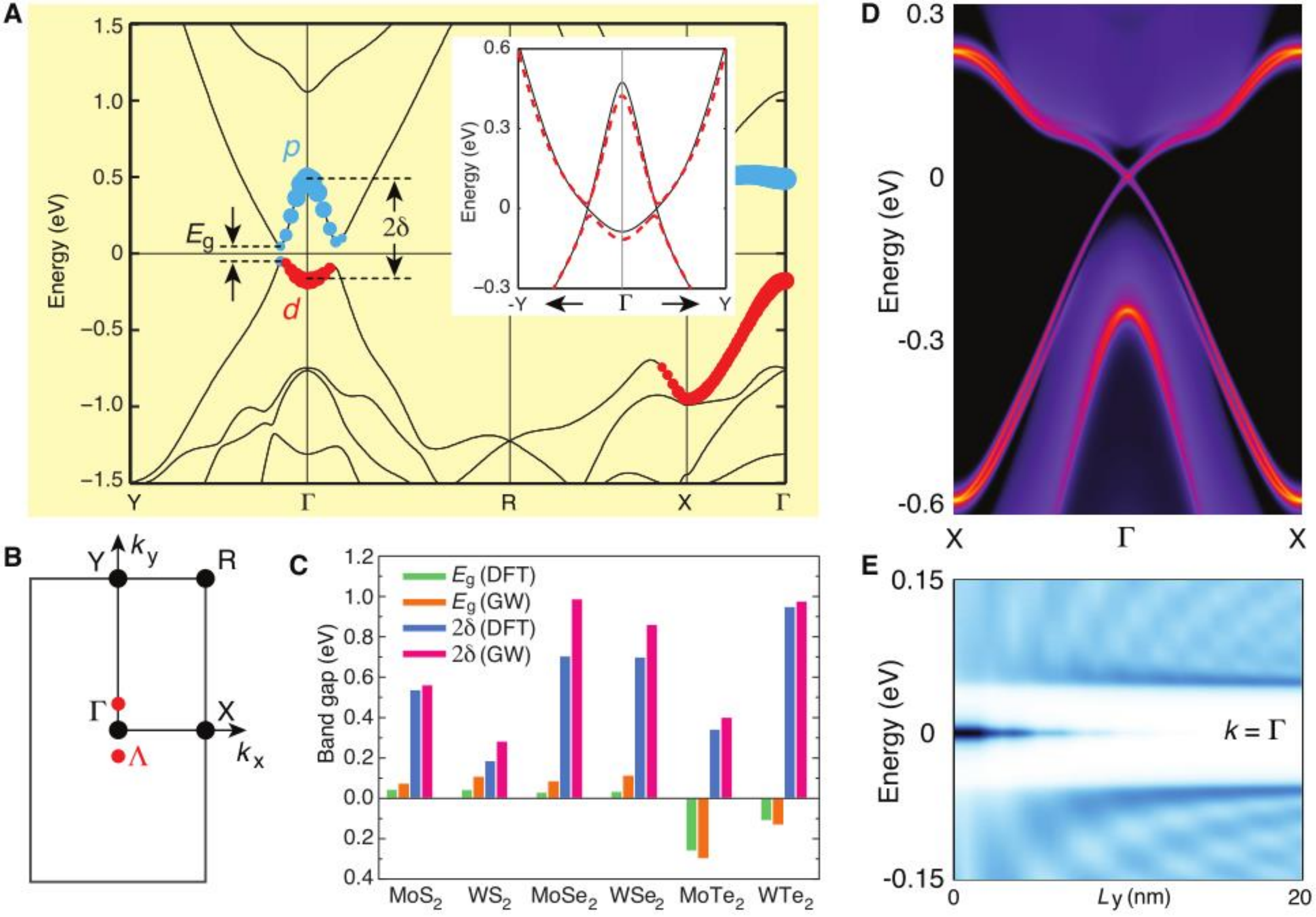
What is computational science?

- There are three types of science based on how the problems are solved in general. Experimental, theoretical, and computational.

- Computation is very important. The science we learn before focuses on the fundamental theories, illustrated with examples, whose solution is almost always possible using nothing more than a pen, a sheet of paper. However, this is not how physics is done in real world.

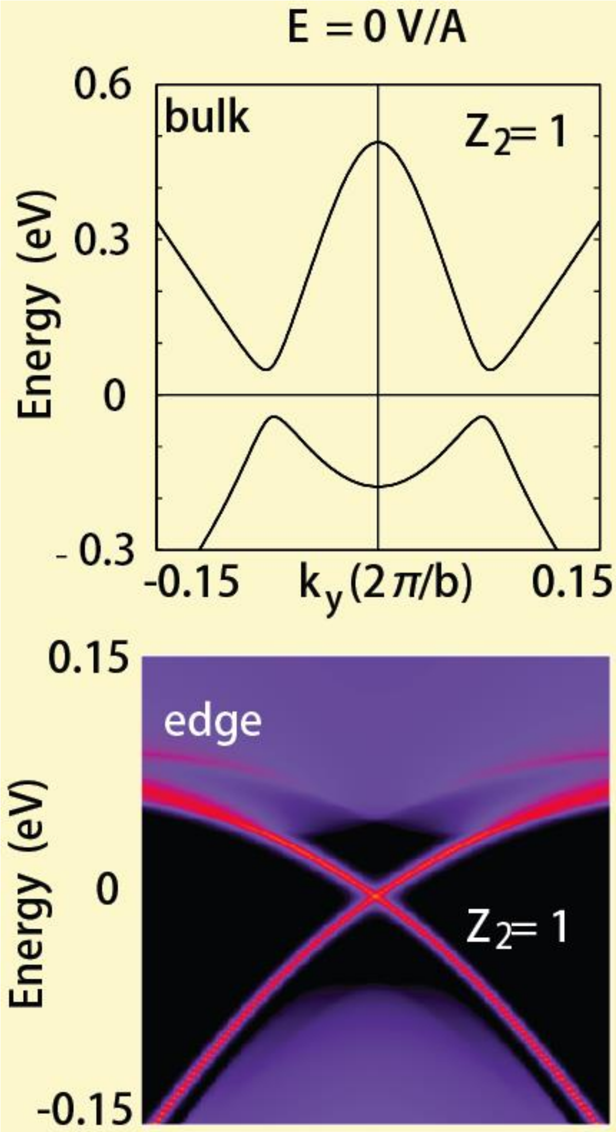
- Most numerical calculations in science fall into one of several general categories, based on the mathematical operations that their solution requires. For example, **integral and derivatives**, **linear algebra tasks** such as matrix inversion or the calculation of eigenvalues, and the solution of **differential equations**, including both ordinary and partial differential equations.

1. Quantum spin Hall effect in 2D materials

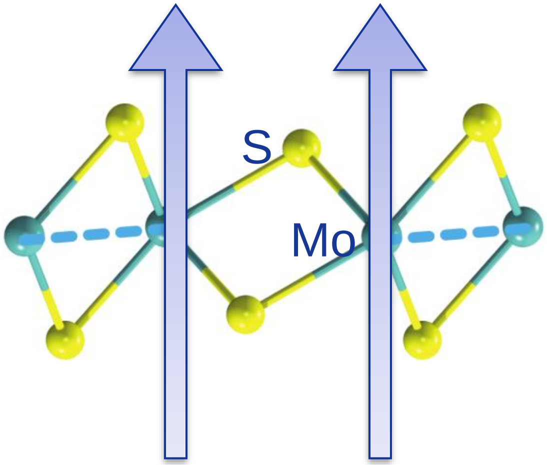


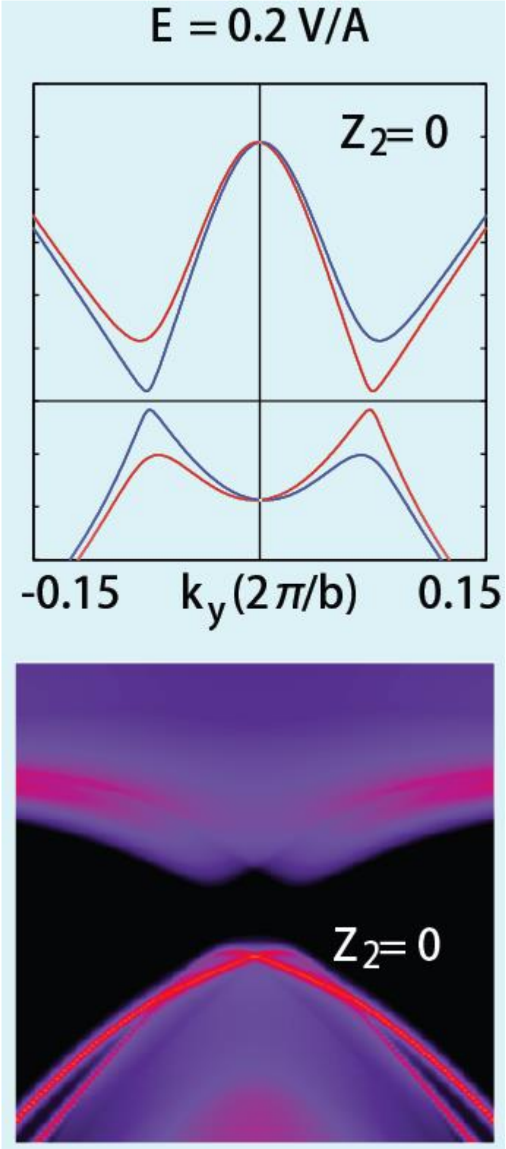
X. Qian, **J. Liu*,** et. al* ***Science*** *346, 1344 (2014)*

