



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Introduction to Computation (CS1602) · Lecture 10

Inheritance, Protocols and Modules

Instructor: Tao Huang

Part 4: Abstraction and Organization · Fall 2026



| Objects, Receivers and Contracts 对象与接口

- **Receiver 接收者**: each call selects an object
- **State and result 状态与结果**: observe both
- **Contract 约定**: supported operations and failures

| Reusable Reports 可复用的说明

Records 记录	Common 共用	Different 差异
Ordinary members	Name and report interface	Role-specific description
Special roles	The same supported operation	Additional state or behavior

Quiz 1 · A Saved Method



扫码作答 · 1 题

1 保存方法引用后的状态观察

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/10/>



| Lecture Structure

- | | | |
|----------|-----------------------|---|
| 1 | Reuse 复用 | Inheritance and behavioral interfaces |
| 2 | Protocols 协议 | Representations, operators and containers |
| 3 | Modules 模块 | Files, bindings and entry points |
| 4 | Environment 环境 | Lookup, installation and execution |
-

I • Inheritance and Behavioral Interfaces

Classes already organize state.

How can related implementations be reused?

| A Base Class and Its Subclass 基类与子类

Excerpt · complete definition in notes

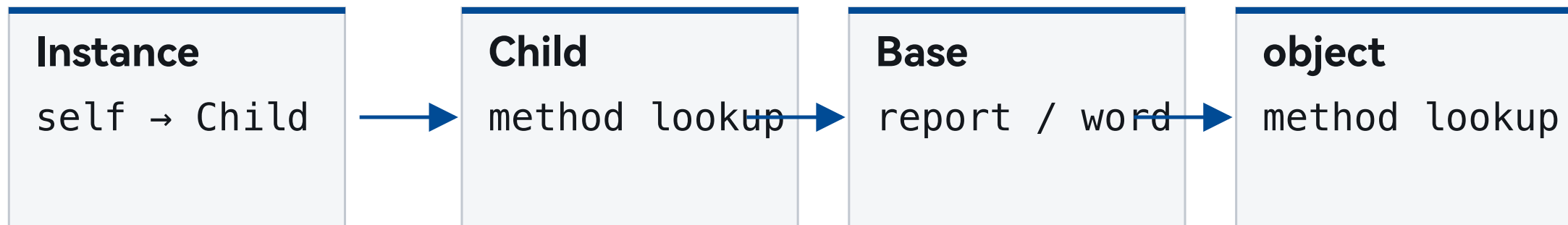
```
class Animal:
    def __init__(self, name: str) -> None:
        self.name = name

class Dog(Animal):
    def speak(self) -> str:
        return "woof"
```

- **Base class 基类:** Animal
- **Subclass 子类:** Dog
- Inherited initialization uses this instance

[Complete runnable example 完整程序](#)

| Inherited Initialization 继承的初始化



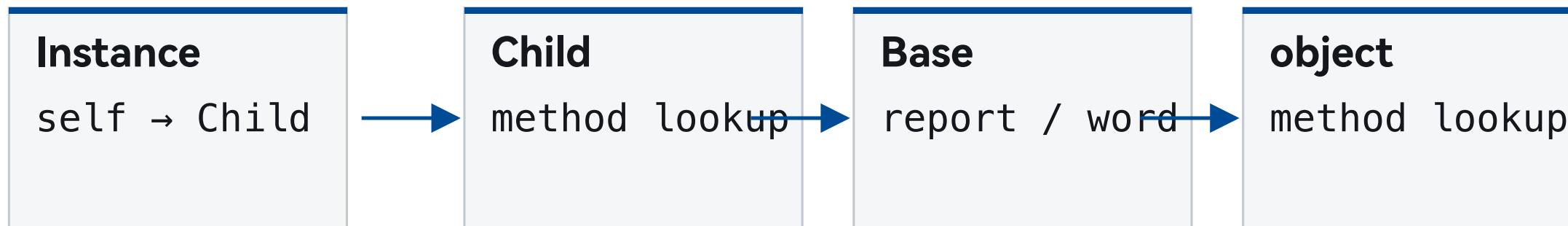
Ordinary method lookup 方法查找; self stays the actual instance.

| Method Overriding 方法重写

Which implementation does the receiver select? 先预测。

```
1  class Base:
2      def word(self):
3          return "base"
4      def report(self):
5          return self.word()
6
7  class Child(Base):
8      def word(self):
9          return "child"
10
11  print(Child().report())
```

| Method Lookup 方法查找



Ordinary method lookup 方法查找; self stays the actual instance.

| The Receiver in Reused Code 复用代码的接收者

code

```
1 class Point:
2     def __init__(self, x):
3         self.x = x
4     def read(self):
5         return self.x
6
7 a = Point(3)
8 b = Point(4)
9 f = a.read
10 a.x = 8
11 print(f(), b.read())
```

output

8 4

| Calls Inside an Inherited Method 继承方法内调用

Does Base.report force Base.word?

```
1  class Base:
2      def word(self):
3          return "base"
4      def report(self):
5          return self.word()
6
7  class Child(Base):
8      def word(self):
9          return "child"
10
11  print(Child().report())
```

| An Overridden Initializer 重写的初始化

Which required field is missing? 缺哪个字段?

```
1 class Base:
2     def __init__(self, value):
3         self.value = value
4
5 class Child(Base):
6     def __init__(self, value):
7         self.extra = value
8
9 print(Child(3).value)
```

| super() and Required State 必需状态

Excerpt · complete definition in notes

```
class Student(Person):  
    def __init__(self, name, age, school):  
        super().__init__(name, age)  
        self.school = school
```

