HD EDU

INFO1110 Introduction to Programming

同步辅导课1

Tutor：Hilbert

HD·EDUCATION
全球累计服务用户超十万

HD EDU



HD EDU

企业介绍

海道教育（HD EDU）成立于2018年，是一家助力国际化人才成长的教育科技企业，同时也是一个力求突破语言和文化障碍、推动国际教育与文化属性相融合的新锐品牌。

海道教育已实现全球规模化发展，旗下产品已走入英国、美国、澳大利亚、新西兰等主流留学的国家40多所院校，为超30万的中国留学生提供高品质的教学服务，成为留学后市场在线教育的领军企业。

2022年，HD EDU发布首个学术诚信合规准则，HD EDU愿始终与所有用户一起，坚守学术尊严，真正意义上打造既保护用户权益又懂用户需求的教学产品，推动海外华人留学生成长与发展。

产品介绍

成立五年来，HD EDU 作为专注于留学后市场服务的后起之秀，秉持着“提升中国留学生核心竞争力”的产品理念，深度洞察留学生教育场景叠加、需求变化，研发多阶段教育产品，为用户提供多元化、高品质的一站式留学教育服务，打造出符合留学生全生命周期的三大教育产品线：HD Tutoring、HD Career、HD Offer。

HD Tutoring

学习有海道，提分更有效

节点辅导　全学期学业辅导

HD Career

让年轻人找到喜欢的工作

技能课程　资格证书　实训项目　求职方案

HD Offer

专注藤校G5级别院校申请

选校定位　文书创作　申请服务　面试辅导


HD EDU
以用户的成长为第一

HD EDU

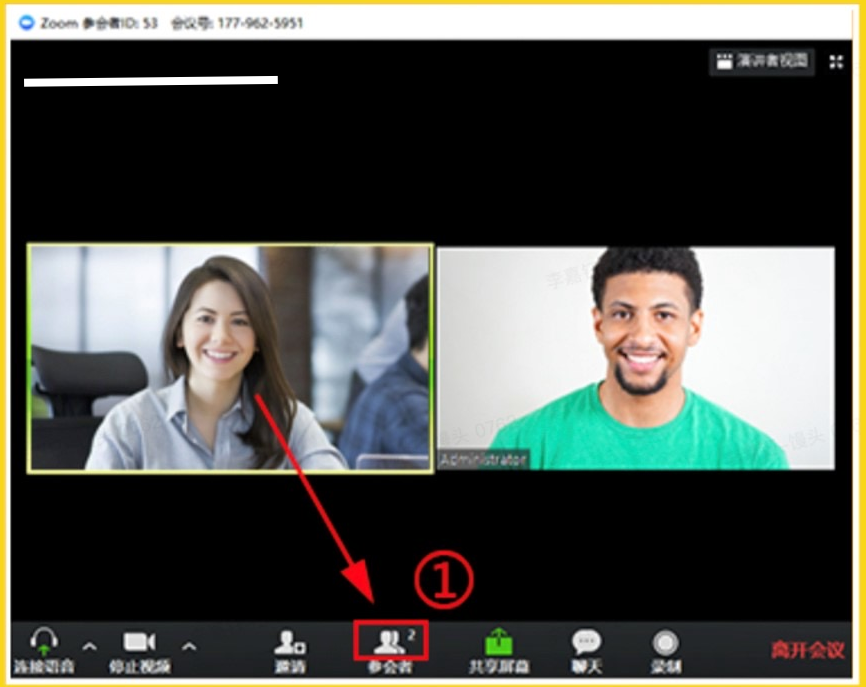


TUTOR Hilbert

Self- Introduction

- USYD Bachelor of Advanced Computing Honours First Class毕业 (Major in Software Development)
- 東京工業大学Master of Artificial Intelligence在读
- 曾任USYD INFO1110的Lab Tutor
- INFO1110总分91.0 HD
- 2018年加入HD Edu，主要教授预科及本科的IT入门课程
- HD Edu学科带头人兼面试官，曾经撰写过HD的INFO课程规范指南

同学们遇到问题怎么办？







1. 点【参会者】
2. 点【举手】即可与老师实时互动
3. 问题被解答了还可以【手放下】
4. 点【聊天】/【Chat】就可以与老师实时文字互动哦

直播平台互动方法

直播平台 ： 举手+聊天室提问



1. 点【参会者】再点【举手】，即可与老师实时互动！在此输入你想问的问题
2. 问题被解答了还可以【手放下】



Housekeeping

How to do well in INFO1110?

HD EDU

学科特点:

1. 理论与实践
2. 软硬件结合
3. Error

Learn Programming from

中文?

English?

学习方法:

1. 理论和实践分开来学习
2. Debug与试错
3. 刷题（认真对待Tutorial的题目和Ed上的练习）

INFO1110 is about Python?

答疑

HD EDU

- 提问之前，务必：
  - 先自己思考
  - Textbook, lecture slides, tutorial materials, etc.
  - Google, StackOverflow, CSDN, etc.



- 交流沟通的前提是尊重。大家都有很多要忙的事情，请珍惜彼此的时间，问有意义的问题！
- 专门答疑时间段：*待定*

PEP8

HD EDU

- *不需要严格参照*
- 但是请尽量写美观的代码！
- 最简单的方法，就是模仿我写代码的Convention。

本章知识点

1. Computer Program
2. Terminal and Shell Commands
3. Built-in functions: `print()`, `input()`
4. Strings and String formatting
5. Variables
6. Data Types and Objects
7. Expression and Operator
8. Comments





知识点讲解

考点1. Computer Program - Basics

HD EDU

What is a program?

