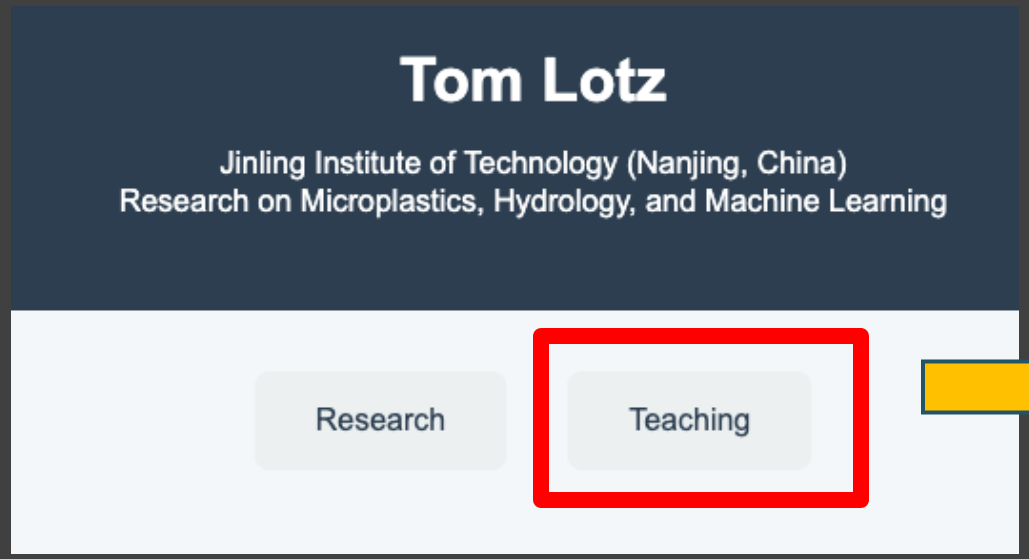
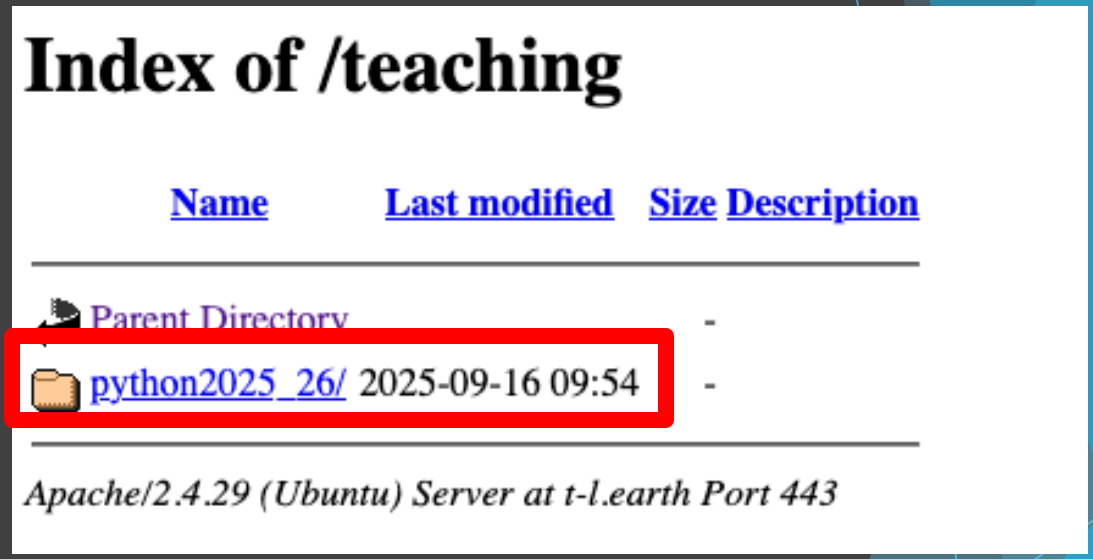
Download the cheat sheets and slides from here

t-l.earth





Python语言程序设计

Python Programming

2025/26

Session 04

Tom Lotz (tom.lotz@outlook.com)

Content

Catch-up

Why this catch-up session?

- Originally, the course concept assumed pre-existing programming skills
- Some students are struggling
- We do a catch-up session to bring everyone up to speed

Catch up Session

- Today we practice step by step, like Duolingo.
- Type along with every example.
- Goal: everyone succeeds!