Electric tunable - on/off states

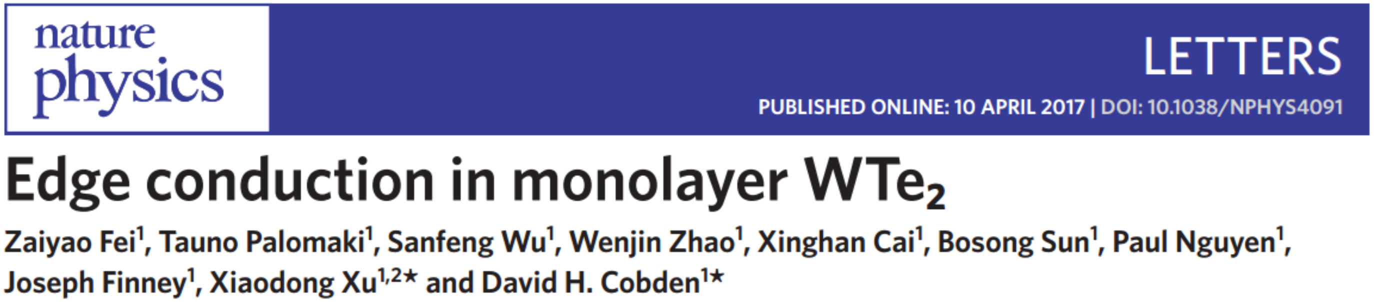


Electric Field





Transport experiments



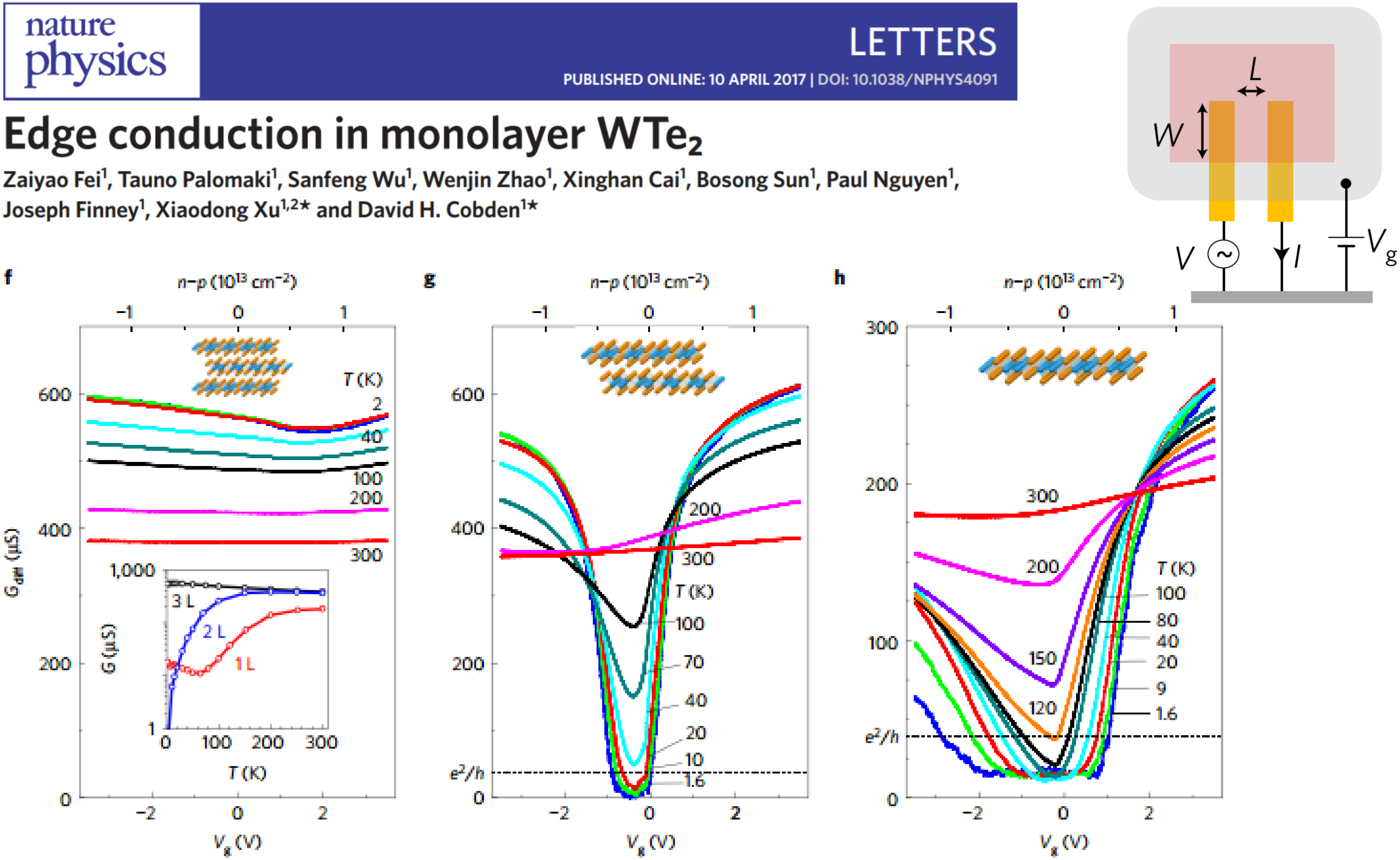
nature physics

LETTERS

PUBLISHED ONLINE: 10 APRIL 2017 | DOI: 10.1038/NPHYS4091

Edge conduction in monolayer WTe_2

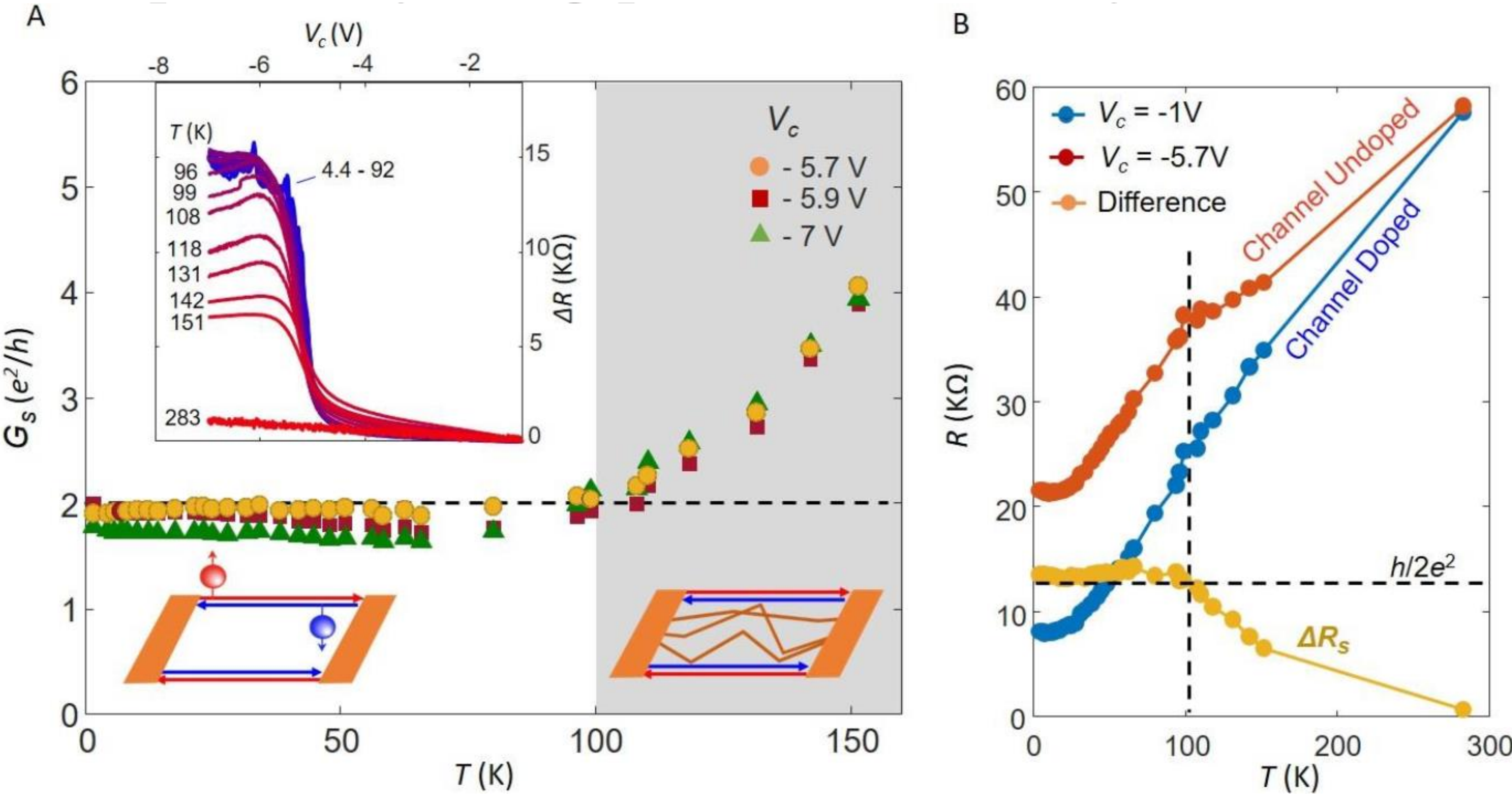
Zaiyao Fei[1], Tauno Palomaki[1], Sanfeng Wu[1], Wenjin Zhao[1], Xinghan Cai[1], Bosong Sun[1], Paul Nguyen[1], Joseph Finney[1], Xiaodong Xu[1,2]* and David H. Cobden[1]*



Z. Fei, et al Nat. Phys. 13, 677 (2017)

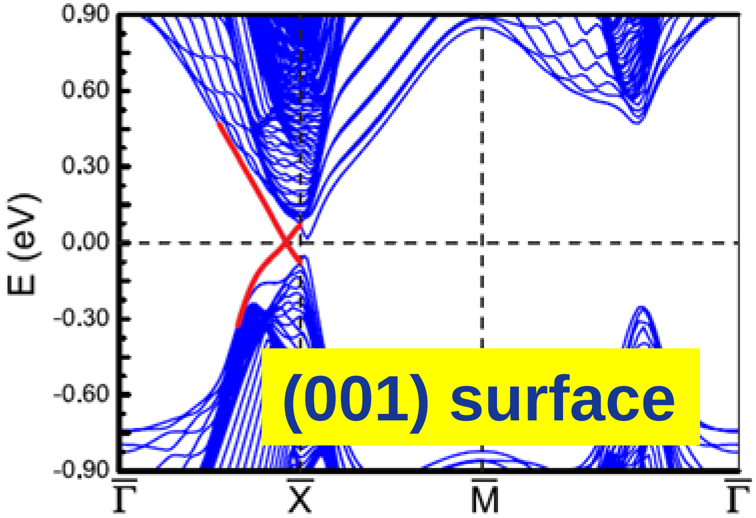
Quantized edge conductance up to 100 K

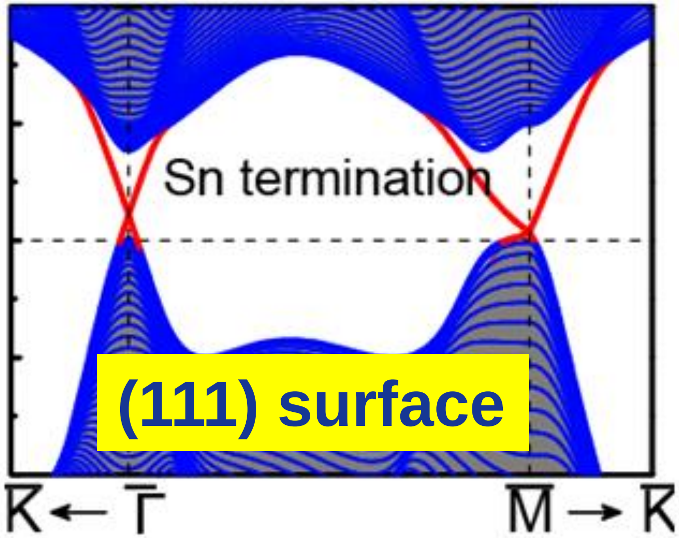
Science



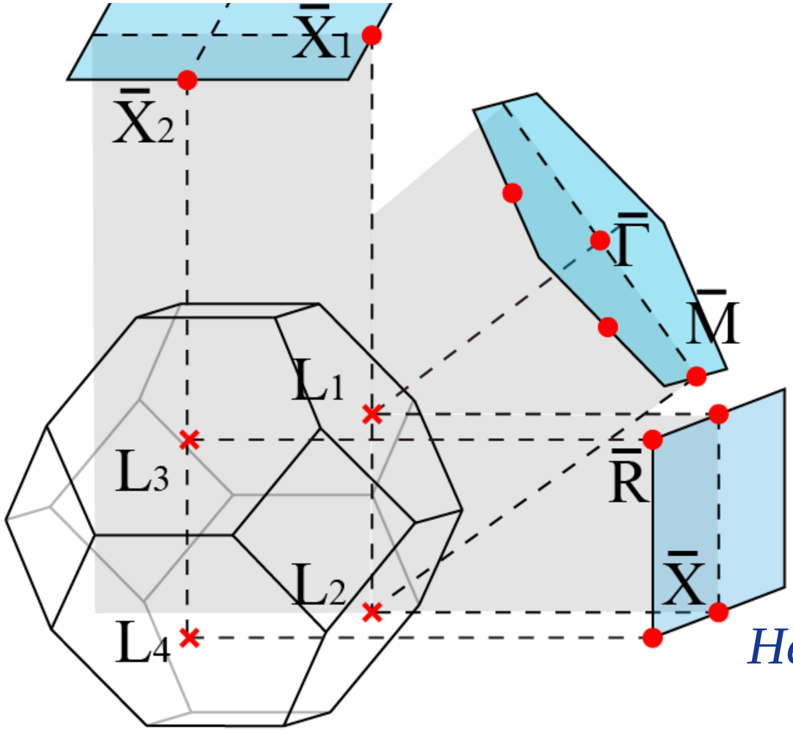
Sanfeng Wu et al, Science 359, 76-79 (2018)

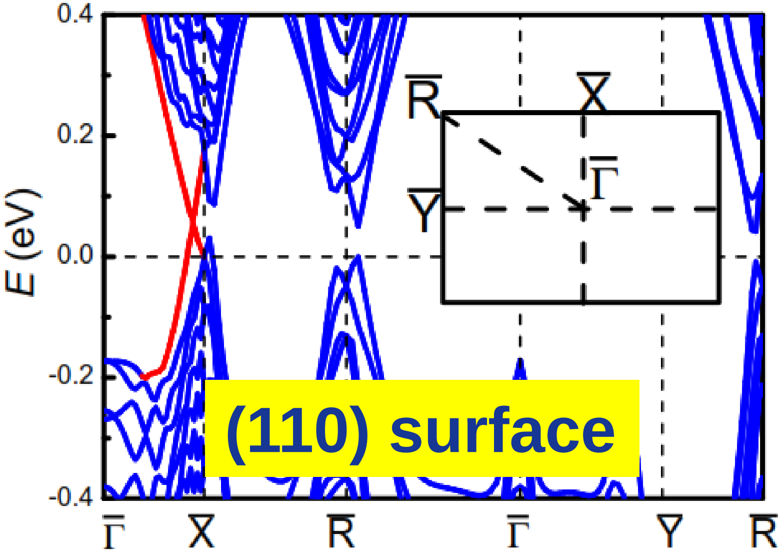
2. Two types of surface states of 3D TCI





Type-II
(111)