计算机有自己思考的能力吗?

ChatGPT

A program is a set of instructions written in a programming language that a computer can interpret and execute.

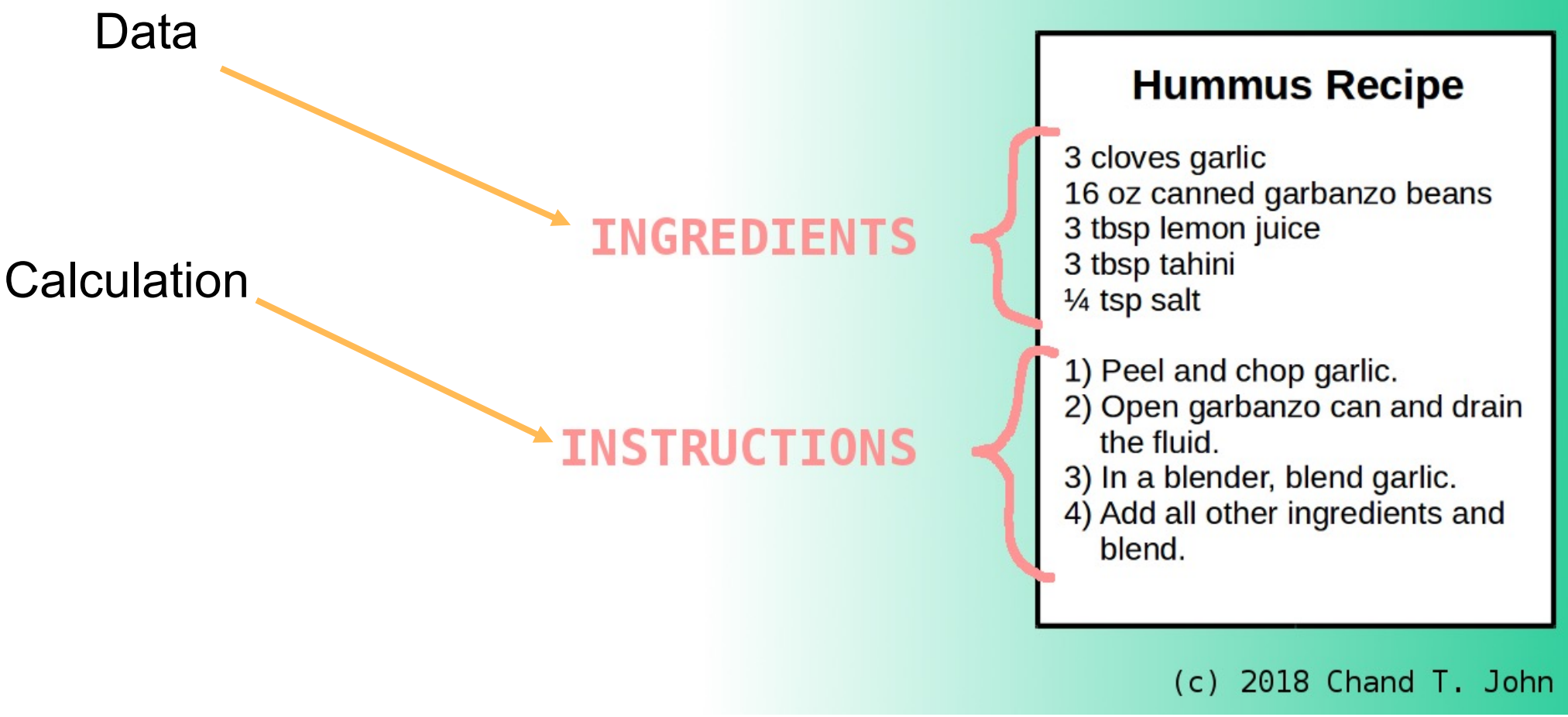
It defines a sequence of operations to perform a specific task, solve a problem, or achieve a desired outcome.

Programs can range from simple scripts to complex software applications, and they are the fundamental building blocks of computer software.

← What does a program do?

考点1. Computer Program – Recipe Example

HD EDU



Ref. https://www.cs-comics.com/computer-programs-are-like-recipes/

考点1. Computer Program – Curry Rice

HD EDU

Let’s make curry rice!

Raw Rice

Water

Vegetables

Oil

Rice

Curry Sauce



Curry Rice

Abstraction

We can cook from the base ingredients, but also, we can just purchase cooked rice and curry sauce or even just the curry rice

考点1. Computer Program – Basic CPU Instructions

The detail of this part is taught in **ELEC1601**.

CPU Instruction sets: CPU指令集，指CPU能理解并且执行的指令的集合。通常，仅包含非常基本的操作如内存的读写，两个数字比大小，两个数字的四则运算等。

How can we achieve complex operations? ➔ Abstraction!

Example:
Let's swap the data between position a and b!