Catch-up: Block 1

PyCharm

- Start your PyCharm, on your own laptop, or the computer in front of you.
- If none of these work, use www.online-python.com

Print Something

- Explanation: print() shows text on the screen. Text must be written as a string, which means characters inside quotes like "hello" or 'hello'.
- Task: Make the computer say hello.
- Solution:

Print Something

- Explanation: print() shows text on the screen. Text must be written as a string, which means characters inside quotes like "hello" or 'hello'.
- Task: Make the computer say hello.
- Solution:

```
print("hello")
```

Print Two Things

- Explanation: in print() you can separate values with commas.
- Task: Make the computer say hello world.
- Solution:

Print Two Things

- Explanation: in print() you can separate values with commas.
- Task: Make the computer say hello world.
- Solution:

```
print("hello", "world")
```

Your Own Text

- Explanation: Text inside quotes can be anything.
- Task: Make the computer introduce you with your own name.
- Solution:

Your Own Text

- Explanation: Text inside quotes can be anything.
- Task: Make the computer introduce you with your own name.
- Solution:

```
print("My name is Bob")
```

Variables

- Explanation: A variable stores a value. You can use it later again and again in your program. Variables are assigned by using variable = value.
- Task: Save your name in a variable and print it.
- Solution:

Variables

- Explanation: A variable stores a value. You can use it later again and again in your program. Variables are assigned by using variable = value.
- Task: Save your name in a variable and print it.
- Solution:

```
name = "Bob"
print("Hi", name)
```

Numbers in Variables

- Explanation: Variables can also store numbers.
- Task: Save your own age in a variable and print it.
- Solution:

Numbers in Variables

- Explanation: Variables can also store numbers.
- Task: Save your own age in a variable and print it.
- Solution:

```
age = 20
print(age) # version 1
print("I am", age, "years old") # version 2
```

Python as a Calculator

- Explanation: Python can do math.
- Task: Show the result of 3 + 4.
- Solution:

Python as a Calculator

- Explanation: Python can do math.
- Task: Show the result of 3 + 4.
- Solution:

```
print(3 + 4)
```

Math with Variables

- Explanation: Variables can be used in calculations.
- Task: Save two numbers in variables and print their sum.
- Solution:

Math with Variables

- Explanation: Variables can be used in calculations.
- Task: Save two numbers in variables and print their sum.
- Solution:

```
a = 5
b = 7
print(a + b)
```

Reassigning Variables

- Explanation: You can change a variable’s value anytime.
- Task: Start with 3 coins, then add 1 and print the new value.
- Solution:

Reassigning Variables

- Explanation: You can change a variable’s value anytime.
- Task: Start with 3 coins, then add 1 and print the new value.
- Solution:

```
coins = 3
print("Coins:", coins)
coins = coins + 1
print("Coins:", coins)
```

Mini Challenge

- Explanation: Now you can combine what you have learned.
- Task: Write your own introduction with your name and age using variables.
- Solution:

Mini Challenge

- Explanation: Now you can combine what you have learned.
- Task: Write your own introduction with your name and age using variables.
- Solution:

```
name = "Alice"
age = 20
print("Hello, my name is", name, "and I am", age, "years old.")
```

Catch-up: Block 2

Lists Introduction

- Explanation: A list stores multiple values in one variable.

```
fruits = ["apple", "banana", "cherry"]
```

- Task: Make a list called foods with your 3 favorite foods.
- Solution:

Lists Introduction

- Explanation: A list stores multiple values in one variable.

```
fruits = ["apple", "banana", "cherry"]
```

- Task: Make a list called foods with your 3 favorite foods.
- Solution:

```
foods = ["pizza", "sushi", "pasta"]
```

Accessing List Items

- Explanation: We can get items by their position (starting at 0).

```
print(fruits[0])
```

- Task: Print the first item in your foods list.
- Solution:

Accessing List Items

- Explanation: We can get items by their position (starting at 0).

```
print(fruits[0])
```

- Task: Print the first item in your foods list.
- Solution:

```
print(foods[0])
```

Modifying a List

- Explanation: We can easily change items in a list, just like we changed variables.
- Task: Change the second item in your foods list.
- Solution:

Modifying a List

- Explanation: We can easily change items in a list, just like we changed variables.
- Task: Change the second item in your foods list.
- Solution:

```
foods[1] = "burger"
```

Adding Items

- Explanation: Use .append() to add to the end of a list.

```
fruits.append("grape")
```

