

Python语言程序设计

Python Programming

2026/27



Session 02

Tom Lotz (tom.lotz@outlook.com)

Content

Review

01 Strings and Messages

02 Numbers and Practice

03 Text Input

04 Recap

05 Homework

Review

Trace a Changing Value

- Do not run the program yet.
- Write the value of score after every line.
- Predict the final displayed result.

```
score = 8
bonus = 3
score = score + bonus
penalty = 2
score = score - penalty
print(score)
```



Explain the Value Changes

- Why does score appear on both sides of =?
- Which instruction changes score for the first time?
- What would the final value be if bonus became 5?
- Which values remain unchanged throughout the program?
- Run the program only after agreeing on your prediction.



Using a Variable

- A variable stores a value under a name
- Create a variable by assigning a value
- Python can reuse that value or assign a new one
- message is the variable name

```
message = "Hello, Python!"  
print(message)
```



Using a Variable

- You can assign a new value to a variable at any time
- Python will always use the latest value

```
message = "Hello, Python!"  
print(message)  
message = "Hello, Crash Course!"  
print(message)
```



Variable Naming Rules



- Use letters, numbers, and underscores: `greeting_1`
- Must start with a letter or underscore (not a number)
- No spaces allowed
- Cannot use reserved Python words (such as `for`, `if`, `class`, or `def`)
- Use lowercase and descriptive names: `user_name`, not `u`

Strings and Messages

Strings and Messages

- Text can be stored, changed, formatted, and displayed.

Problem 1: Create a Course Introduction

- The stored name is "mina chen".
- The stored course is "python programming".
- The required output is: "Mina Chen is studying Python Programming"
- Do not write Python yet.
- Compare the stored text with the required output. What is different?

What Must the Program Do?

- What information is stored?
- Which words in the required sentence remain fixed?
- What is different about "mina chen" and "Mina Chen"?
- What is different about "python programming" and "Python Programming"?
- What must happen in addition to storing and displaying the text?
- We need a Python tool that can change capitalization.

Strings Store Text

- A string is text surrounded by quotation marks.
- The quotation marks show where the text begins and ends.
- The quotation marks are not displayed as part of the value.

```
student_name = "mina chen"  
course_name = "python programming"
```

Change How Text Is Displayed

- `.title()` capitalizes the first letter of each word.
- `.upper()` displays every letter in uppercase.
- `.lower()` displays every letter in lowercase.
- The original stored string does not change.

```
student_name = "mina chen"  
print(student_name.title())  
print(student_name.upper())  
print(student_name.lower())
```

Combine the Stored Information

- `print()` can receive more than one value.
- `.title()` changes how each stored string is displayed.
- Predict the exact capitalization and spacing before running it.

```
student_name = "mina chen"  
course_name = "python programming"  
print(student_name.title(), "is studying", course_name.title())
```

Change the Stored Information

1. Change the stored name and course, but not the `print()` instruction.
2. Predict the complete output, including capital letters and spaces.
3. Run the program and compare the result with your prediction.
4. Explain why the same instructions still work with different stored values.

Change How One Value Is Displayed

1. Keep the stored values unchanged.
2. Change `.title()` to `.upper()` for the course only.
3. Predict the exact output before running the program.
4. Explain why the stored course did not change.

Debug: Find and Fix String Errors

```
message = 'One of Python's strengths is...'  
# SyntaxError: unterminated string
```

```
message = "One of Python's strengths is..."  
message2 = 'One of Python\'s strengths is...'
```

Numbers and Practice

Numbers and Calculations

- A calculation begins with understanding the available information and required result.

Problem 2: Work Within a Budget

- One event ticket costs 35.50 yuan.
- A student wants 3 tickets and has a budget of 150 yuan.
- The program must display the total cost and the money remaining.
- Decide which result must be calculated first. Do not write Python yet.