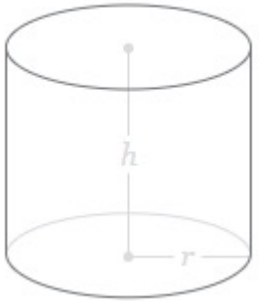


考点1. Programming Language

HD EDU

Me: Can you calculate the volume of a cylinder for me?
PC: 0b10101001010100101010010011010 ← Machine Code

$V = \pi r^2 h$



← **What does a program do?**

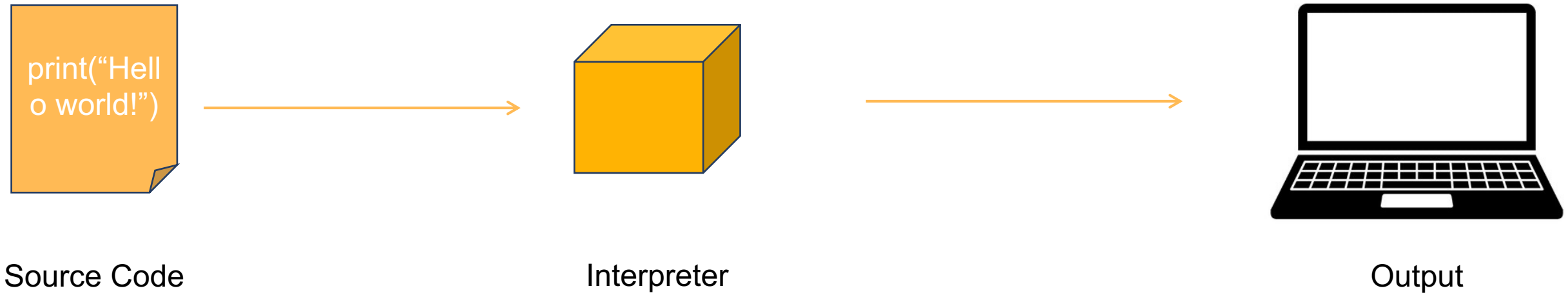
我们只能用Machine Code和电脑交流，但是它过于抽象。。。
人类语言 ➔ 二进制机械语言 X

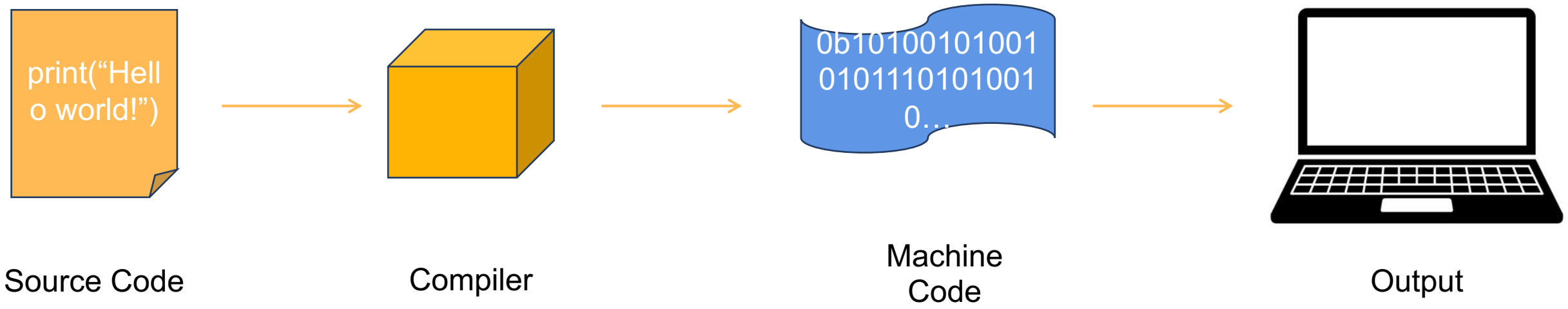
我们从Machine Code出发，确立了一系列指令集（电脑能理解的Operation）如加减乘除及内存存取
但是指令集本身还是二进制还是抽象

我们发明了另一套只有有限个词汇，符合特定格式的语法（Syntax）以及一个能理解这个语法并且将其翻译成Machine Code的程序 (Compiler/Interpreter)。

现在我们是不是只需要用这个语法表达我们的指令，就能让电脑执行了？
这样一套语法就是一个编程语言。

考点1. Compiler vs Interpreter





考点1. Compiling vs Interpreting Language

HD EDU

Before there is clear difference between two categories. Now not anymore!
For instance, python now have properties of both interpreting languages and compiling languages (.pyc).

Interpreting Language	Compiling Language
• Flexibility and Ease of Change • Local Execution	• Speed and Efficiency Control over Hardware
Python, MATLAB	Java, C

考点1. Compiling vs Interpreting Language

HD EDU

Compiled Languages:

- Speed and Efficiency:
  - Pros: Compiled languages are converted directly into machine code that the processor can execute. As a result, they tend to be faster and more efficient to execute than interpreted languages.
  - Cons: However, compiling the entire program requires more time upfront.
- Control Over Hardware:
  - Pros: Developers have more control over hardware aspects, such as memory management and CPU usage.
  - Cons: The need for a manual “build” step – you must compile the program before execution. Any changes require rebuilding the entire program.