Actual complete-program
output 实跑输出

Alex, 18, SJTU

- Establish required name / age
- Then establish school
- In this case, super finds Person

[Complete runnable example 完整程序](#)

| Reuse and Override 复用与重写

Excerpt · complete definition in notes

```
def give_raise(self, fraction, bonus=0.1):  
    super().give_raise(fraction + bonus)
```

Actual complete-program
output 实跑输出

Chen 115.00

- Reuse the existing pay update
- The actual receiver is retained
- Base.report selected child in the previous example

[Complete runnable example 完整程序](#)

| Polymorphism and Duck Typing 多态与鸭子类型

code

```
1 class Circle:
2     def area(self) -> float:
3         return 3.141592653589793
4 class Square:
5     def area(self) -> float:
6         return 4.0
7
8 total = 0.0
9 for shape in [Circle(), Square()]:
10     total += shape.area()
11 print(f"{total:.4f}")
```

output

7.1416

Quiz 2 · Dispatch and Initialization



扫码作答 · 2 题

1 父方法内的self调用 2 重写初始化与缺字段

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/10/>

| Quiz Review · Reuse and State

Question	Actual result	Reason
Q2	child	Inherited code retains the Child receiver
Q3	AttributeError	No assignment established value

| 方法查找和状态建立分开核对。



II · Built-in Operation Protocols

Methods define behavior.

How can ordinary Python syntax select it?

| String Representations 字符串表示

Which representation does each operation use?

```
1  class Point:
2      def __init__(self, x: float, y: float) -> None:
3          self.x, self.y = x, y
4      def __str__(self) -> str:
5          return f"({self.x}, {self.y})"
6      def __repr__(self) -> str:
7          return f"Point({self.x}, {self.y})"
8
9  p = Point(3, 4)
10 print(p)
11 print(repr(p))
12 print([p])
```

| __str__ and __repr__

code

```
1 class Point:
2     def __init__(self, x: float, y: float) -> None:
3         self.x, self.y = x, y
4     def __str__(self) -> str:
5         return f"({self.x}, {self.y})"
6     def __repr__(self) -> str:
7         return f"Point({self.x}, {self.y})"
8
9 p = Point(3, 4)
10 print(p)
11 print(repr(p))
12 print([p])
```

output

```
(3, 4)
Point(3, 4)
[Point(3, 4)]
```



| Supported and Unsupported Operands 操作数范围

Return/action	Meaning 含义
Result	This implementation supports the operands
NotImplemented	Let operator dispatch try other applicable paths
raise NotImplementedError	An exception, not operator fallback

| 区分返回哨兵与抛出异常。

| Addition and New Results 加法与新结果

Excerpt · complete definition in notes

```
def __add__(  
    self,  
    other: "Vector",  
) -> "Vector":  
    if not isinstance(other, Vector):  
        return NotImplemented  
    x = self.x + other.x  
    y = self.y + other.y  
    return Vector(x, y)
```

Actual complete-program output 实跑输出

Vector(4, 6)

- This + creates a new Vector
- The operands remain unchanged
- Check support before reading fields

[Complete runnable example 完整程序](#)

| Reflected Operations 反射运算

Excerpt · complete definition in notes

```
def __rmul__(  
    self,  
    scalar: float,  
) -> "Vector":  
    return self * scalar
```

Actual complete-program
output 实跑输出

```
Vector(3, 6)  
Vector(3, 6)
```

- Vector $\times$ scalar: `__mul__`
- Scalar $\times$ Vector: `__rmul__`
- Reflected dispatch has type-dependent rules

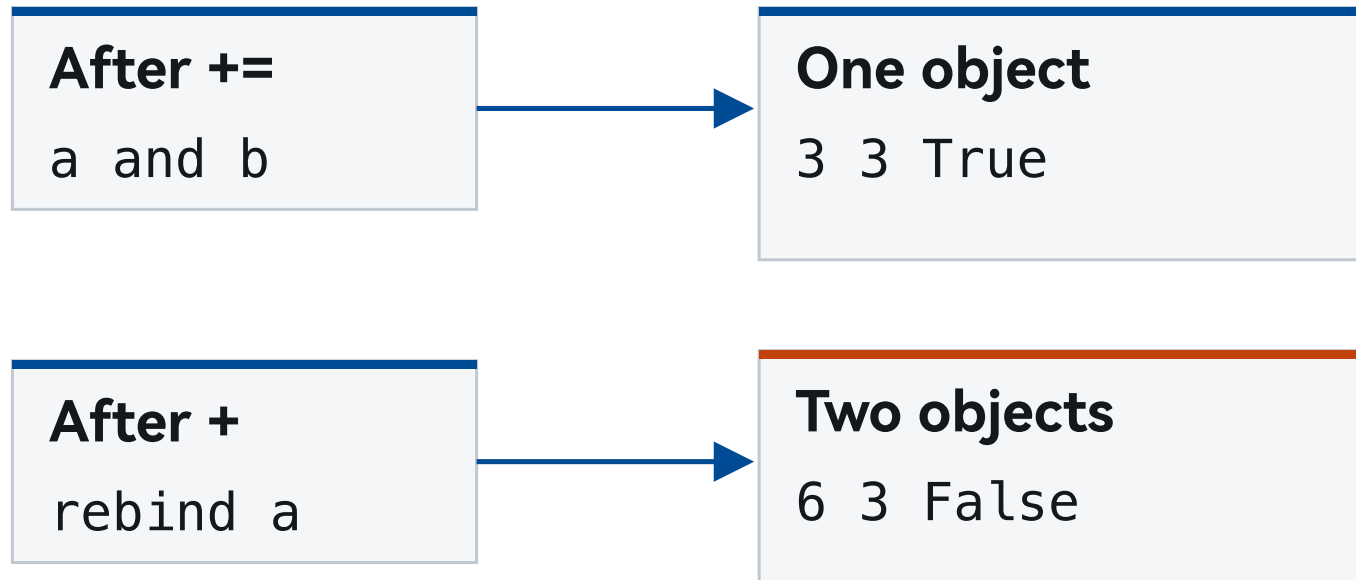
[Complete runnable example 完整程序](#)