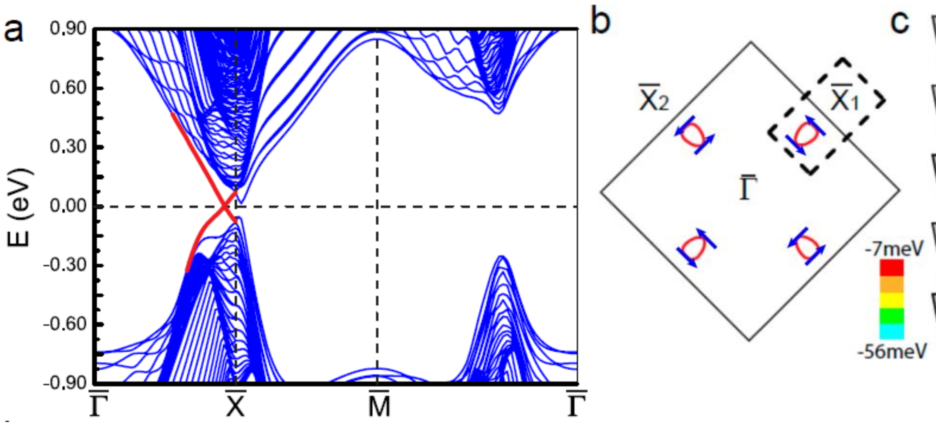


Type-I
(001)
(110)

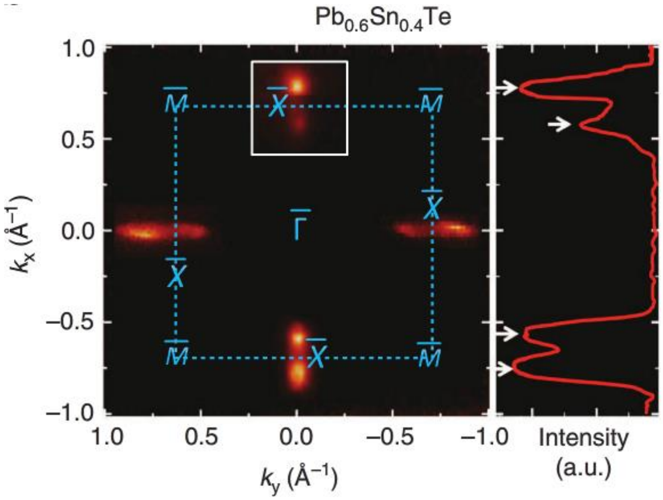
*Heish, Hsin, **J. Liu** et al, **Nat. Commun**. 3, 982 (2012)*

***J. Liu**, et al , **PRB** 88, 241303(R) (2013)*

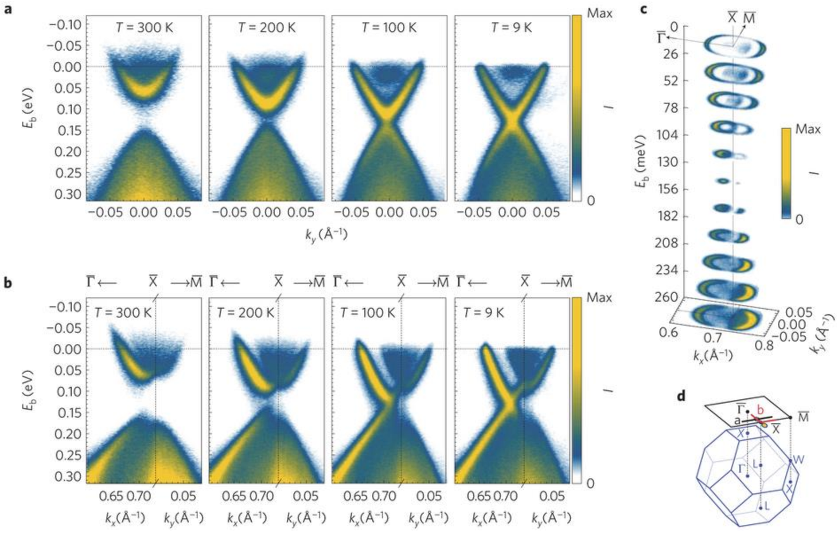
Experiments of $Pb_{1-x}Sn_xTe$ (001) surface



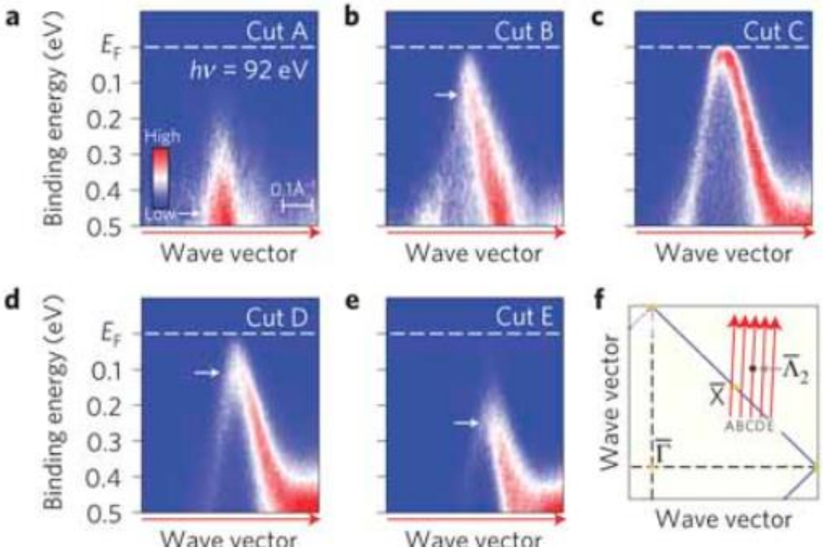
T. Hsieh et al. Nat. Comm. (2012)



S. Xu et al. Nat. Comm. (2012)

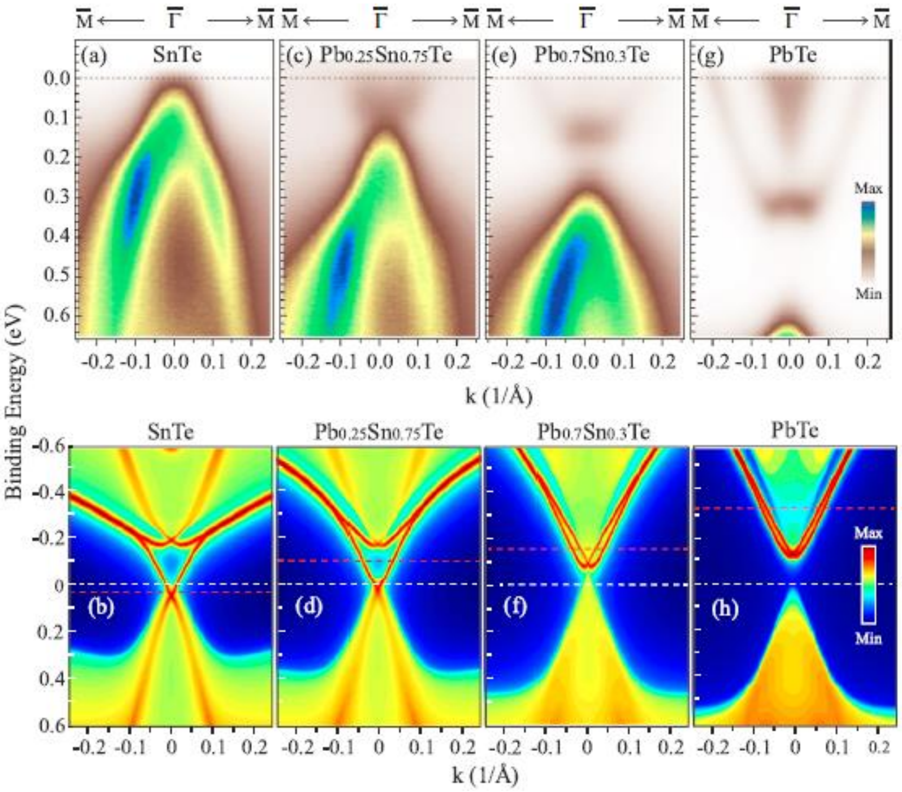


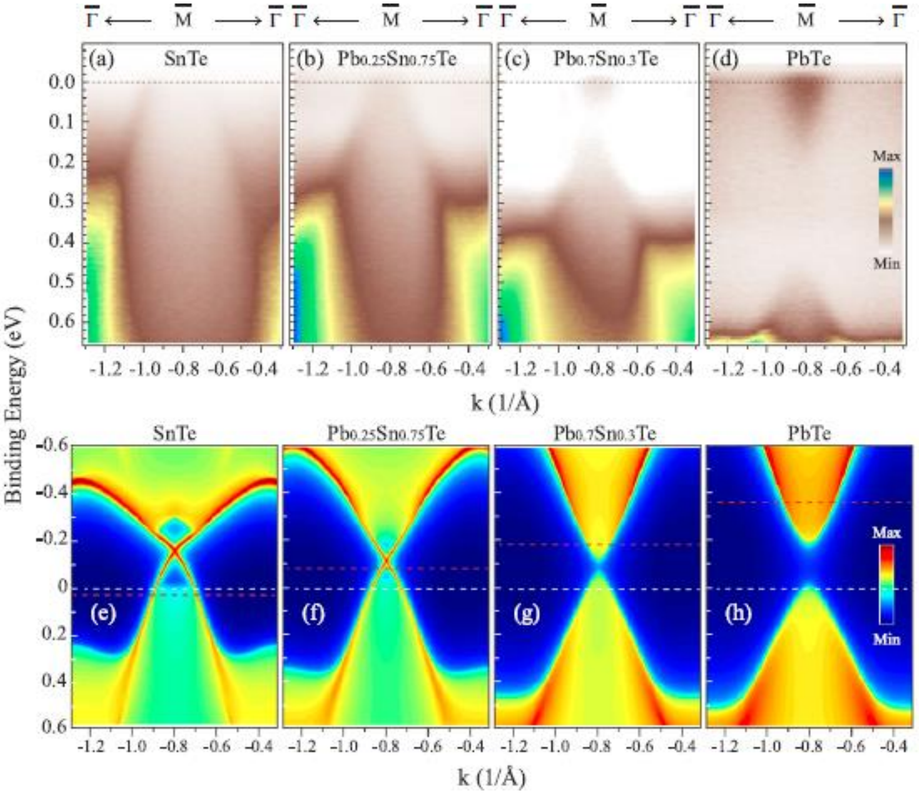
P. Dziawa et al. Nat. Mat. (2012)



Y. Tanaka et al. Nat. Phys. (2012)

Experiments of Pb$_{1-x}$Sn$_x$Te (111) surface

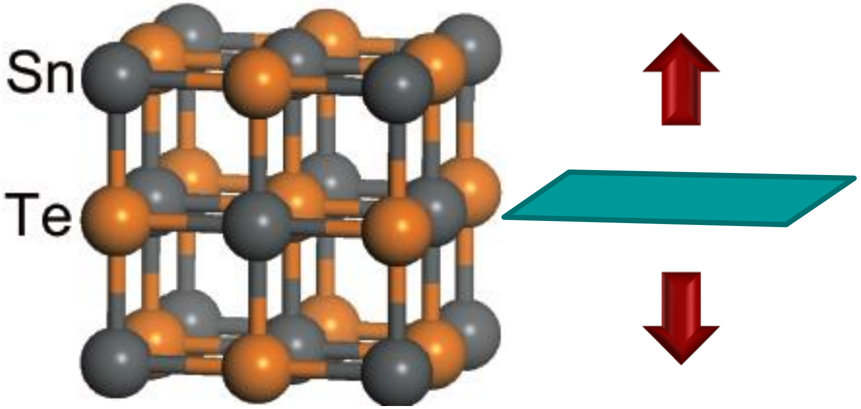


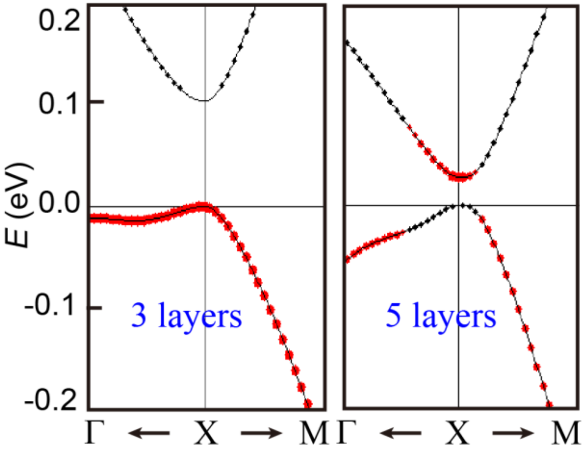


1. Phase transition by varying the ratio of Pb/Sn
2. Dirac cone in both Gamm and M points
3. Band bending is important to understand the surface states
4. Te terminated surface