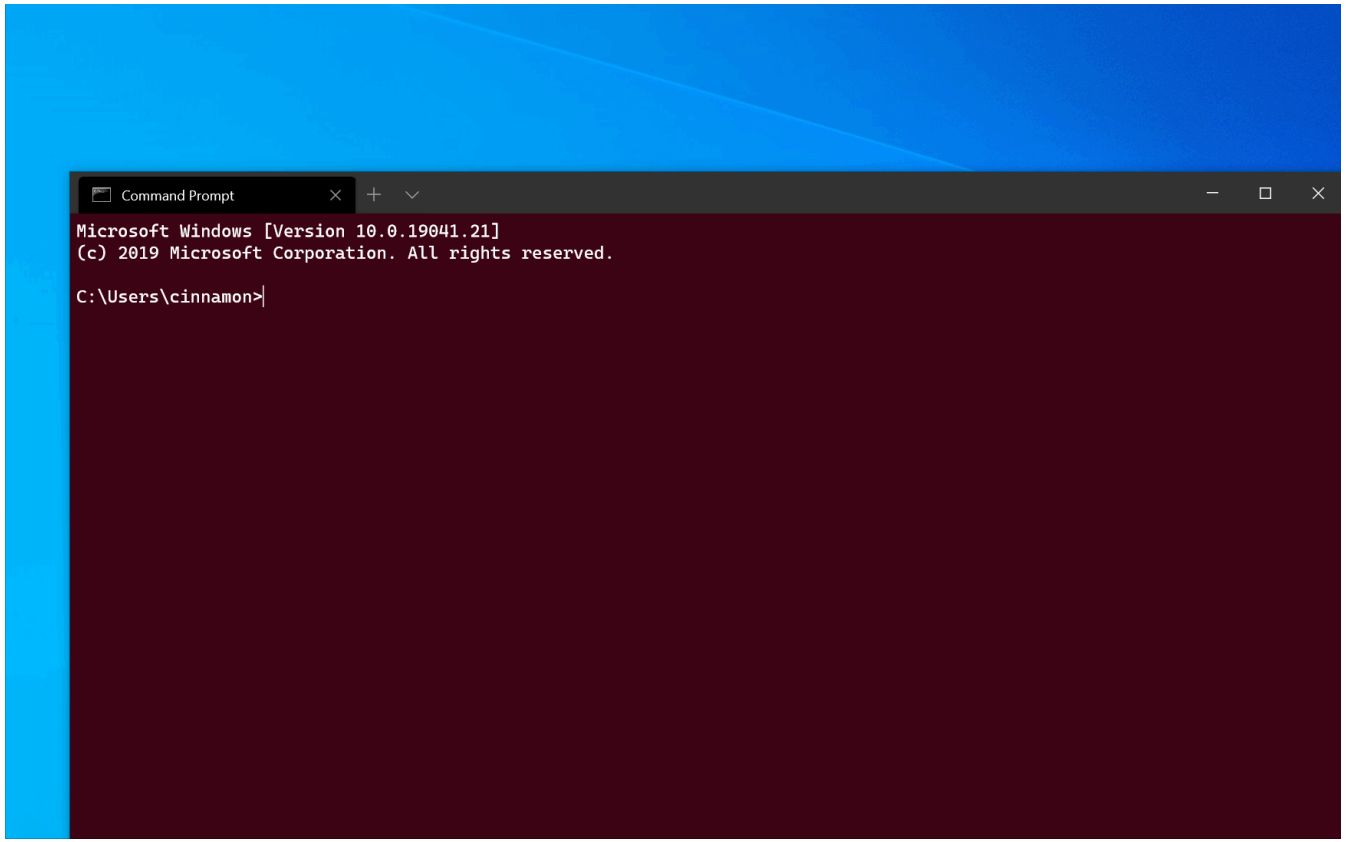
Interpreted Languages:

- Flexibility and Ease of Change:
  - Pros: Interpreters run through a program line by line and execute each command. If you need to make changes, you can modify the code directly without a separate compilation step.
  - Cons: Historically, interpreted languages were slower than compiled languages, but with just-in-time compilation, this gap is shrinking.
- Local Execution:
  - Pros: You don’t need to compile the entire program upfront. An interpreter analyzes and executes code on the fly.
  - Cons: Execution in an interpreter is less efficient than regular program execution.

考点2. Terminal/Shell

HD EDU

A terminal or shell is a text-based interface that allows users to interact with a computer's operating system through command-line commands.



Some tricks you should know:

- <Tab> for auto complete.
- <Up> and <Down> to browse through the input history.
- Ctrl + C to abort.