| Augmented Assignment 增强赋值

Does the alias observe the changed object?

```
1  class Value:
2      def __init__(self, n: int) -> None:
3          self.n = n
4      def __add__(self, other: "Value") -> "Value":
5          return Value(self.n + other.n)
6      def __iadd__(self, other: "Value") -> "Value":
7          self.n += other.n
8          return self
9
10 a = Value(1)
11 b = a
12 a += Value(2)
13 print(a.n, b.n, a is b)
14 a = a + Value(3)
15 print(a.n, b.n, a is b)
```

| Object Mutation and Rebinding 修改与重新绑定



This implementation only 按本例实现观察; += need not preserve identity.

| Length, Indexing and Membership 长度与下标

Excerpt · complete definition in notes

```
def __getitem__(self, index: int) -> str:  
    return self._songs[index]
```

```
def __setitem__(  
    self,  
    index: int,  
    value: str,  
) -> None:  
    self._songs[index] = value
```

Actual complete-
program output 实跑
输出

```
2 changed True  
first  
changed
```

- `len` → `__len__`
- Membership → `__contains__`
- Negative indices and `IndexError` follow this list contract

[Complete runnable example](#) 完整程序

| Sequence-Style Iteration 序列式迭代

What index is requested after the last item?

```
1 class Values:
2     def __init__(self):
3         self.items = ["A", "B"]
4     def __getitem__(self, index):
5         print("index", index)
6         return self.items[index]
7
8 print(list(Values()))
```

| Equality and Identity 相等与身份

code

```
1 class Item:
2     def __init__(self, value):
3         self.value = value
4     def __eq__(self, other):
5         if not isinstance(other, Item):
6             return NotImplemented
7         return self.value == other.value
8
9 a = Item(2)
10 b = Item(2)
11 c = a
12 print(a == b, a is b, a is c)
```

output

True False True

I Identity and Type Checks 身份与类型

Question 问题

Operation 操作

Same object?

`a is b`

Equal by the contract?

`a == b`

The None singleton?

`x is None`

Accept subclasses?

`isinstance(x, Base)`

Exact runtime type?

`type(x) is Base`

Quiz 3 · Equality and Identity



扫码作答 · 1 题

1 相等字段、不同实例与别名

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/10/>

I Quiz Review · Values and Aliases

code

```
1 class Item:
2     def __init__(self, value):
3         self.value = value
4     def __eq__(self, other):
5         if not isinstance(other, Item):
6             return NotImplemented
7         return self.value == other.value
8
9 a = Item(2)
10 b = Item(2)
11 c = a
12 print(a == b, a is b, a is c)
```

output

True False True

HALFWAY

Break

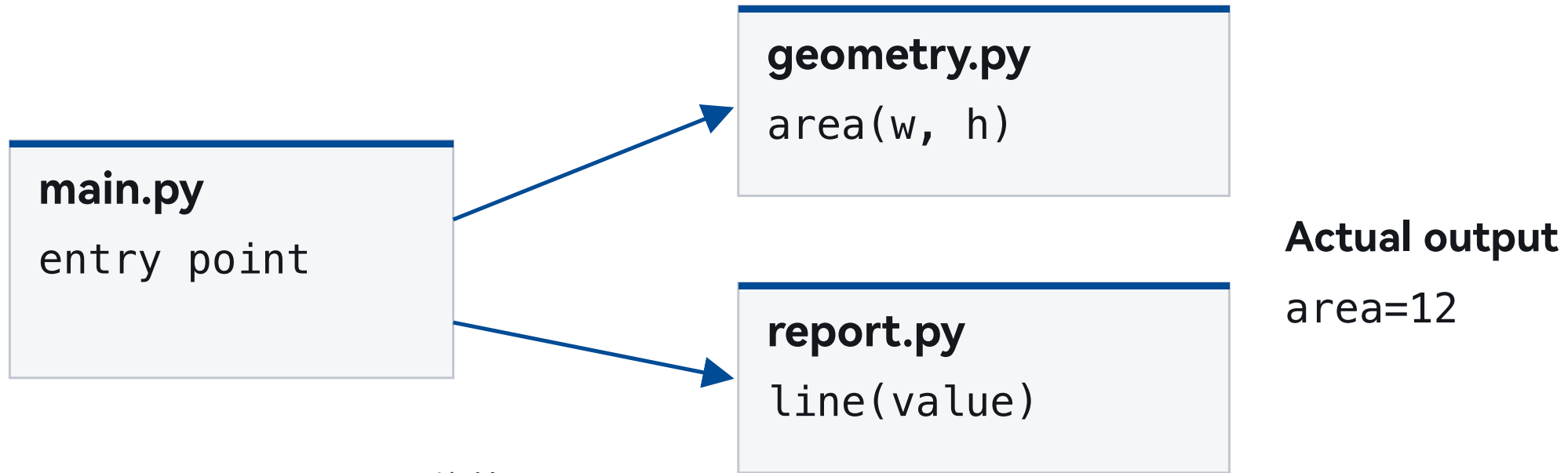
Next: modules, environment and execution.