Yan*, **J. Liu*,** et al, **_PRL_**. 112, _186801(2014)_

3. 2D topological crystalline insulators





Band Inversion

$$H(k) = \tilde{m}\tau_z + \left(\tilde{v}_x k_x s_x - \tilde{v}_y k_y s_y\right)\tau_x$$

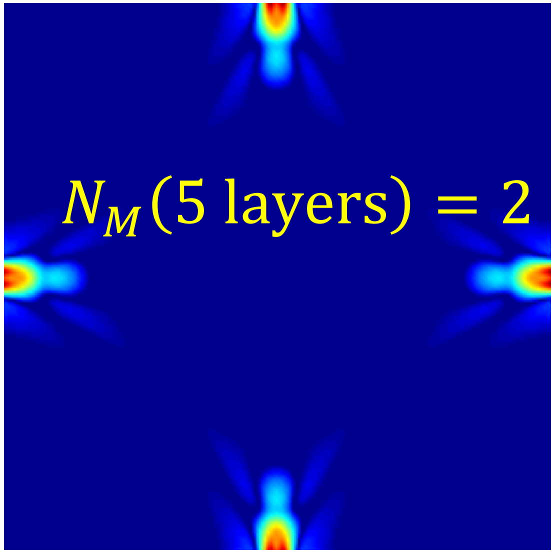
$$MH(k)M^{-1} = H(k)$$

$$M = -is_z\tau_z$$

Mirror Chern number

$$N_M = (n_{+i} - n_{-i})/2$$

$$|N_M(\tilde{m} > 0) - N_M(\tilde{m} < 0)| = 2$$

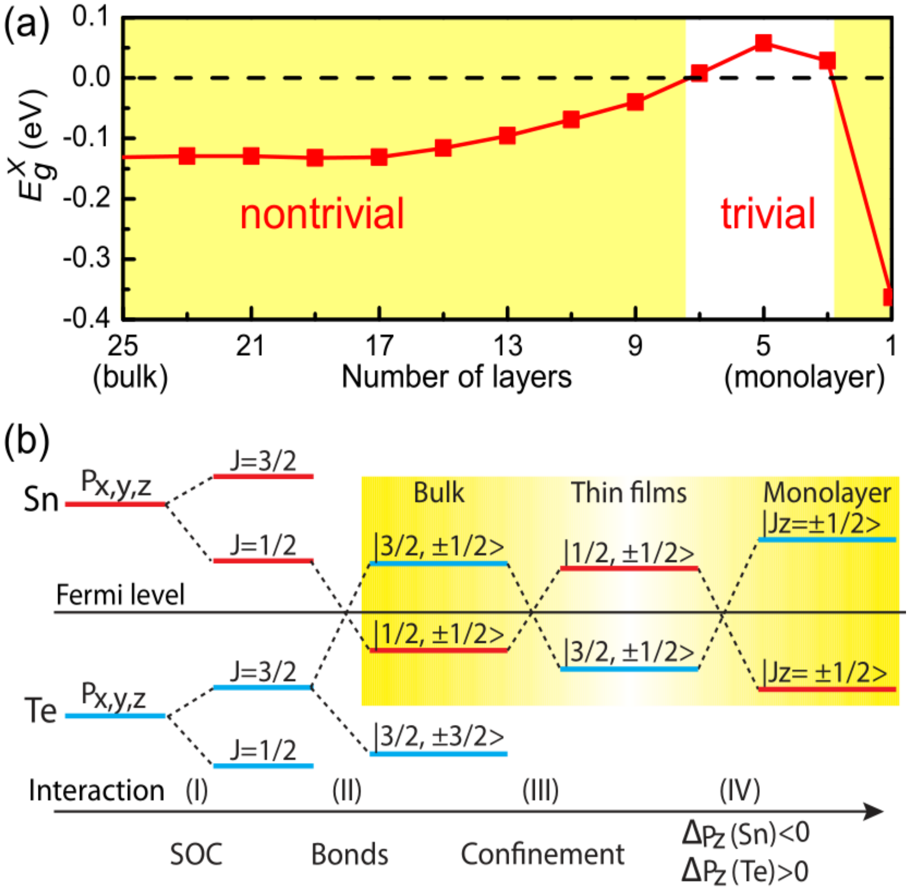


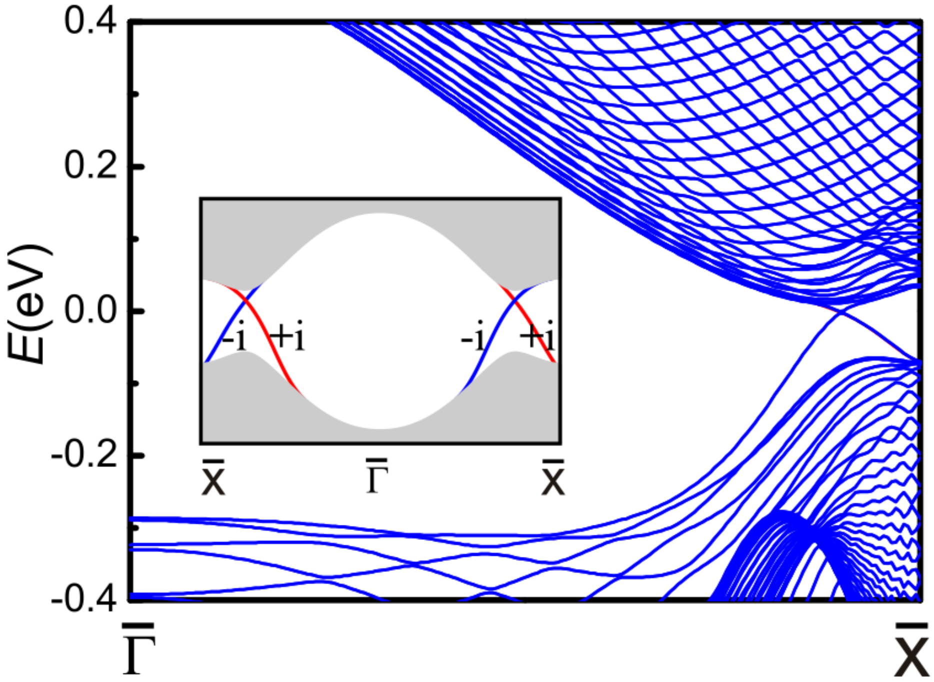
$N_M(5\ \text{layers}) = 2$

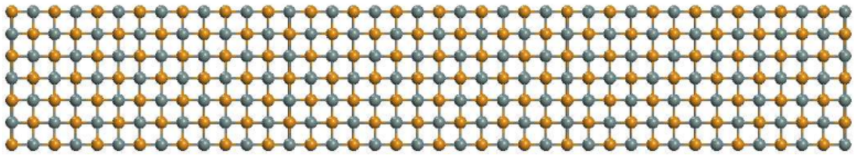
Non-zero MCN

J. Liu**, et al.* ***Nat. Mater. *13, 178 (2014)*

Odd number of layers thin films are 2D TCIs







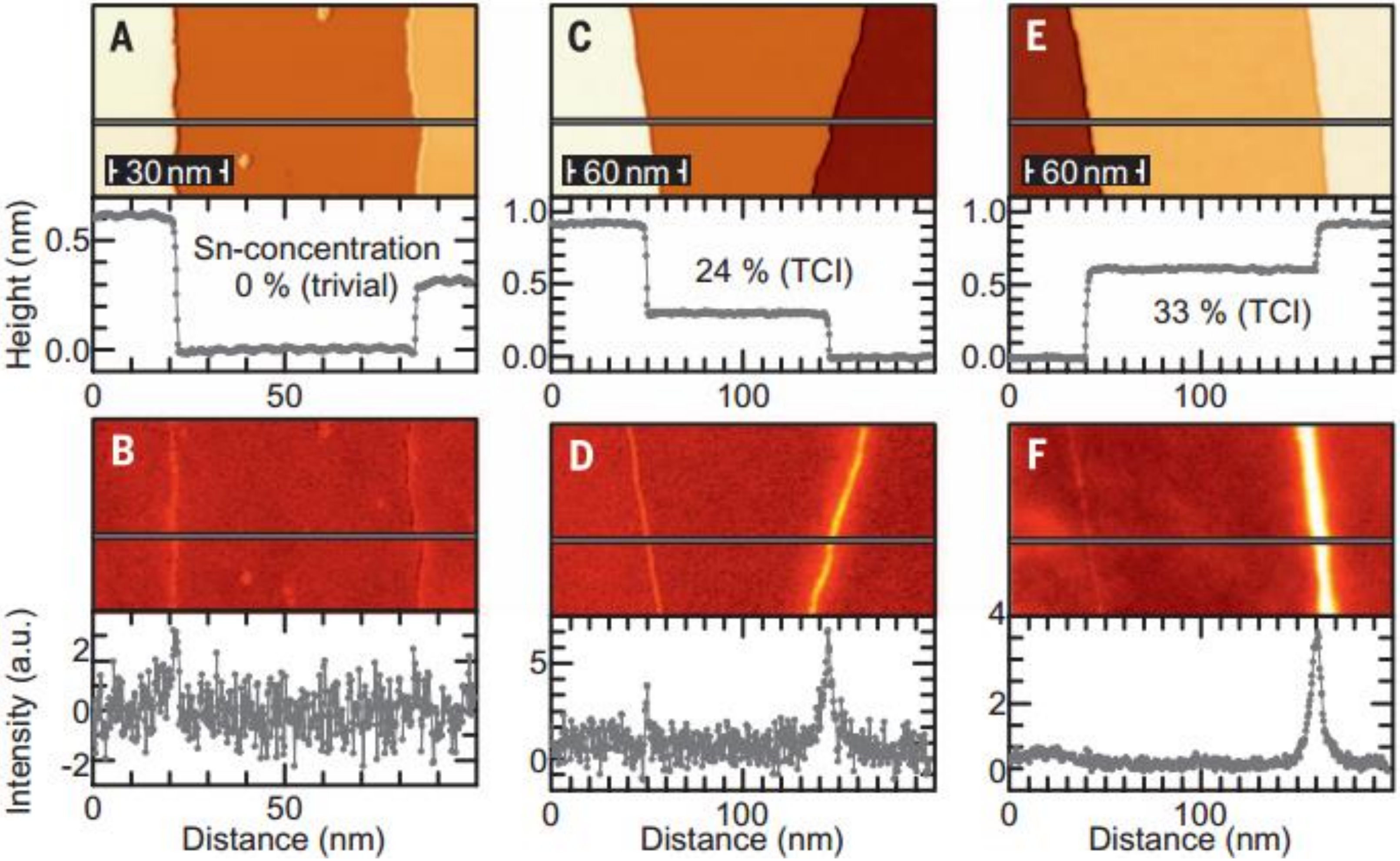
- Odd number of layers thin films can host topological protected edge states
- Even number of layers thin films cannot