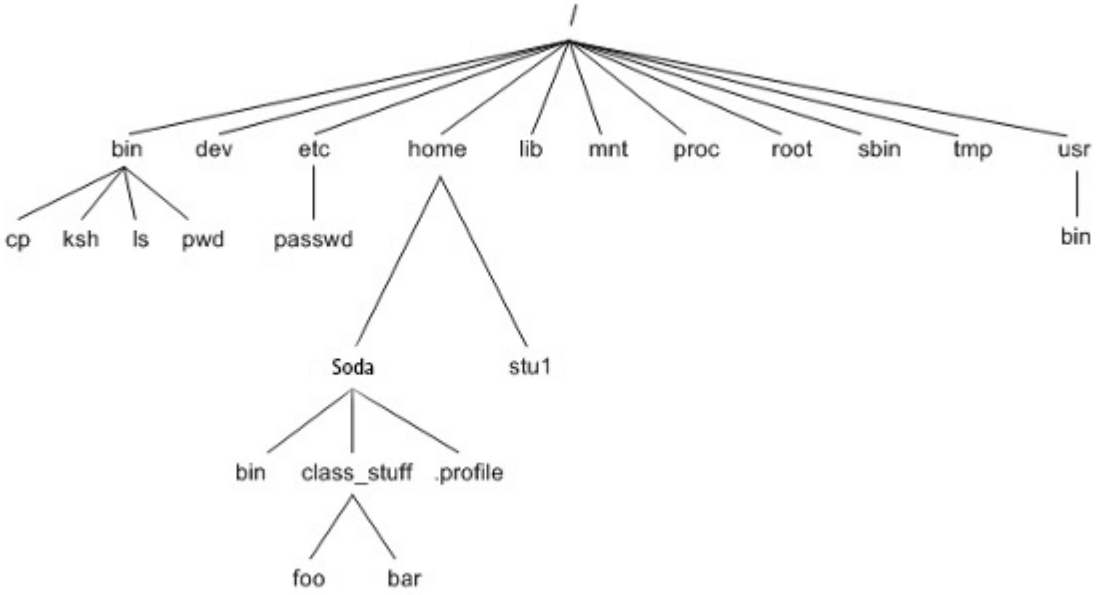
Windows Command Prompt is slightly different from Unix-based Terminals. Unfortunately, INFO1110 enforces the Unix-style.

考点2. Terminal/Shell - Commands

HD EDU

- Navigation Operation
  - `pwd` – Print working directory
  - `ls` – List items in the current directory
  - `mkdir <name>` - Make a directory with <name>
  - `rmdir <name>` - Remove an empty directory with <name>
  - `cd <name>` - Change directory to <name>
- File Operation
  - `touch <name>` – Create an empty file named <name> in the current directory
  - `mv <file> <location>` – Move the file with name <file> into <location>
  - `cp <file> <location>` – Copy the file with name <file> into <location>
  - `rm <name>` – Remove the file with name <name>
  - cat <name> - Print the content of the file
  - `python3 <name>` - Use Python3 interpreter to run the file with name <name>
- For your interest
  - `man <command>` – Get the 说明书 of a command
  - `ls -l` – List items in the current directory in vertical format
  - `ls -a` – List all items (including the hidden) in the current directory
  - `rm -r <name>` – Remove a directory named <name> containing files
  - `echo <text>` - Print <text> as a message

考点2. Unix File System



Absolute Path vs **Relative** Path

- Absolute: You can access the same file from any location. E.p. /bin/sh
- Relative: You can access the file based on current location e.g., soda/Downloads

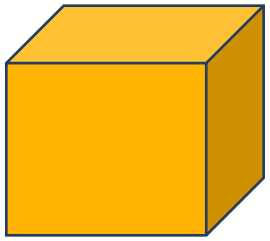
考点3. Built-in Functions – print( )

HD EDU

查任何与Built-in相关的内容，务必优先看Official Documentation! 这是由最懂Python的人编写的。

Function

Inputs
(Call functio时括号里放的内容)



Do something

Outputs
(return)

What is the difference between built-in keywords and built-in functions?

```
print("Hello")
print("Hello", "World", 123321)
print(a)
print(12)
print(-12.5)
print(True)
```

```
>>> b = print("Hello world!")
Hello world!
>>> print(b)
None
>>> type(b)
<class 'NoneType'>
```

考点3. Built-in Functions – print( )

HD EDU

print(**objects, sep=' ', end='\n', file=None, flush=False*)

Print *objects* to the text stream *file*, separated by *sep* and followed by *end*. *sep*, *end*, *file*, and *flush*, if present, must be given as keyword arguments.

All non-keyword arguments are converted to strings like `str()` does and written to the stream, separated by *sep* and followed by *end*. Both *sep* and *end* must be strings; they can also be `None`, which means to use the default values. If no *objects* are given, `print()` will just write *end*.

The *file* argument must be an object with a `write(string)` method; if it is not present or `None`, `sys.stdout` will be used. Since printed arguments are converted to text strings, `print()` cannot be used with binary mode file objects. For these, use `file.write(...)` instead.

Output buffering is usually determined by *file*. However, if *flush* is true, the stream is forcibly flushed.

Changed in version 3.3: Added the *flush* keyword argument.

```
>>> print("Hello world!", end='')
Hello world!>>> print("Hello world!")
Hello world!
```

- sep: 当print( )接收若干个inputs时，逗号会被sep设置的字符所取代。

```
>>> print(1, 2, 3, sep='*')
1*2*3
```

- end: print( )会自动在每次print结束后添加end设置的字符。默认为\n所以我们每次print都会自动换行。

Ref. https://docs.python.org/3.11/library/functions.html#print

考点4. Strings

HD EDU

String指的就是**文本**。
Syntax：Must be enclosed by ‘ or “ （一定要前后对应）

特殊字符：单靠键盘按键无法打出来的字符
Escape character: backslash \ 反斜杠
\n: Newline character等效于按回车键（换行）
\t: Tab characte等效于按Tab键（indent）
\\: 如果我只想print一个普通的反斜杠
\ + 引号：用于告诉Python，把该引号作为文本处理而不是关闭String的提示符。

```
>>> print('a\nb\t\\\'')
a
b \'
```

```
>>> print("'")
'
>>> print('"')
"
```

考点4. String Concatenation

HD EDU

String在Python里有加法和乘法，严格意义上来讲不叫运算。

```
>>> "abc" + "efg"
'abcefg'
>>> "abc" * 10
'abcabcabcabcabcabcabcabcabcabc'
>>> 10 * "abc"
'abcabcabcabcabcabcabcabcabcabc'
```

- 不能说String Addition or Multiplication!
- 乘号一定是跟一个正整数!