III · Modules and Name Bindings

Classes are organization units.

How can files share definitions without sharing every name?

| A Small Program Across Files 多文件程序



Dependency edges 导入依赖; not threads or simultaneous calls.



| Modules as Objects 模块对象

code

```
1 import math
2 print(math.sqrt(16))
3 print(type(math).__name__)
```

output

```
4.0
module
```

| Explicit Imports 显式导入

Form 写法

Access 访问

```
import math
```

```
math.sqrt(16)
```

```
import math as m
```

```
m.sqrt(16)
```

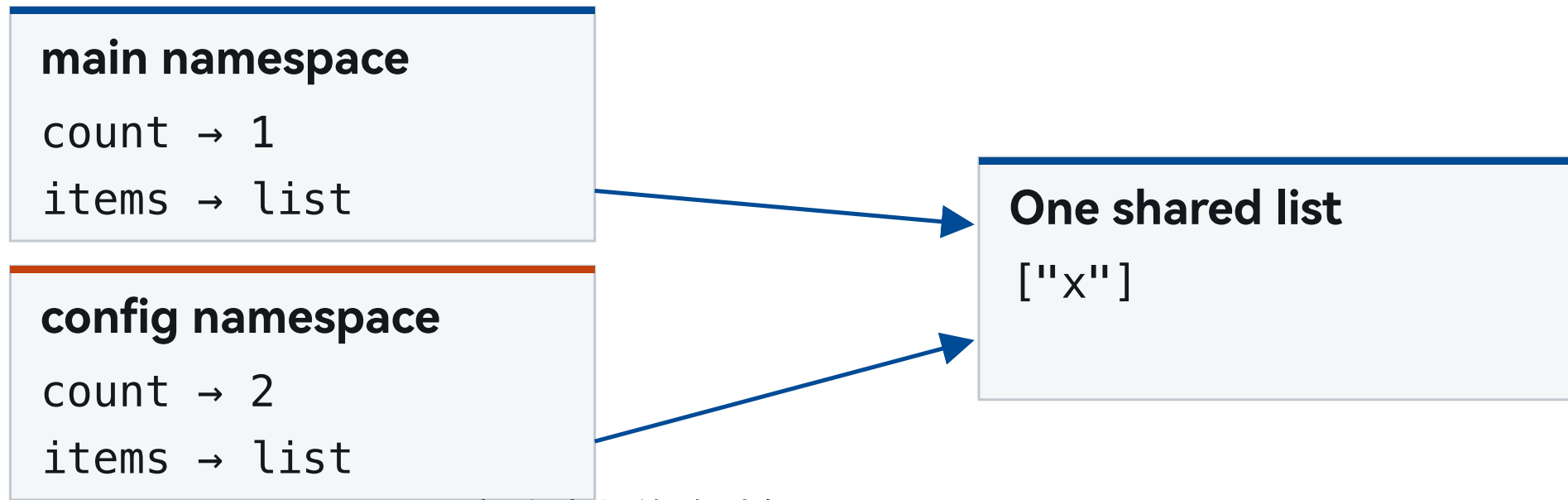
```
from math import sqrt
```

```
sqrt(16)
```

```
from math import sqrt as root
```

```
root(16)
```

| Module and Member Bindings 模块与成员绑定



Rebinding vs mutation 重新绑定与修改对象; observed: 1 2 ['x']



| Import Style and Exports 导入风格

- **Explicit names 显式名称**: keep origins readable
- **Separate groups 导入分组**: standard, third-party, local
- **Avoid wildcard imports** in ordinary course code

| 来源与导出范围须明确，不把风格建议当语法限制。

| Module Initialization 模块初始化

helper.py

```
print("module body")  
def action():  
    print("function body")
```

main.py

```
import sys  
import helper  
import helper  
same = sys.modules["helper"]  
print("cached", same is helper)  
helper.action()
```

| Multi-file example 多文件示例 · save and run locally.

| Repeated Imports and sys.modules 导入缓存

helper.py

```
print("module body")  
def action():  
    print("function body")
```

main.py

```
import sys  
import helper  
import helper  
same = sys.modules["helper"]  
print("cached", same is helper)  
helper.action()
```

Actual Python 3.13 output · real files

```
module body  
cached True  
function body
```

| Multi-file example 多文件示例 · save and run locally.