- Task: Add the food “ice cream” to your list.
- Solution:

Adding Items

- Explanation: Use .append() to add to the end of a list.

```
fruits.append("grape")
```

- Task: Add the food “ice cream” to your list.
- Solution:

```
foods.append("ice cream")
```

Looping Through a List

- Explanation: A loop repeats an action for each item.

```
for fruit in fruits:
    print(fruit)
```

- Task: Print each food in your foods list.
- Solution:

Looping Through a List

- Explanation: A loop repeats an action for each item.

```
for fruit in fruits:
    print(fruit)
```

- Task: Print each food in your foods list.
- Solution:

```
for food in foods:
    print(food)
```

Mini Challenge

- Explanation: Combine lists and loops.
- Task: Make a list of 3 hobbies and print a sentence for each.
- Solution:

Mini Challenge

- Explanation: Combine lists and loops.
- Task: Make a list of 3 hobbies and print a sentence for each.
- Solution:

```
hobbies = ["reading", "cycling", "gaming"]
for hobby in hobbies:
    print("One of my hobbies is", hobby)
```

Catch-up: Block 3

Decisions With if

- Explanation: An if statement lets the computer make a decision.

```
age = 18
if age >= 18:
    print("You are an adult")
```

- Task: Make a program that prints “Positive” if a number is greater than 0.

- Solution:

Decisions With if

- Explanation: An if statement lets the computer make a decision.

```
age = 18
if age >= 18:
    print("You are an adult")
```

- Task: Make a program that prints “Positive” if a number is greater than 0.

- Solution:

```
number = 5
if number > 0:
    print("Positive")
```

if … else …

- Explanation: else runs when the condition is not true.

```
age = 18
if age >= 18:
    print("You are an adult")
else:
    print("You are not an adult")
```

- Task: Make a program that prints “Positive” if a number is greater than 0, or “Negative” otherwise.

- Solution:

if … else …

- Explanation: else runs when the condition is not true.

```
age = 18
if age >= 18:
    print("You are an adult")
else:
    print("You are not an adult")
```

- Task: Make a program that prints “Positive” if a number is greater than 0, or “Negative” otherwise.
- Solution:

```
number = 5
if number > 0:
    print("Positive")
else:
    print("Negative")
```

Multiple Conditions (elif)

- Explanation: elif allows to add more conditions.

```
age = 18
if age >= 18:
    print("You are an adult")
elif age > 1:
    print("You are a child")
else:
    print("You are a baby")
```

- Task: Add one more condition to your number check.
- Solution:

Multiple Conditions (elif)

- Explanation: elif allows to add more conditions.

```
age = 18
if age >= 18:
    print("You are an adult")
elif age > 1:
    print("You are a child")
else:
    print("You are a baby")
```

- Task: Add one more condition to your number check.
- Solution:

```
number = 5
if number > 0:
    print("Positive")
elif number == 0:
    print("What is 0?")
else:
    print("Negative")
```

Getting Input

- Explanation: We can ask the user for input with input().

```
text = input("Type something: ")
print("You entered", text)
```

- Task: Ask the user for their name and greet them.
- Solution:

Getting Input

- Explanation: We can ask the user for input with input().

```
text = input("Type something: ")
print("You entered", text)
```

- Task: Ask the user for their name and greet them.
- Solution:

```
name = input("What is your name? ")
print("Hello", name)
```

Input + Condition

- Explanation: We can combine input() with an if statement.
- Task: Ask the user for their favorite color, and if they answer blue, say that you also like blue.
- Solution:

Input + Condition

- Explanation: We can combine input() with an if statement.
- Task: Ask the user for their favorite color, and if they answer blue, say that you also like blue.
- Solution:

```
color = input("What is your favorite color? ")
if color == "blue":
    print("I like blue!")
```

Mini Challenge

- Explanation: Combine input and if-statements.
- Task: Ask for the user’s name. If the name is "Alice", print "Hello Alice!". Otherwise, print "Hello stranger!".
- Solution:

Mini Challenge

- Explanation: Combine input and if-statements.
- Task: Task: Ask for the user’s name. If the name is "Alice", print "Hello Alice!". Otherwise, print "Hello stranger!".
- Solution:

```
name = input("What is your name? ")
if name == "Alice":
    print("Hello Alice!")
else:
    print("Hello stranger!")
```

Congratulations!

- You have finished the Python Catch-up Session!