***J Liu**, et al, **Nano Lett.** 15, 2657 (2015)*

Experimental confirmation of our prediction

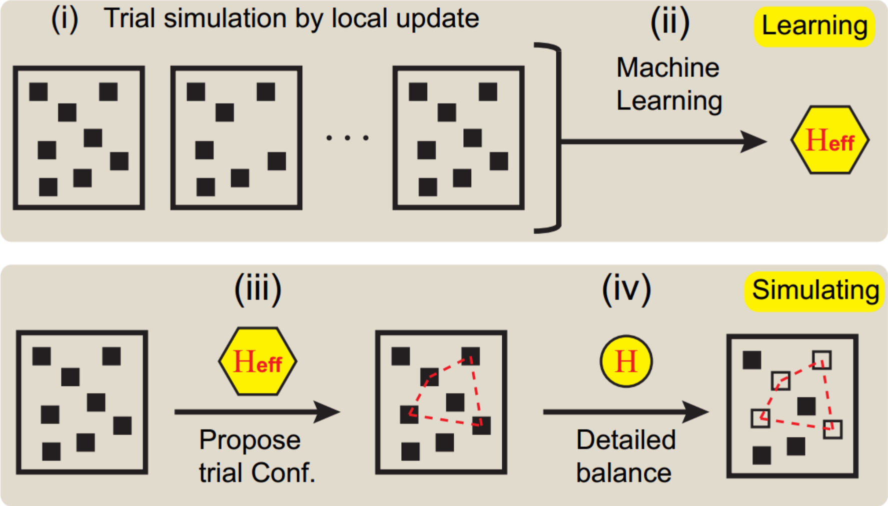
Science

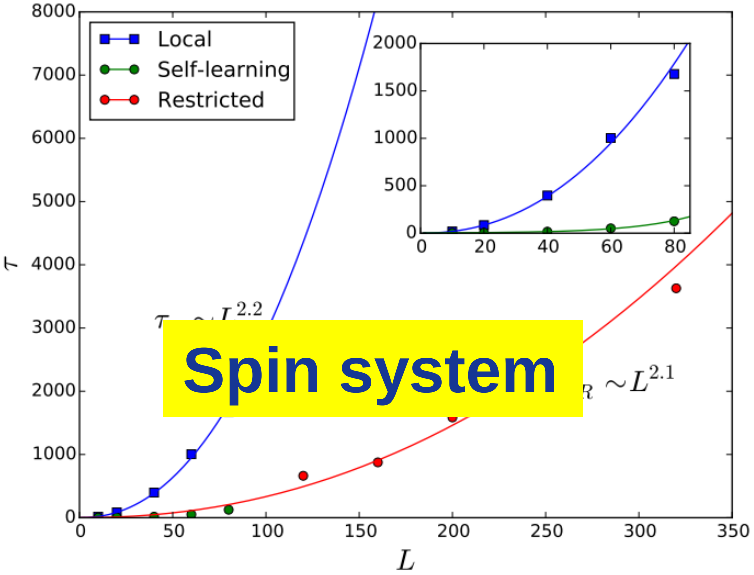
Robust spin-polarized midgap states at step edges of topological crystalline insulators
Paolo Sessi, Domenico Di Sante, Andrzej Szczerbakow, Florian Glott, Stefan Wilfert, Henrik Schmidt, Thomas Bathon, Piotr



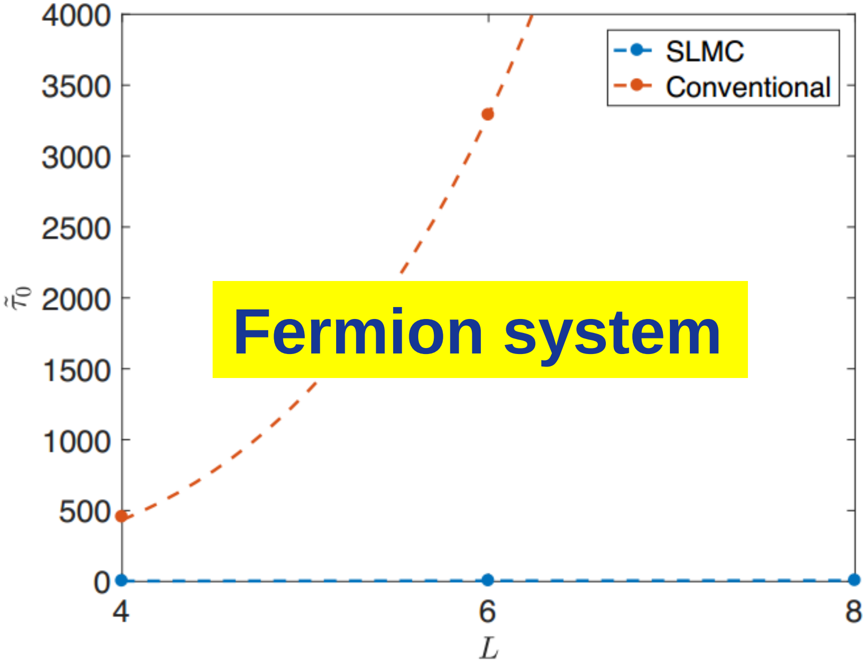
P Sessi, et al., Science 354, 1269-1273 (2016)

4. Self-learning Monte Carlo method





Spin system

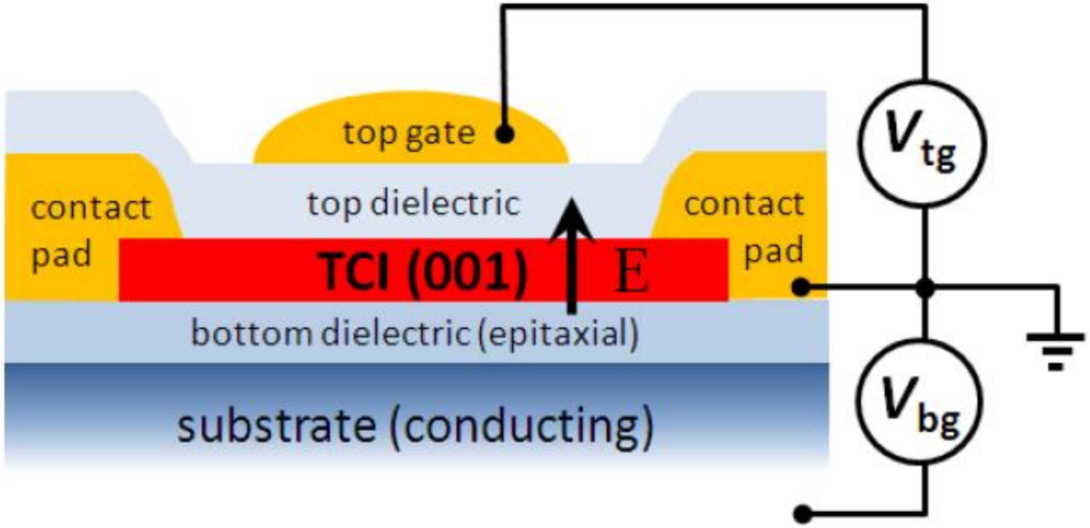


Fermion system

- SLMC can generally speed up Monte Carlo simulations
- SLMC in Fermion systems are more powerful and general, which can easily obtain 1000 times speedup.

***J Liu**, et al. PRB 95, 041101(R) (2017)*
***J Liu**, et al. PRB 95, 241104(R) (2017)*

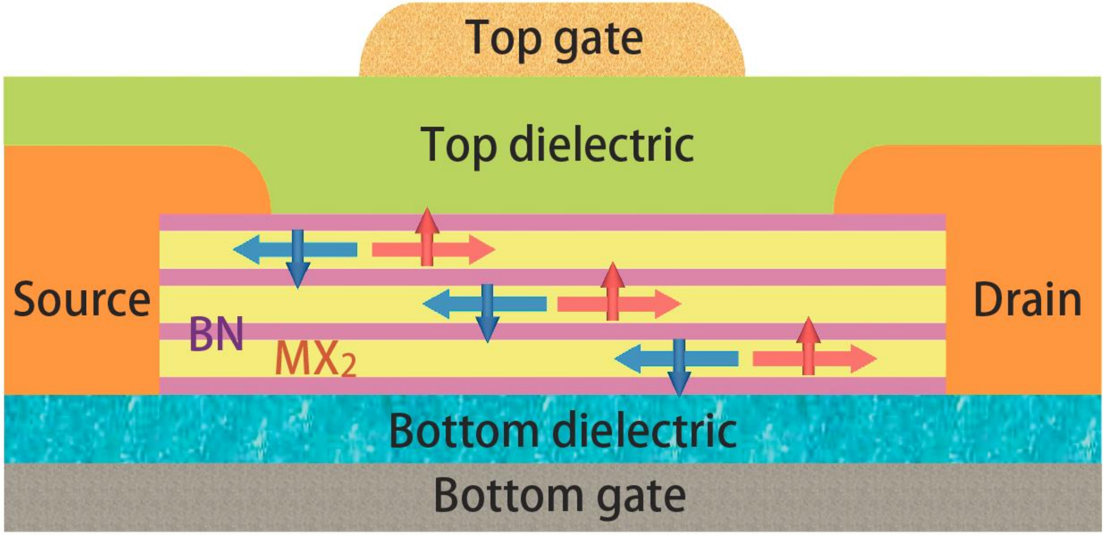
5. Applications: topological field transistor



- Based on 2D TCI
- Electric field breaks the mirror symmetry, hence switches ON/OFF states

***J. Liu**, et al.*
***Nat. Mater.** 13, 178 (2014)*

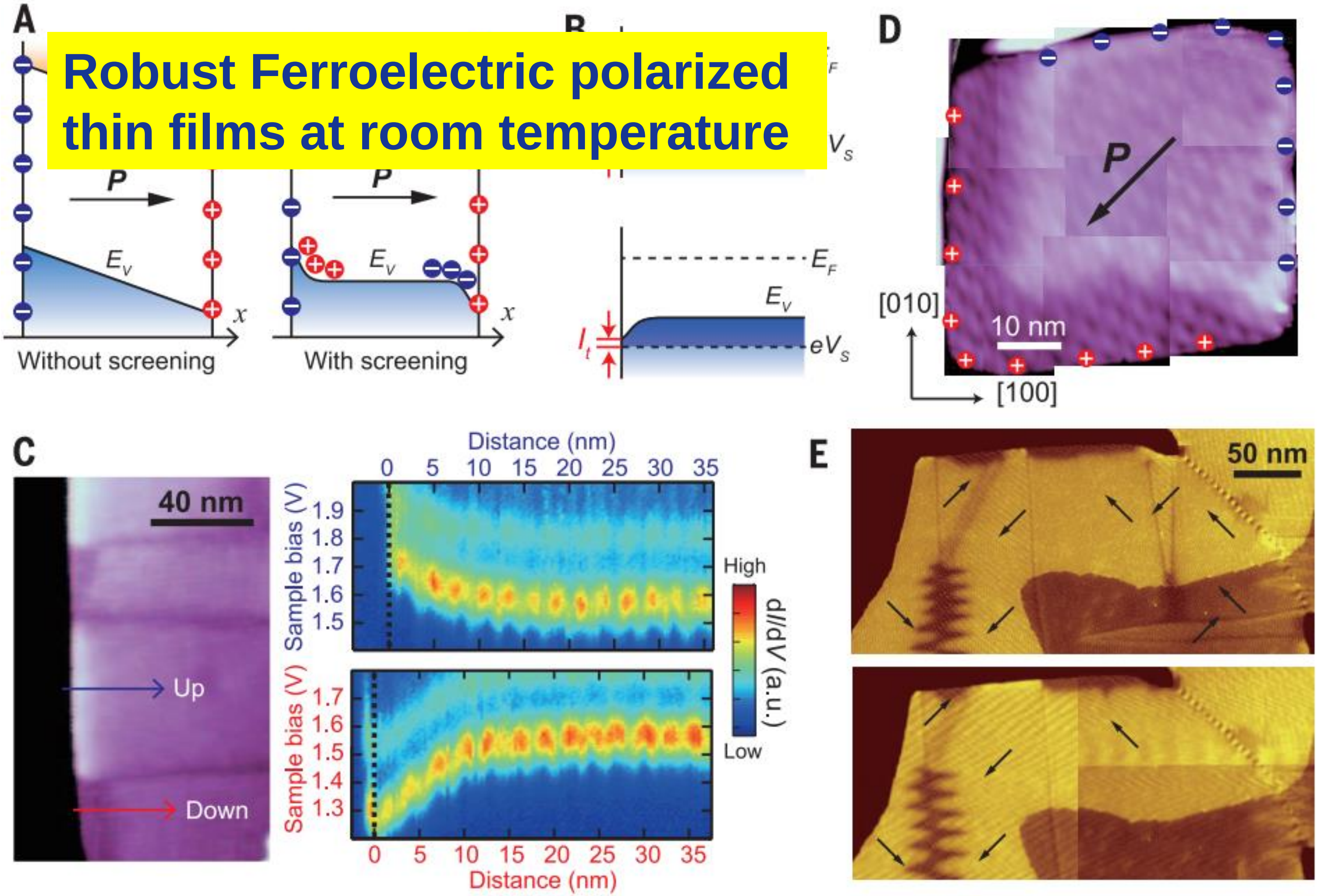
- Based on QSHI
- Electric field induced topological phase transition, hence switches ON/OFF states