| Mutation and Module Rebinding 模块更新

config.py

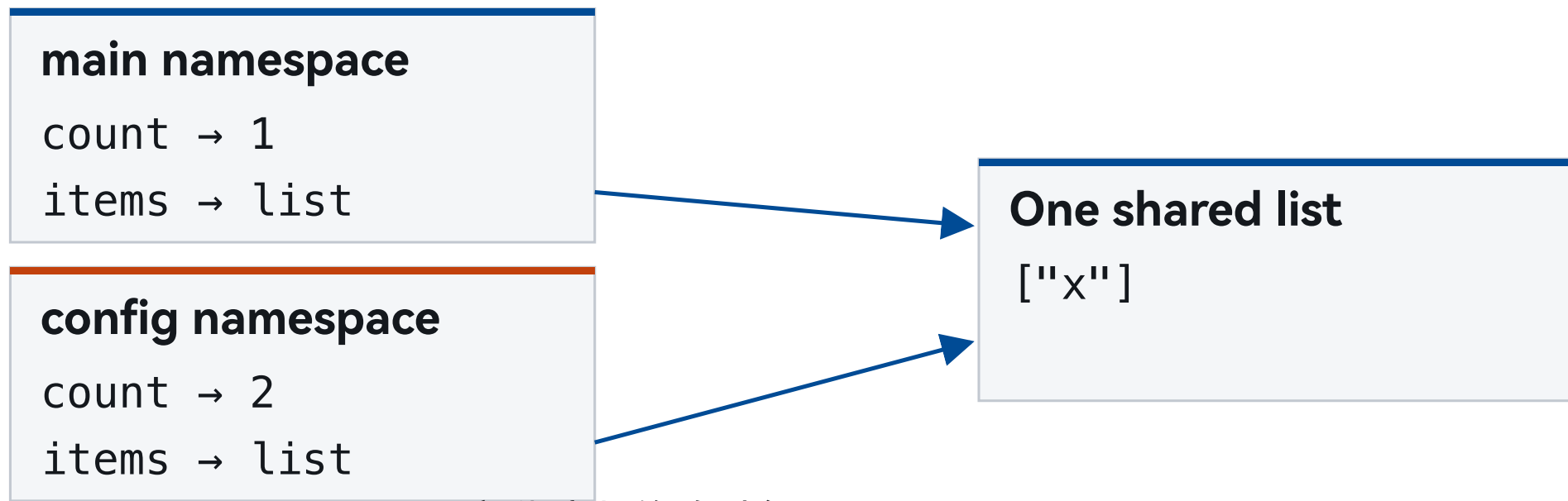
```
count = 1  
items = []
```

main.py

```
import config  
from config import count, items  
config.count = 2  
config.items.append("x")  
print(count, config.count, items)
```

| Multi-file example 多文件示例 · save and run locally.

| Imported References 导入引用



Rebinding vs mutation 重新绑定与修改对象; observed: 1 2 ['x']

| Module Names and Entry Points 模块名与入口

tools.py

```
print("load")  
if __name__ == "__main__":  
    print("demo")
```

main.py

```
import tools  
import tools  
print("main")
```

| Multi-file example 多文件示例 · save and run locally.

| Direct Execution and Import 直接与导入

Entry 入口

Actual output 实际输出

Run tools.py

load
demo

Run main.py

load
main

| main guard只控制该块，其他顶层副作用仍会发生。

Quiz 4 · Imported State and Entry Guards



扫码作答 · 2 题

- 1 模块数字改绑与共享列表 2 普通导入缓存与入口guard

Submit this round once. 按已教规则推理。

<https://taohuang.info/cs1602/zh/quiz/10/>



I Quiz Review · Imports and State

Question	Actual output
----------	---------------

Q5	1 2 ['x']
----	-----------

Q6	load main
----	--------------

| Regular Packages 普通包

`mypackage/geometry.py`

```
def area(w, h):  
    return w*h
```

`main.py`

```
from mypackage import geometry  
print(geometry.area(3, 4))
```

Actual Python 3.13 output · real files

12

| Multi-file example 多文件示例 · save and run locally.

IV · Environment and Execution

Correct source can still load another module.

Which interpreter and source are actually selected?



| Module Search 模块搜索

- | | | |
|----------|--------------------|--|
| 1 | Cache 缓存 | Check an existing module entry. |
| 2 | Finders 查找器 | Built-in, frozen and other import finders. |
| 3 | Paths 路径 | Top-level path-based lookup uses sys.path. |
-

| 启动方式与缓存影响选择，当前目录不永远第一。

| A Same-Named Local File 同名本地文件

random.py

```
print("local random")
```

main.py

```
import random  
print(random.randint(1, 3))
```

| Multi-file example 多文件示例 · save and run locally.

I Identifying the Actual Environment 确认环境

Previous prediction · actual result

```
local random
```

```
AttributeError: module 'random' has no attribute 'randint' (consider  
renaming '/privaterandom.py' since it has the same name as the  
standard library module named 'random' and prevents importing that  
standard library module)
```

- **sys.executable**: selected interpreter
- **module.__file__**, when present: source
- **python -m pip --version**: matching installer

| Virtual Environments 虚拟环境

- `python -m venv .venv`
- **Use that environment's interpreter and pip.**
- **Dependency isolation**, not a security sandbox