考点4. Strings – Multi-line String

HD EDU

当你想要打多行文本的时候会发现：

```
# soda @ Hilberts-MacBook-Pro in ~ [3:56:33]
$ cat temp.py
print("This is one line.
")

# soda @ Hilberts-MacBook-Pro in ~ [3:56:36]
$ python3 temp.py
  File "/Users/soda/temp.py", line 1
    print("This is one line.
          ^
SyntaxError: unterminated string literal (detected at line 1)
```

Python Multi-line string syntax —— 用三个单引号或双引号做提示符：

```
# soda @ Hilberts-MacBook-Pro in ~ [3:59:17]
$ cat temp.py
print("""This is one line.
This is another line.""")

# soda @ Hilberts-MacBook-Pro in ~ [3:59:20]
$ python3 temp.py
This is one line.
This is another line.
```

考点5. Variables

HD EDU

- What are variables?

  Variables are **named containers** to store information.

- Basic data types
  - int 整数
  - float~~小~~数 实数 (Not accurate)
  - bool (`True` or `False`)
  - str
  - (NoneType)
  - Etc.

When to use which data types?

- What is a valid variable name?

  Upper/Lowercase letters, numbers, special symbols?

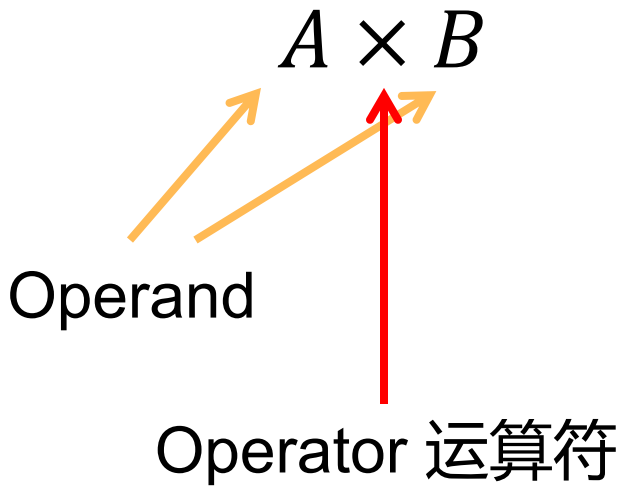
考点6. Data Types and Objects

HD EDU

- Data Type: Class类
  - 我能自称一类说明我有特别的性质（attribute）和特别的行为（method）。
- Object：一个类下的个体（instance）。一个object能包含多个别的object。（初涉，以后还会讲到）
- Method
  - 可以把它理解成一种特殊的只有一类objec有的function。
  - 使用Syntax格式为：object.<method name>(inputs)
  - 比如String的built-in methods可见：https://docs.python.org/3.11/library/stdtypes.html#string-methods

```
>>> "a".upper()
'A'
>>> "A".lower()
'a'
>>> "a".islower()
True
>>> "A".islower()
False
```

考点7. Expression Terminologies



- Unary Operator: These operators work with one operand.
- Binary Operator: These operators require two operands.
- Ternary Operator: The ternary operator is unique because it requires three operands.

考点7. Operators

HD EDU

- Mathematical

```
+ - * / // % ** math.sqrt() += -= pow() round() abs()
```

- Relational

```
== !=
> < >= <=
```

- Logical



做运算的前提是两者同类!

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

Truth Tables?