X. Qian, **J. Liu***, et. al **Science** 346, 1344 (2014)*

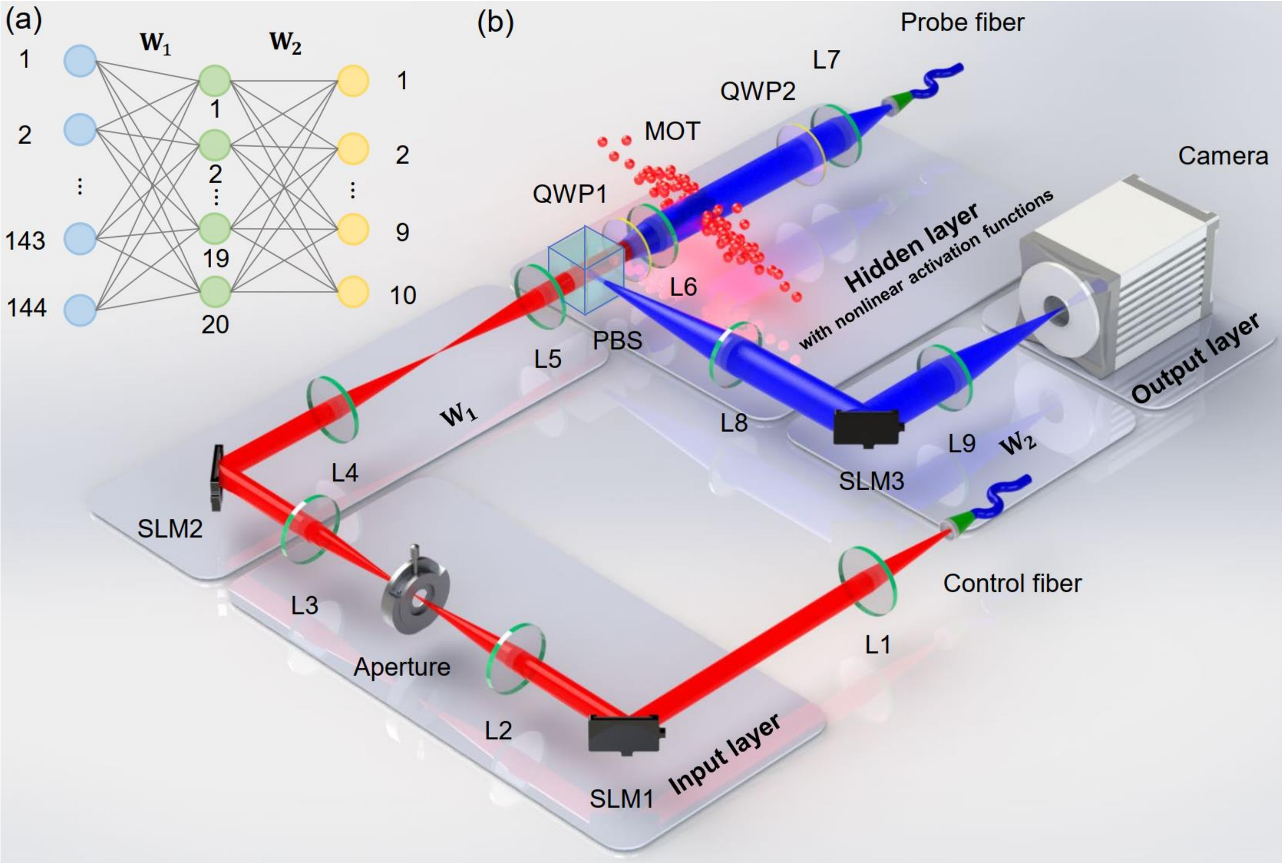
6. Even number of layers thin films of 3D TCI

Robust Ferroelectric polarized thin films at room temperature



K Chang, **J Liu***, et al* ***Science*** *353, 274 (2016)*

All-optical deep neural networks



[***Optica*** *6, 1132 (2019)*]

[***Phys. Rev. App.*** *15, 054034 (2021)*]

- At the speed of light
- Infinite parallel calculations
- Reprogrammable
- Scalable

World-first all-optical deep neural network with non-collinear activation functions

What is computer?

- Computer is really stupid, and it can only do what it is told. And the most important advantage of a computer is that it can do the very tedious repeating operations. It does not feel boring or tired and can give us the reliable results. Such tedious tasks are really challenging for the human.
- On the other hand, Alan Turing demonstrate that computer is really powerful. It can compute anything by using only six primitive operations and a long enough tap: **move left, move right, read, write, scan and do nothing.**
- A programming language is the tool or translator between human and computer. Choose your favorite language, and find the most suitable language for the given problems.

A numerical example

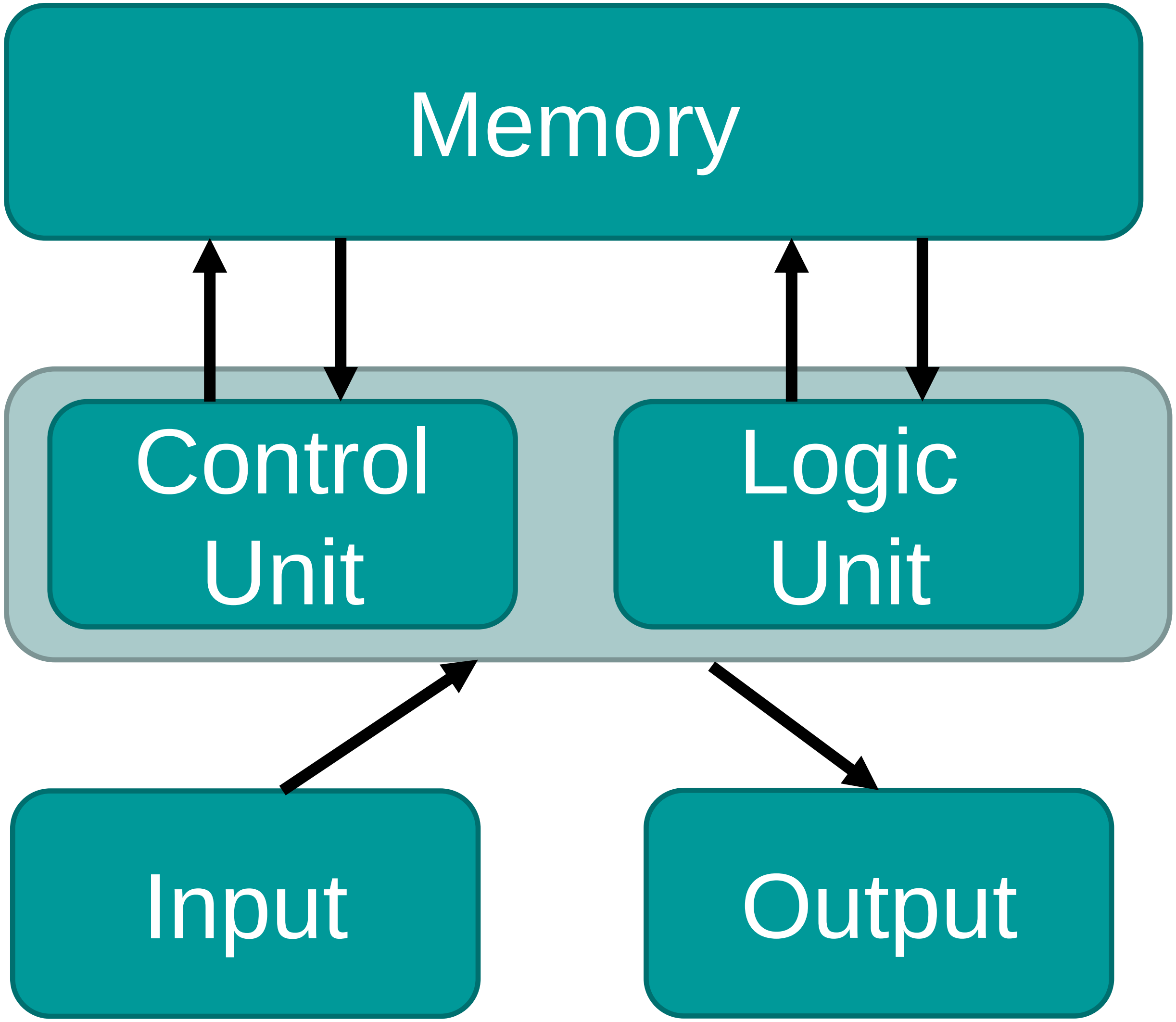
- Square root of a number ***x*** is ***y***, ***y*y=x***
- How to get y if I give you the value of ***x***, e.g. ***x=17***

1) Start with a guess, ***a***
2) If ***a*a*** is close enough to ***x***, stop and say ***a*** is the answer
3) Otherwise, we make a new guess as ***(a+x/a)/2***
4) Repeat 2) and 3), until we get the right answer

a	a*a	x/a	(a+x/a)/2
3	9	5.6667	4.3333
4.3333	18.7775	3.9231	4.1282
4.1282	17.0420	4.1180	4.1231
4.1231	17.0000	4.1231	4.1231

From Ana Bell at MIT

How does computer work?



From Ana Bell at MIT

Introduction to Python

1. Basic programming
    - Variables and assignments
    - Variable types
    - Output and input
    - Arithmetic
    - Comments
    - List and arrays
2. Controlling programs
    - If statement
    - While statement
    - For statement
3. Functions, packages, and modules
4. Graphics and visualization
    - Various plots
    - Animation
5. Good programming style

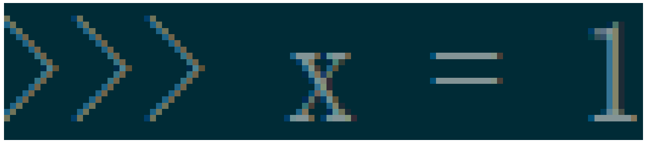
How to use Python

1. Command line
2. Direcly Run .py file
3. Jupyter Notebook
4. Different Integrated Development Environment (IDE)
    - Spyder
    - Pycharm
    - Visual Studio Code
    - Visual Studio
    - Others

✓ Carefully maintain the environment and packages.

✓ Take care of different versions of Python and Packages.

Variables and assignments



```
>>> x = 1
```

This is an *assignment* statement

1. There is a variable called x
2. Assign the value of x to be 1

Name of a variable:

- Can be as long as you like
- Can be any letter or number and underscore symbol "_"
- Cannot start with number
- Cannot contain any other symbols, or spaces
- Upper- and lower-case letters are distinct from one another (x and X are different)
- Give your variable meaningful names that describe what they represent

Variable types

Five types of variable we mainly use:

- Integer: can take integer values and integer values only, both positive and negative values are allowed
- Float: can take real, or floating-point, values. Notice that a floating-point variable can take an integer value, but an integer variable cannot take non-integer values
- Complex: In python the unit imaginary number is called j, not i.
- Bool: True, False
- String: is often used in Python programs but comes up only rarely in physics programming

Why do we need different types of variables?

- Memory space could become a limiting factor in writing the program of many problems. Different type of variables take different size of Memory. (Integer< float < complex)
- Calculations with complex number take longer to complete, because the computer has to calculate both the real and imaginary parts.
- Integer variable are actually more accurate than floating-point variables. Floating-point calculations on computers are not infinitely accurate. The difference between1 and 0.9999999999999999 or 1.0000000000000001 could be crucially important.
- Numerous bugs and problems in computer programs have arisen because of exactly this kind of issue.

How to set the type for a variable?

- The type of variable is set by the value what we give it.
  x=1 and x=1.0 are different

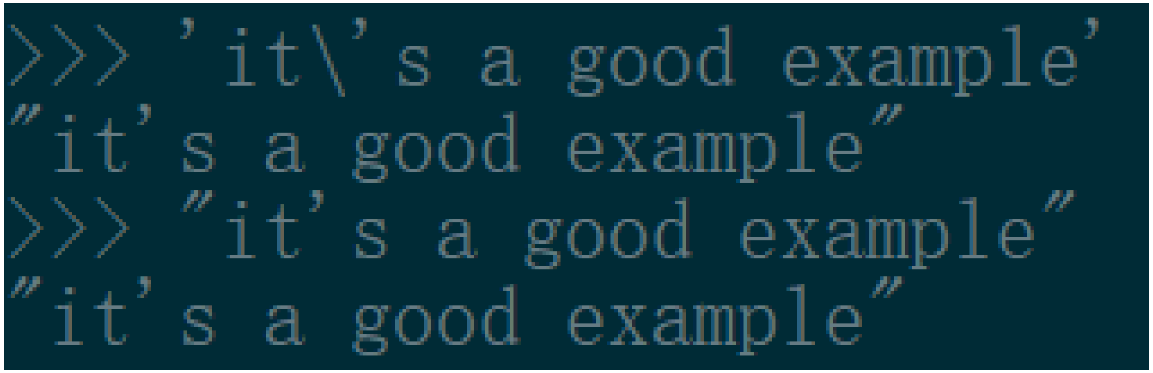
- We can use functions to set the variable type
  x=float(1)

- Check the variable type
  type(x)

- The type of a variable can change as a Python program runs. It is better not to do that although you can. It will make the program more difficult to follow and increase the chance that you may make a mistake in your programming.