| 确认解释器与安装器对应，再处理依赖。



| Installing Compatible Distributions 安装兼容包

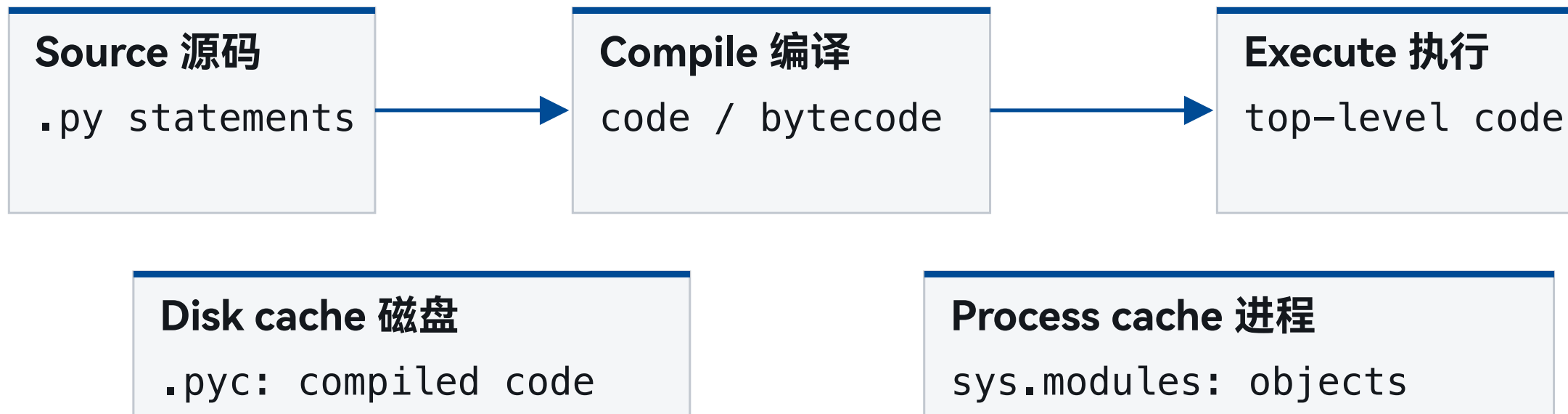
Task	Command pattern
Install	<code>python -m pip install PACKAGE</code>
Inspect	<code>python -m pip list</code>
Upgrade	<code>python -m pip install --upgrade PACKAGE</code>
Remove	<code>python -m pip uninstall PACKAGE</code>
Local wheel	<code>python -m pip install compatible.whl</code>



| The Scientific Python Ecosystem 科学计算生态

Library	Typical use
NumPy	Arrays and numerical operations
Matplotlib	Visualization
SciPy	Scientific algorithms
pandas	Tabular data
pytest	Behavioral tests

| CPython · Source and Execution 源码与执行



CPython 3.13 model 缓存层不同; new processes still initialize modules.

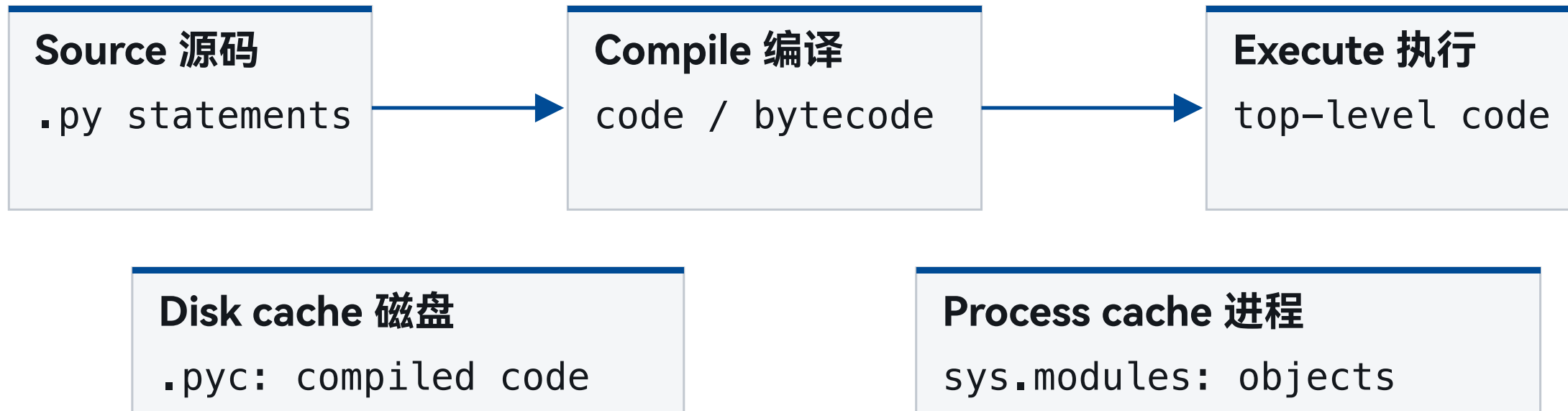
| Bytecode Operations 字节码操作

Actual output 实跑输出

3	RESUME	0
4	LOAD_FAST_LOAD_FAST BINARY_OP RETURN_VALUE	1 (a, b) 0 (+)

- Load operands 取值
- Add 运算
- Return 返回
- Version-specific instructions

| Two Kinds of Caching 两层缓存



CPython 3.13 model 缓存层不同; new processes still initialize modules.

| Cached Code and Runtime Work 缓存与运行

Will another fresh process print the top-level message?

```
1  # helper.py
2  print("helper")
3  value = 3
4
5  # main.py
6  import helper
7  print(helper.value)
```

| Implementation and Cost 实现与代价

Process	Actual output
---------	---------------

1	helper 3
---	-------------

2	helper 3
---	-------------

- New processes still initialize modules.
- A cache does not hide source or prove faster code.



| Solo Assignment 2 · Counting Primes

- **Five methods:** brute force, $\sqrt{n}$, $6k \pm 1$, sieve, Miller–Rabin
- **First verify**, then measure a feasible workload
- **Three solo assignments: 15% total; this one: 5%**

| Due: 1 Dec, 23:59 GMT+8. 本周任务是素数，大整数第12周启动。

| Lab 10 · Interfaces and Module Organization

Part	Work	Raw points
A	Shapes / accounts / Money / Deck	16 auto + 8 staff
B	Real module project + evidence	8 auto + 4 staff
C	Environment and debugging	0-point record

| Reuse, Protocols and Evidence

- Inherited state and behavior
- Supported operands and failures
- Imported bindings and side effects
- Interpreter and cache conditions

| Complexity, Exceptions and Testing

- How much work does the implementation perform?
- How should failures be handled?
- What evidence supports correctness?