Plan the Two Calculations

- Information: ticket price, number of tickets, and budget.
- First result: $\text{total cost} = \text{ticket price} \times \text{number of tickets}$.
- Second result: $\text{money remaining} = \text{budget} - \text{total cost}$.
- Expected results: total cost 106.5; money remaining 43.5.
- Why must this version assign `total_cost` before calculating `money_left`?

Python Number Types

- **Integers** (whole numbers)
 - Examples: -2, 0, 42
- **Floats** (numbers with decimals)
 - Examples: 3.14, 0.0, -2.5
- Python automatically chooses the type

```
print(type(5))      # <class 'int'>  
print(type(2.0))    # <class 'float'>
```

- You don't need to declare types in advance

Code the Planned Calculation

- Trace the value created by each calculation before predicting both output lines.

```
ticket_price = 35.50
number_of_tickets = 3
budget = 150
total_cost = ticket_price * number_of_tickets
money_left = budget - total_cost
print("Total cost:", total_cost)
print("Money left:", money_left)
```

Test a Changed Situation

1. Change the number of tickets to 4.
2. Predict `total_cost` and `money_left` without running the program.
3. Is the budget still sufficient? Explain using the calculated values.
4. Run the program and compare it with your reasoning.
5. Which instructions could remain unchanged for any ticket quantity?

Arrange the Program

- The lines below are not in a useful order.
- Arrange them so the program calculates the money remaining.
- Explain why each line must come before or after another line.
- Only then type and run the program.

Lines to Arrange

- Required output: Money left: 43.5

```
print("Money left:", money_left)
money_left = budget - total_cost
budget = 150
total_cost = ticket_price * number_of_tickets
number_of_tickets = 3
ticket_price = 35.50
```

Choose and Justify

- A program must show the money remaining after buying tickets.
- Which expression is correct?
- A: `total_cost`
- B: `budget - total_cost`
- C: `budget - number_of_tickets`
- Explain what each incorrect expression actually calculates.

User input

Text Input

- Programs can ask users for changing information instead of storing every value in advance.

Problem 3: Personalize the Introduction

- Until now, the student name was written directly in the program.
- Change the program so the user supplies the name.
- The program must display the entered name in title case.
- Decide where asking, storing, transforming, and displaying belong.

input() Waits for Text

- `input()` displays a prompt and waits for the user.
- It returns the text entered by the user.
- The assignment stores that text in `student_name`.

```
student_name = input("What is your name? ")
```

Ask, Transform, and Display

- Predict the complete interaction if the user enters "mina chen".
- Then enter a lowercase name and check the capitalization.

```
student_name = input("What is your name? ")  
print("Welcome to Python,", student_name.title())
```

Change: Ask for the Course Too

1. Ask the user for both a student name and a course name.
2. Store the two answers in different variables.
3. Display both answers in title case.
4. Predict the complete interaction before running the program.
5. Explain the order: ask → store → transform → display.

Recap

What You Learned Today

- Understand the information, output, and process before coding.
- Strings store text inside quotation marks.
- Variables give stored values meaningful names.
- String methods can change how text is displayed.
- Integers and floats can be used in calculations.
- `input()` returns text entered by the user.

Homework

Session 2 Homework

- Complete both programming tasks.
- Predict and plan before running your code.
- You must be able to explain everything you submit.

Task 1: Personal Course Introduction

1. Ask the user for a student name and a course name.
2. Store both answers in clearly named variables.
3. Display both answers in title case.
4. Required pattern: "Mina Chen is studying Python Programming"
5. Test the program with two different names and courses.

Task 2: Travel Budget

1. Store the transport cost per person, number of travelers, and total budget.
2. Calculate `total_travel_cost`.
3. Use `total_travel_cost` to calculate the `remaining_budget`.
4. Display both results with clear labels.
5. Predict the results before running the program.
6. Change the number of travelers and test the program again.

Submit and Explain

- Submit: studentID_session02.py
- Send files by email, not through WeChat.
- Do not submit AI-generated code that you cannot explain.