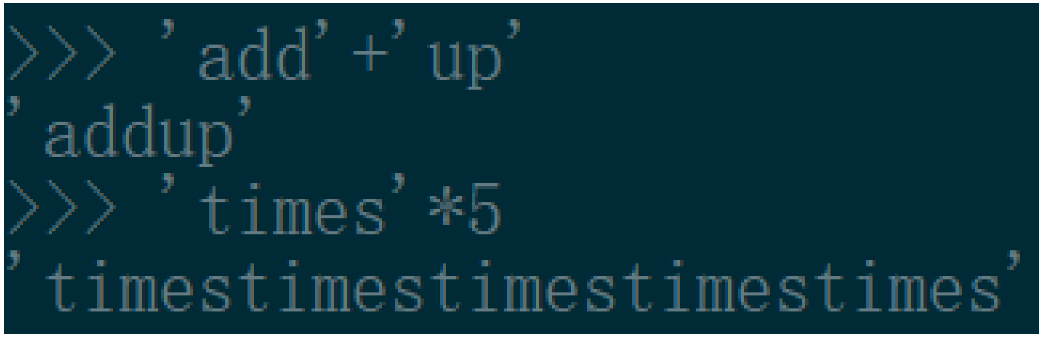
string

- Python can also manipulate strings, which can be expressed in several ways. They can be enclosed in single quotes ('...') or double quotes ("...") with the same result [2]. \ can be used to escape quotes:



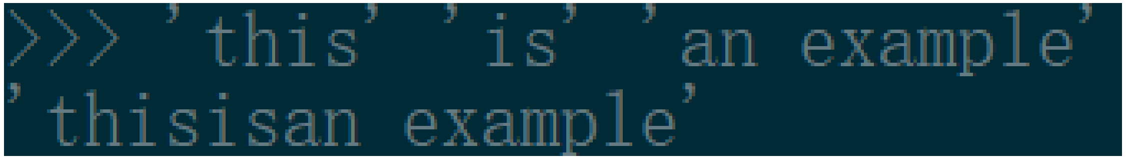
```
>>> ’it\’s a good example’
”it’s a good example”
>>> ”it’s a good example”
”it’s a good example”
```

- Strings can be glued together with the + operator and repeated with *.



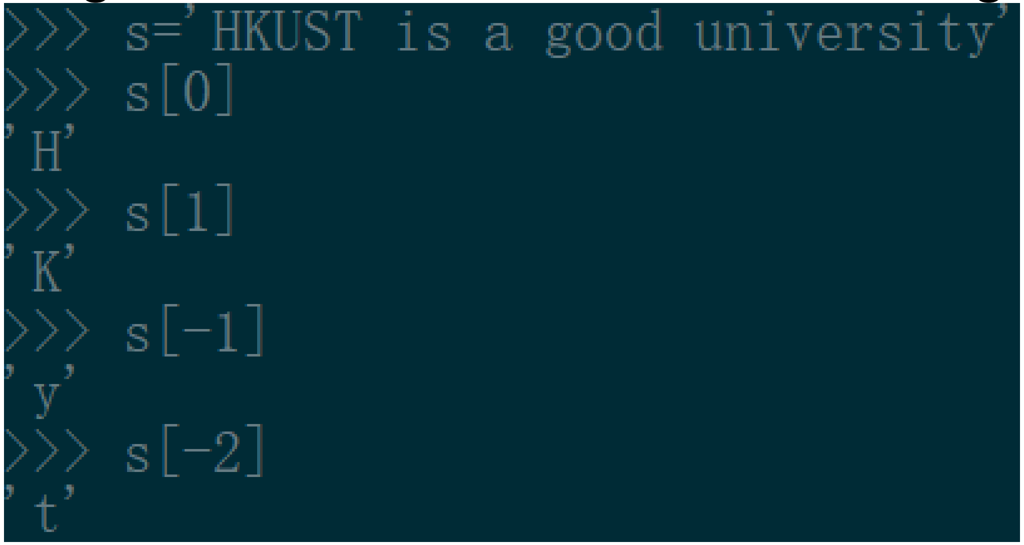
```
>>> ’add’+’up’
’addup’
>>> ’times’*5
’timestimestimestimestimes’
```

- Two or more string literals (i.e. the ones enclosed between quotes) next to each other are automatically concatenated.



```
>>> 'this' 'is' 'an example'
'thisisan example'
```

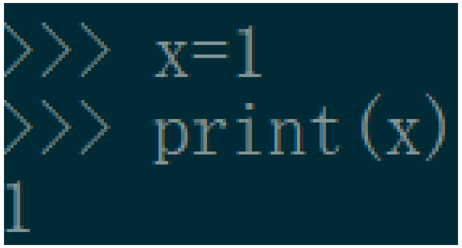
- Strings can be indexed (subscripted), with the first character having index 0. There is no separate character type; a character is simply a string of size one; Indices may also be negative numbers, to start counting from the right:



```
>>> s='HKUST is a good university'
>>> s[0]
'H'
>>> s[1]
'K'
>>> s[-1]
'y'
>>> s[-2]
't'
```

- The built-in function len() returns the length of a string:

Output statements



```
>>> x=1
>>> print(x)
1
```

- print() always prints the current value of the variable at the moment the statement is executed.

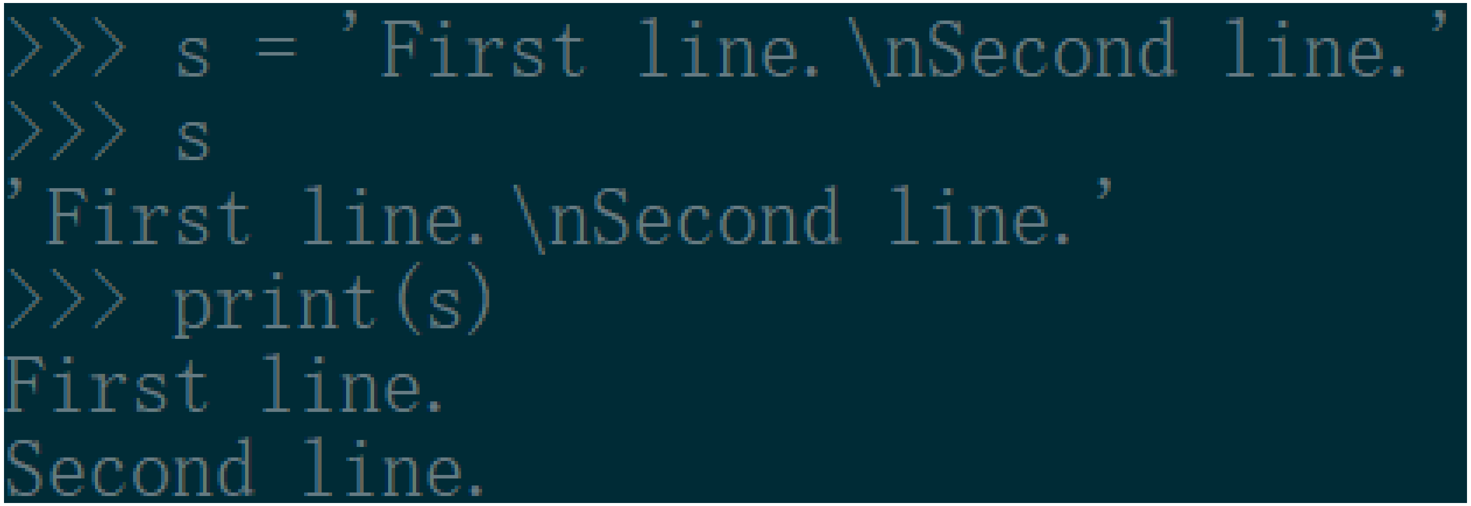
- print(x) and print(‘x’) is totally different

- Adding a few words to your program in the output statement can make the output much easier to read and understand.

- It is a dumb but a very useful way to use print() to debug your codes.

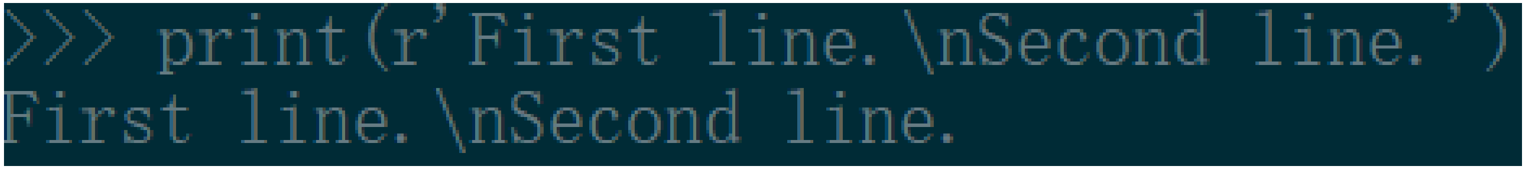
Some examples

- \n means a new line



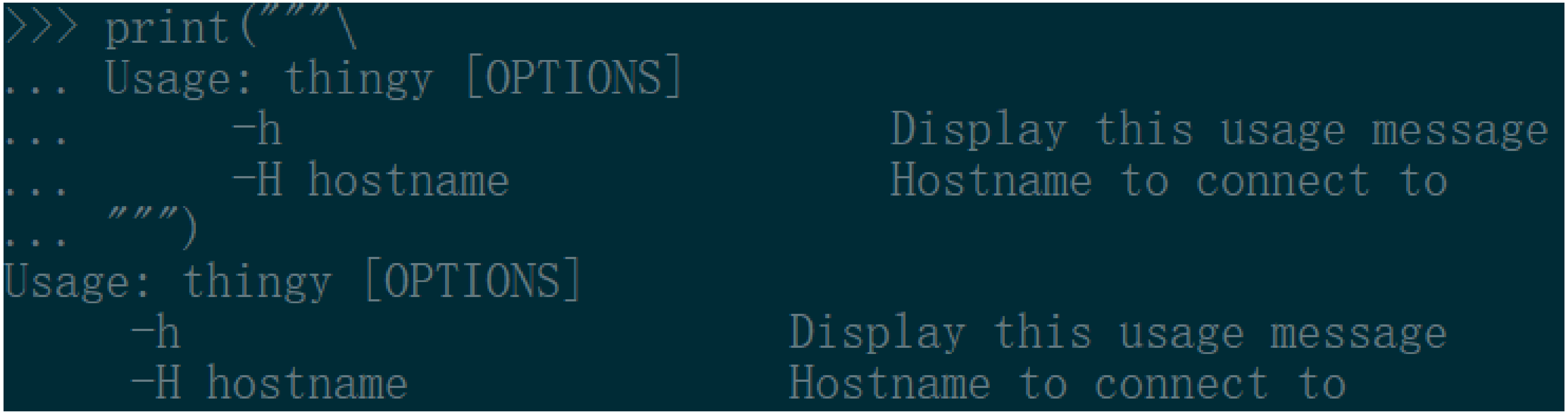
```
>>> s = 'First line.\nSecond line.'
>>> s
'First line.\nSecond line.'
>>> print(s)
First line.
Second line.
```

- If you don’t want characters prefaced by \ to be interpreted as special characters, you can use raw strings by adding an r before the first quote:



```
>>> print(r'First line.\nSecond line.')
First line.\nSecond line.
```

- String literals can span multiple lines. One way is using triple-quotes: """...""" or '''...'''. End of lines are automatically included in the string, but it’s possible to prevent this by adding a \ at the end of the line. The following example:



```
>>> print("""\
... Usage: thingy [OPTIONS]
...      -h                         Display this usage message
...      -H hostname                Hostname to connect to
... """)
Usage: thingy [OPTIONS]
     -h                         Display this usage message
     -H hostname                Hostname to connect to
```

Format output

- The general form

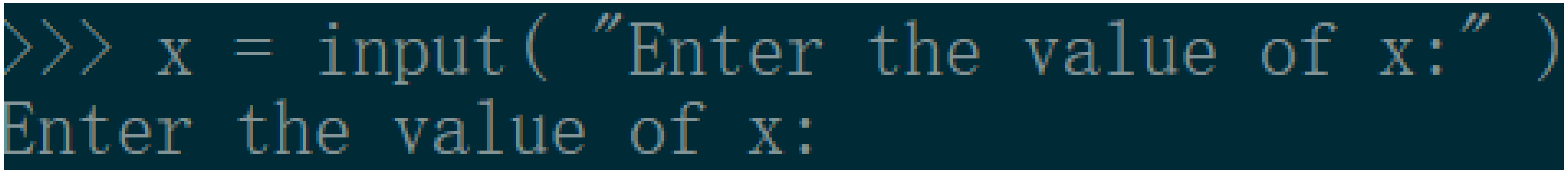
print(“a = %letter, b = %letter, c = %letter.” % (value1, value2, value3))

- %letter could be %s, %d, %x, %o, %f, %e, %g, %r

%*m*s	String
%*m*d	Integer number
%*m*x, %*m*X	hexadecimal number
%*m*o, %*m*O	Octonary number
%*m.n*f	Float number
%*m.n*e	Scientific notation
%g	Automatically choose suitable type %f or %e
%r	The original format

m,n are integers

Input statements

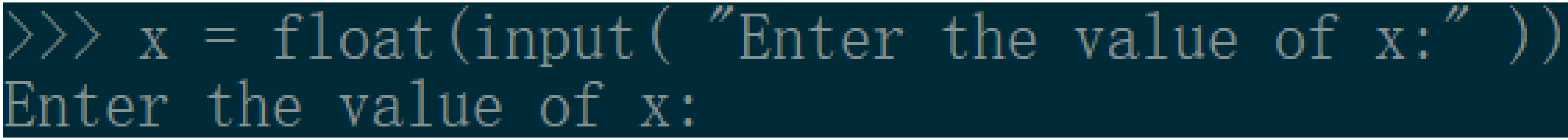


```
>>> x = input( ”Enter the value of x:” )
Enter the value of x:
```

- The computer will wait until you type a value on the keyboard.