Reference 参考

Complete programs and extended mechanisms.

After class, not another required lecture.

| Person and Manager · Complete Example

Excerpt · complete definition in notes

```
class Manager(Person):  
    def __init__(self, name, pay):  
        super().__init__(name, "manager", pay)  
    def give_raise(  
        self,  
        fraction,  
        bonus=0.1,  
    ):  
        super().give_raise(fraction + bonus)
```

Actual complete-
program output 实跑
输出

Chen 115.00

- name / job / pay
- get_last_name / give_raise
- Same-name def rebinding is not argument overloading

[Complete runnable example 完整程序](#)

I Required State and Compatible Interfaces

- Inherited names do not create instance fields.
- Override signatures and outcomes need a contract.
- `NotImplementedError` is raised when called.
- Initialization order depends on state requirements.

| 字段存在、行为兼容与是否抽象是不同问题。

| Cooperative Multiple Inheritance 合作式多继承

Actual output 实跑输出

```
['Combined', 'Left', 'Right', 'Root', 'object']  
['Combined', 'Left', 'Right', 'Root']
```

- **C3 MRO**, not BFS
- **super continues after the current class**
- Cooperative methods must call the next path

| MRO Constraints 方法顺序约束

code

```
1 class A: pass
2 class B(A): pass
3 class C: pass
4 class D(B, C): pass
5 print([c.__name__ for c in D.__mro__])
```

output

```
['D', 'B', 'A', 'C', 'object']
```



| Inheritance and Composition 继承与组合

Choice

Behavioral requirement

Subclass

Preserve what clients depend on

Delegate / compose

Reuse another object without claiming its full interface

Mutable Square/Rectangle

Independent width/height updates may conflict with equal sides

| Name Mangling Across Subclasses

Excerpt · complete definition in notes

```
def __init__(self, values):  
    self.items = []  
    self.__update(values)  
  
__update = update
```

Actual complete-program
output 实跑输出

```
[1, 2, ('a', 3)]
```

- `__update` → `_Mapping_update`
- Subclass update may have another signature
- Name mangling is not access control

[Complete runnable example](#) 完整程序

| Introspection Tools 内省工具

Actual output 实跑输出

```
True False
True
{'name': 'Rex'}
['Dog', 'Animal', 'object']
['name']
```

- Type vs isinstance
- Class MRO and ordinary instance dict
- dir is a diagnostic aid

| Functions and Code Objects 函数与代码对象

code

```
1  def add(a, b):  
2      return a + b  
3  
4  print(add.__name__)  
5  print(add.__code__.co_varnames)  
6  print(add.__code__.co_names)  
7  print(add.__code__.co_consts)
```

output

```
add  
( 'a', 'b' )  
( )  
(None, )
```

| Rich Comparisons and Sorting 富比较

Symbol

Method

< / <=

__lt__ / __le__

== / !=

__eq__ / __ne__

> / >=

__gt__ / __ge__

Unsupported equality

Try applicable paths, then identity

Equality and hash

Equal keys must have equal hashes

| Readable and Informative Representations

Excerpt · complete definition in notes

```
def __repr__(self) -> str:  
    return f"Point({self.x}, {self.y})"
```

Actual complete-program
output 实跑输出

```
(3, 4)  
Point(3, 4)  
[Point(3, 4)]
```

- str may fall back to repr
- Containers display repr
- Both must return str

[Complete runnable example 完整程序](#)

| Container Operations 容器操作

Excerpt · complete definition in notes

```
def __setitem__(  
    self,  
    index: int,  
    value: str,  
    ) -> None:  
    self._songs[index] = value
```

Actual complete-program
output 实跑输出

```
2 changed True  
first  
changed
```

- Negative indices
- Length and membership
- IndexError ends sequence fallback

[Complete runnable example 完整程序](#)

| Binary and Reflected Dispatch

Excerpt · complete definition in notes

```
def __mul__(
    self,
    scalar: float,
) -> "Vector":
    if not isinstance(scalar, (int, float)):
        return NotImplemented
    return Vector(self.x * scalar, self.y * scalar)

def __rmul__(
    self,
    scalar: float,
) -> "Vector":
    return self * scalar
```

Actual complete-
program output 实跑
输出

```
Vector(3, 6)
Vector(3, 6)
```

- A right subtype can have priority
- Same-type reflection is not always retried
- Implicit methods are usually type-level

[Complete runnable example 完整程序](#)

| In-Place Operations and Aliases

Actual output 实跑输出

```
3 3 True
6 3 False
```

- This iadd returns self
- Another iadd may return a new object
- Missing/unsupported iadd can fall back

| Triangle and Vector Protocol Examples

Excerpt · complete definition in notes

```
def __lt__(  
    self,  
    other: "Triangle",  
) -> bool:  
    if not isinstance(other, Triangle):  
        return NotImplemented  
    return self.area() < other.area()
```