考点7. Bitwise Operators

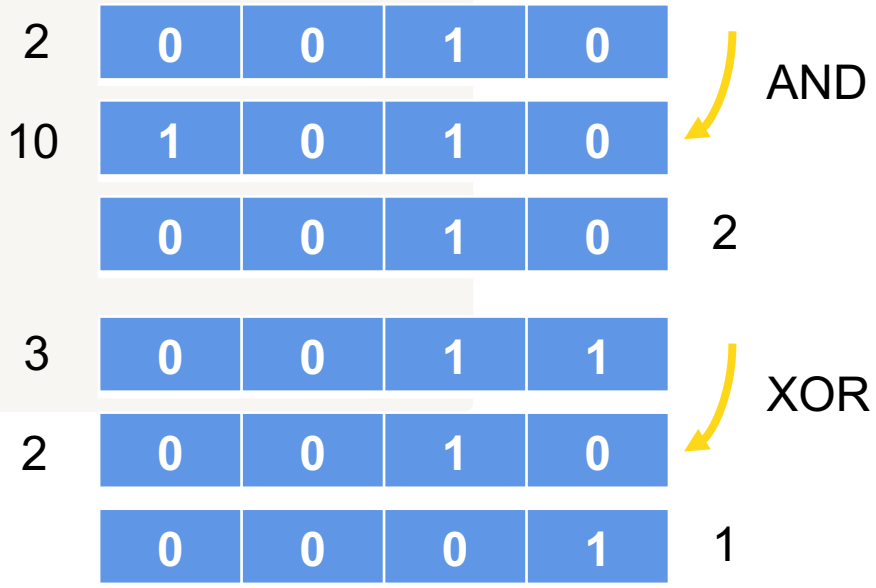
HD EDU

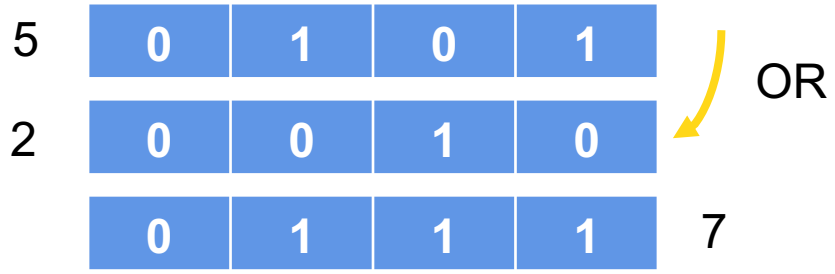
- There are three bitwise operators in Python:
  - & (ampersand) AND
  - ^ (hat) XOR
  - | (pipe) OR
- These operators operate on binary numbers.

XOR Truth Table

1	1	0
1	0	1
0	1	1
0	0	0

```
>>> 2 & 10
2
>>> 3 ^ 2
1
>>> 5 | 2
7
```





考点8. input( )

HD EDU

通用样式

```
<variable> = input("Message")
```

```
# temp.py
a = input("Enter your name: ")
print(a)
```

```
$ python3 temp.py
Enter your name: Hilbert
Hilbert
```

- 无论用户input什么，return type都是str
- input( ) 可以接受一个String作为argument，运行时会先print出这个string，同时在同一行接受input

考点9. type( ) and Type Conversion

HD EDU

- We can use `type( )` to find the datatype of certain data.
  - 每一个type function给的class type都是一个built-in functions
- We use:
  - `int( )` to convert data into an integer.
  - `str( )` to convert data into a string.
  - `float( )` to convert data into a float-point number.
- Conversion works only, when possible, otherwise **ERROR**!

```
>>> a = 2.4
>>> str(a)
'2.4'
>>> int(a)
2
>>> int("a")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'a'
```

考点10. Comments

HD EDU

- Write some **memos**
- **Deactivate** certain lines of code
  - Ctrl + / to quickly toggle commenting in mainstream code editors.

- Single-line Comment
  - Start with #
- Multi-line Comment
  - We use the same syntax as multi-line strings.

```
# This is a single line comment.
c = input("Enter your name: ")

"""
This is a multi-line comment.
Please have a look of the code below!
"""
a = 1
b = a + 100
```

考点11. String Formatting

HD EDU

❖ The most intuitive way —— String concatenation

```
>>> print("The price is", "$" + str(12.05) + ".")
The price is $12.05.
```

❖ `str.format()` method

Template(str).format(data_to_insert)

```
>>> "No.{} | Name: {} | Cost: ${:.2f}".format(10, "Soda", 3.87654321)
'No.10 | Name: Soda | Cost: $3.88'
```

❖ F-strings —— a simplified syntax of `str.format()`

```
>>> no = 10
>>> name = "Soda"
>>> cost = 3.87654321
>>> f"No. {no} | Name: {name} | Cost: ${cost:.2f}"
'No. 10 | Name: Soda | Cost: $3.88'
```



- 如何**保留Float小数点后若干位数**
- 如何**提供和使用多个插入位置**

❖ 最详解见 `str.format()`

❖ Formatted string literals



经典题型放送

Short Answer

HD EDU

(自编题)

```
x = 15.5 * 2
print(int(x))
print(x - 2)
x % 2
print(x)
```

```
import math
x = 12 / 4 > 21 - 16 and len(str(12)) < math.pi
print(x)
y = 12
y **= 2
print(y)
y % 4
print(y)
y //= 7
print(y)
```

What is printed when the code is executed?

Short Answer

HD EDU

Consider the below scenarios and answer which datatype should be used.

(自编题)

```
You're building a program to track the inventory of an online store. The quantities of items in stock can range from a few units to several thousand.
```

```
In a real-time gaming application, you need to store information about whether a player is currently logged in or not.
```

```
Imagine you're designing a voting system for a competition, and you need to keep track of the number of votes each participant receives.
```

```
You wanna store temperature information to perform analysis to predict future weather.
```

```
You perform web scrapping on a news website. What datatype should you use?
```

Short Answer

HD EDU

(自编题)

Describe an unary operator.

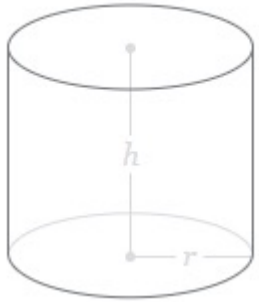
Describe whether the below variable names are valid:

- `123$`
- `A_aIs`
- `This is a variable`
- `2px`
- `a&b`

Code

(自编题)