- The value entered is always interpreted as a string value, even if you type in a number. The computer does not care whether you enter digits, letter, a complete work, or several words.

- We need to convert a string into a number if you want to input a number.



```
>>> x = float(input( ”Enter the value of x:” ))
Enter the value of x:
```

Arithmetic

- x+y addition
- x-y subtraction
- x*y multiplication
- x/y division
- x**y raising x to the power of y
- x//y the integer part of x divided by y
- x%y modulo, the remainder after x is divided by y
- The end result is the same type as the starting values, or the more general type if there are two different starting types
- Number can multiply string, which mean repeat the string several times.
- It is not true for the matrix

More tricks

- Python modifiers
  x+=1, x -=4, x*= 3.5, x /= 5*y

- The ability to assign the values of two variables with a single statement.
  x, y = 1, 2.5 equivalent to x=1 and y=2.5

  how about this one,   x, y = 2*z+1, (x+y)/3

- We do not need an additional temporary variable to swap the values of x and y
  x, y = y, x

Controlling programs: if statement

-
  ```
  if conditions :
        codes
  ```

-
  ```
  if conditions :
        codes
  else :
        codes
  ```

-
  ```
  if conditions :
        codes
  elif conditions :
         codes
  else :
         codes
  ```

- logical operators
  ==, >, >=, <, <=, !=

- x==1 is different form x=1

- Combine different conditions
  or, and

- The indentation is crucial in Python, spaces at the beginning of lines do have an effect with an ***if*** statement or other controlling statement.

Controlling programs: while statement

-
  ```
  while conditions :
      codes
  ```

-
  ```
  while conditions :
      codes
  else :
      codes
  ```

-
  ```
  while conditions :
      codes
      if conditions:
          codes
          break or continue
      codes
  ```

- check if the condition is met. If it is, it will executes the indented block of code immediately following; if not, it skips the block.

- If the condition is met and the block is executed, the program then loops back to the beginning and checks the condition again.

- Used to ensure some conditions is met in a program or to keep on performing an operation until a certain situation is reached.

- Infinite loops

Combine different conditions

- In many cases, we need to combine different conditions to get more complicated condition in if and while statement

- Three type of operators are

“and”, “or”, “not”

- If there are more than two operators, the rules are as following

1. From left to right
2. The priority of “and” is higher than “or”

- The return is not always a bool type

- Check it carefully before use it

Examples and Practices

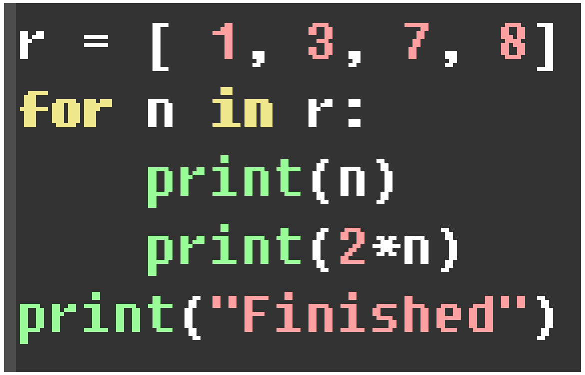
1. a= True or False and True and False
2. a= (True or False) and True and False
3. a= 1 and 2
4. a= 2 and 1
5. a= 1 or 2
6. a= 2 or 1
7. a= not 1
8. a= not 0
9. a= 0 and True;   b = 0 or True
10. a= 0 and False; b = 0 or False
11. a= 1 and True;   b = 1 or True
12. a= 1 and False; b = 1 or False
13. a= 2 and True;   b = 2 or True
14. a= 2 and False; b = 2 or False

type(a); type(b); print(a); print(b)

Controlling programs: for statement

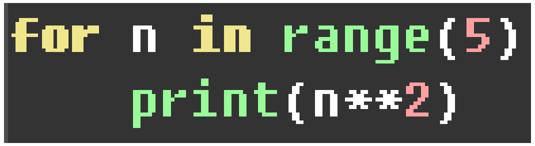
- We can use while statement to repeatedly execute a given section of code. However, it is rarely used in practice. Instead, there is another much more commonly used loop construction in the Python, ***for*** loop.

- A for loop is a loop that runs through the elements of a list or array in turn.



```
r = [ 1, 3, 7, 8]
for n in r:
    print(n)
    print(2*n)
print("Finished")
```

- Use function ***range(a,b, steps)***



```
for n in range(5)
    print(n**2)
```

Comments

- Like other computer languages, Python also have the important feature, ***comments***

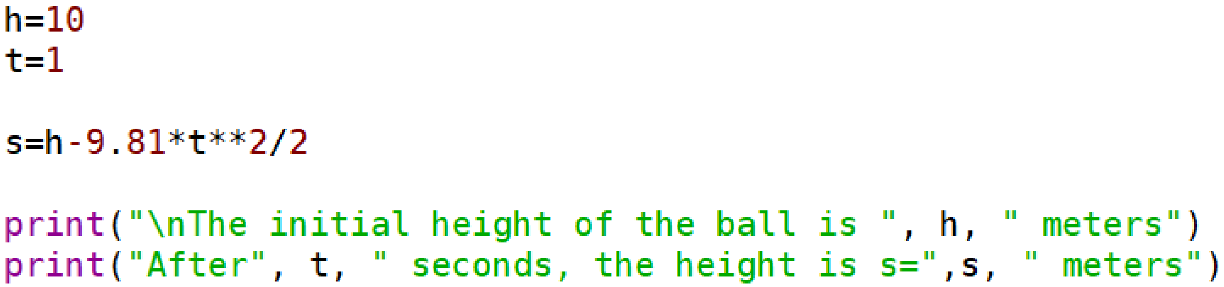
- In Python, any program line that starts with a hash mark “#” is ignored completely by the computer.

- Comments make no difference to the way a program runs, but they do make huge difference for you!

- Comments do not have to start at the beginning of a line. Python ignores any portion of a line that follows a hash mark, whether the hash mark is at the beginning or not.

Example I

- The problem: a ball is dropped from a tower of height ***h***. It has initial velocity zero and accelerates downwards under gravity. The challenge is to write a program that asks the user to enter the height ***h*** in meters of the tower and a time interval ***t*** in seconds, then prints on the screen the height of the ball above the ground at time ***t*** after it is dropped, ignoring air resistance.

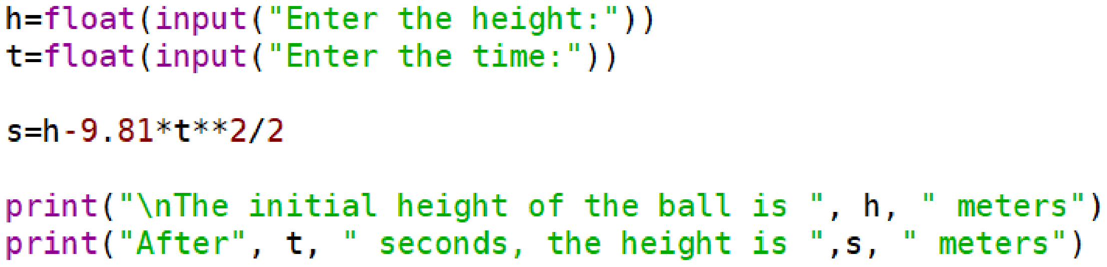


```
h=10
t=1

s=h-9.81*t**2/2

print("\nThe initial height of the ball is ", h, " meters")
print("After", t, " seconds, the height is s=",s, " meters")
```

- You need to change your code every time you change the parameters



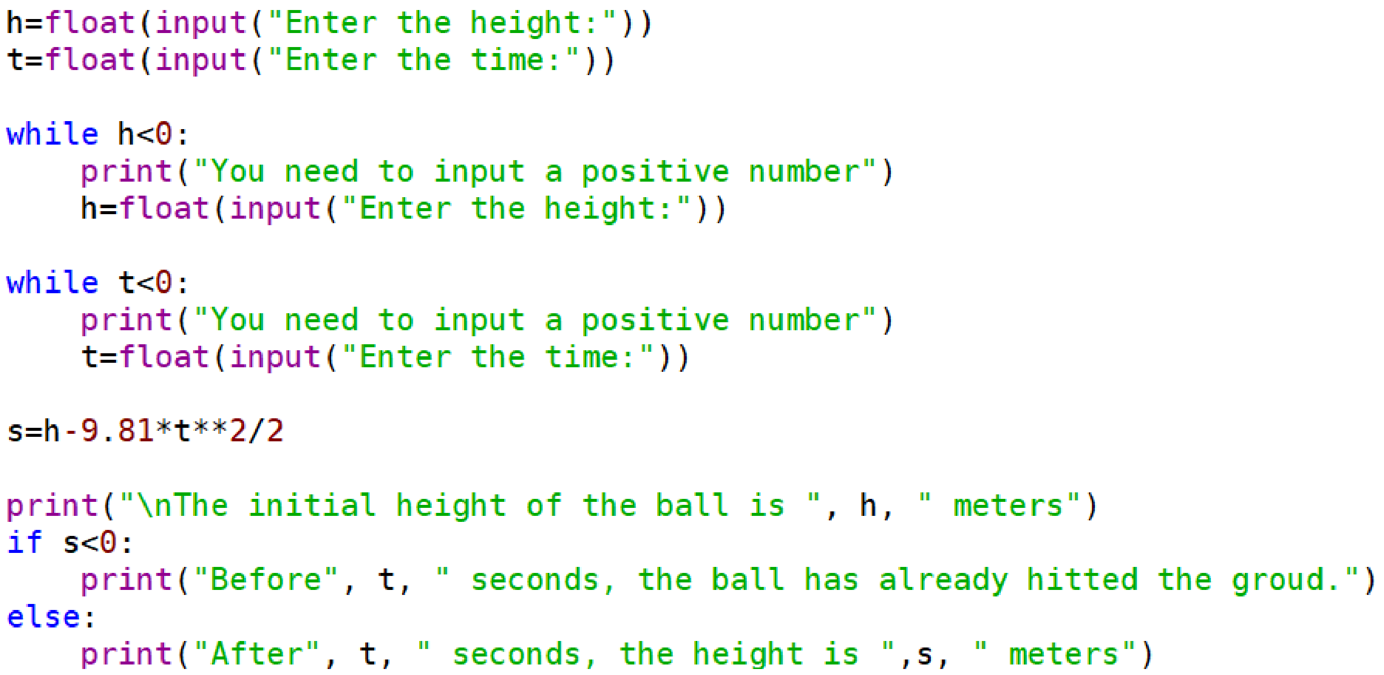
```
h=float(input("Enter the height:"))
t=float(input("Enter the time:"))

s=h-9.81*t**2/2

print("\nThe initial height of the ball is ", h, " meters")
print("After", t, " seconds, the height is ",s, " meters")
```

- What if h<0, t<0 or s<0? It does not make any sense!

A ball dropped from a tower with control



```
h=float(input("Enter the height:"))
t=float(input("Enter the time:"))

while h<0:
    print("You need to input a positive number")
    h=float(input("Enter the height:"))

while t<0:
    print("You need to input a positive number")
    t=float(input("Enter the time:"))

s=h-9.81*t**2/2

print("\nThe initial height of the ball is ", h, " meters")
if s<0:
    print("Before", t, " seconds, the ball has already hitted the groud.")
else:
    print("After", t, " seconds, the height is ",s, " meters")
```

- The program becomes more reasonable with the controlling statements and less unexpected (wrong) results
- Is it good enough?