Actual complete-program
output 实跑输出

```
True 3  
4
```

- This Triangle orders by area
- Indexing selects edges
- Invalid updates preserve old edges

[Complete runnable example 完整程序](#)

| Point, Complex and Mapping Examples

code

```
1  def isomorphic(s: str, t: str) -> bool:
2      if len(s) != len(t):
3          return False
4      forward, reverse = {}, {}
5      for a, b in zip(s, t):
6          if a in forward and forward[a] != b:
7              return False
8          if b in reverse and reverse[b] != a:
9              return False
10         forward[a], reverse[b] = b, a
11     return True
12
13 print(isomorphic("egg", "add"))
14 print(isomorphic("foo", "bar"))
```

output

```
True
False
```

| A Complete Multi-File Program

geometry.py

```
"""Geometry operations."""  
def area(w, h):  
    return w*h
```

main.py

```
import geometry  
import report  
value = geometry.area(3, 4)  
print(report.line(value))
```

Actual Python 3.13 output · real files

```
area=12
```

| Multi-file example 多文件示例 · save and run locally.

| Initialization Order and Circular Imports

a.py

```
import b  
value = 1
```

b.py

```
from a import value
```

Actual Python 3.13 output · real files

```
ImportError: cannot import name 'value' from 'a' (consider renaming  
'/privatea.py' if it has the same name as a library you intended to  
import)
```

| Multi-file example 多文件示例 · save and run locally.

| importlib.reload

settings.py

```
print("load")  
value = 1
```

main.py

```
import importlib  
import settings  
from settings import value  
settings.value = 9  
print(value, settings.value)  
importlib.reload(settings)  
print(value, settings.value)
```

Actual Python 3.13 output · real files

```
load  
1 9  
load  
1 1
```

| Multi-file example 多文件示例 · save and run locally.

| Aliases and Future Statements 别名与future

code

```
1  from __future__ import annotations
2
3  class Node:
4      def copy(self) -> Node:
5          return Node()
6
7  print(Node().copy().__class__.__name__)
```

output

Node

| Search Paths and Environment Details

Mechanism

Condition

sys.modules

Already initialized process objects

Import finders

Built-in / frozen / custom and path-based

Path entries

Script context / PYTHONPATH / site / zip

Launch options

Normal script, -m and -I differ

| Packages and Exports 包与导出

Item	Meaning
Regular package	<code>__init__.py</code> establishes initialization
Namespace package	A distinct package mechanism
<code>__all__</code>	Controls intended wildcard exports
Relative imports	Need a suitable package context

I pip, Wheels and Compatible Versions

Concern

Check

Interpreter

`python -m pip` targets that environment

Distribution

Python / ABI / platform compatibility

Sources

PyPI or a known compatible local wheel

Operations

`Install / list / upgrade / uninstall`

| Library Reference 库参考

Libraries

Scope

NumPy / Matplotlib

Arrays / visualization

SciPy / pandas

Scientific algorithms / tables

Scrapy / BeautifulSoup

Web processing

Pygame / pytest

Games / tests

| Implementations and Toolchains 实现与工具链

Examples

Category

CPython / PyPy / Jython

Python implementations

Cython / Nuitka

Compilation tools

Native / bytecode / JIT

Compilation and execution can be combined

| Explicit Compilation 显式编译

code

```
1 source = "answer = 2 + 3"
2 code = compile(source, "trusted.py", "exec")
3 namespace = {}
4 exec(code, namespace)
5 print(namespace["answer"])
```

output

5



dis and Function Metadata 字节码与元数据

code

```
1  def add(a, b):  
2      return a + b  
3  
4  print(add.__name__)  
5  print(add.__code__.co_varnames)  
6  print(add.__code__.co_names)  
7  print(add.__code__.co_consts)
```

output

```
add  
( 'a', 'b' )  
( )  
(None, )
```

I .pyc Cache Conditions 缓存条件

- **Validity and version tags** matter
- **Permissions and -B** can change cache writing
- **Imported modules and direct scripts** differ
- **Compiled code is not source confidentiality**

| 缓存存在并不意味着新进程跳过模块初始化。



| BigInt · A Later Assignment 大整数预告

Representation

Supported interface

Decimal input string

Sign and grouped digits

Display / compare

str, repr, eq, lt

Arithmetic

add, sub, mul without whole-operand int
conversion

| 当前个人作业2是素数计数，大整数后续启动。

| BigInt · Verification and Report

- **Zero, leading zeros and signs**
- **Unequal lengths, negative differences, carries and borrows**
- **Use int as a test oracle**, not the implementation
- **One-page report**: representation, signs, evidence

| Counting Primes · Contracts and Evidence

Contract

Evidence

$0 \leq p < n; n \leq 100000$

Small values and cross-method counts

Five genuine methods

Inspect source and report, not wrappers

MR bases 2 and 3

Deterministic within the stated bound

Timing after correctness

Feasible sizes, repeated measures, environment



| Lab 10 · Fixed Interfaces 固定接口

Entry	Required boundary
Shapes	Finite positive dimensions; ValueError
Accounts	Integer cents and explicit overdraft rules
Money	Decimal display → rounded cents; sentinel operands
Deck	Fixed cards; invalid deal preserves state
Project	Five real files; silent imports and guarded self-tests