$V = \pi r^2 h$



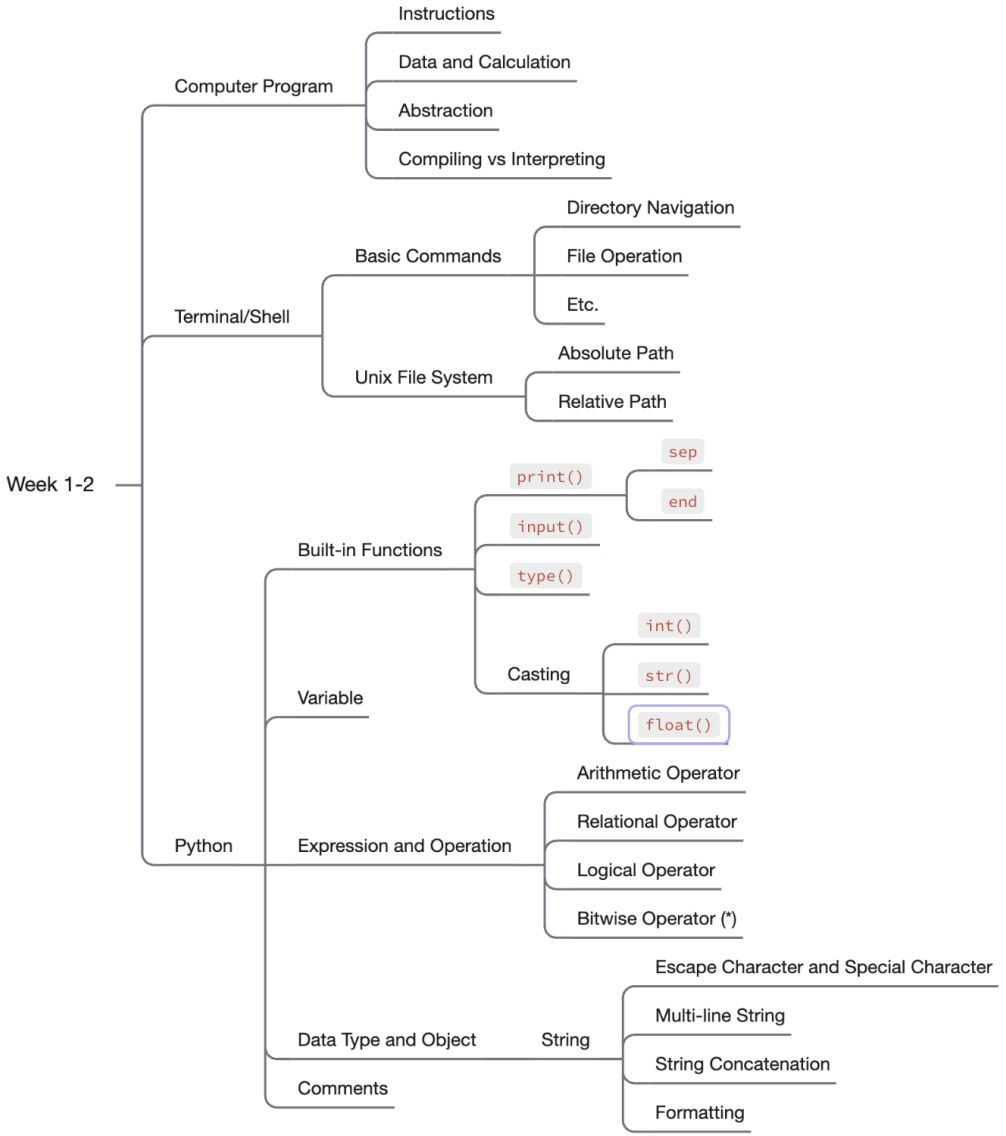
Write a program that calculate the volume of a cylinder round to 2 decimal places.

```
$ python cylinder.py
r = #10
h = #2
The volume of the cylinder is 628.32.
```



重难点总结

重难点总结





- 如果觉得前两周内容吸收的还可以，千万不要松懈喔！



下节课预告

下节课预告

HD EDU

- Conditionals
  - `if`
  - `else`
  - `elif`
- Iteration
  - `While`
  - `break`
  - `continue`



您每一份诚挚的问卷都会拯救一根濒临掉落的头发。

HD EDU



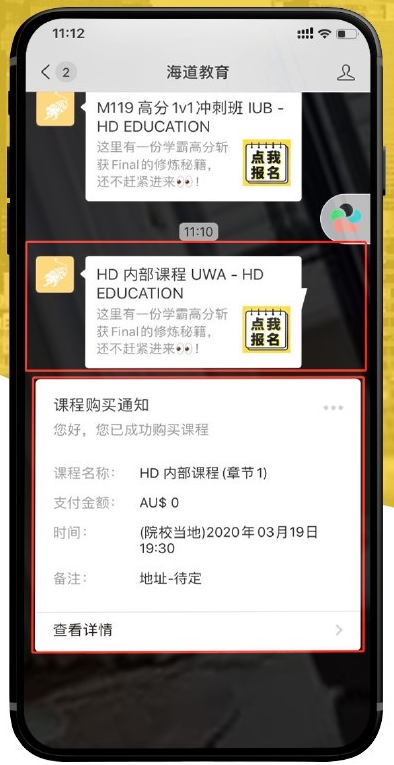
帮我开个张吧，求求你了！



课程结束后，如果您对课程或者服务的任何建议和意见

请给予我们提高和改进的机会，感谢您对 HD·EDUCATION 课程和服务的信任！

· 填写问卷操作流程 ·



第一步

关注【海道教育】服务号后
点击【购买通知】或【上课提醒】



第二步

点击【前往学习】



第三步

点击【去评价】就可以为课程进行评价

HD EDU



Q & A