Talking to Your Program: input()

Python for Absolute Beginners · Lesson 7

What you'll learn

- How input() asks a question and waits for an answer
- How to store what the person types in a variable
- Why input() always gives back text, even when the person types digits
- A friendlier print: showing text and numbers together with commas
- How int() turns typed text into a number you can do arithmetic with

Programs that listen

- Until now, every run was the same — input() lets a program ask, wait, and use the answer
- The text in parentheses is the prompt, the words shown to the person
- input() waits until the person presses Enter, then hands back exactly what they typed, as text
- End the prompt with a space so the typing doesn't bump against the question

Your first input

```
name = input("What is your name? ")
print("Nice to meet you, " + name + "!")

Output:
What is your name? Sam
Nice to meet you, Sam!
```

The text in parentheses is the prompt — end it with a space so the typing doesn't bump against the question.

Type in the right place

- When the program waits, click in the Shell window and type there, then press Enter
- If the program seems frozen, it is waiting for you
- Typing the answer into the editor window is the most common first-day mistake

What you type is always text

- Everything input() returns is text, even digits
- + joins text together, so age + age gave 1212, not 24
- To Python, your 12 is the characters "1" and "2", not the number twelve

Text plus text, in code

```
                age = input("How old are you? ")
print(age)
print(age + age)

Output:
How old are you? 12
12
1212
```

age + age printed 1212, not 24 — + joins text instead of adding.

The TypeError

```
age = input("How old are you? ")
print(age + 1)

Output:
TypeError: can only concatenate str (not "int") to str
```

Read it slowly: concatenate means join, str means text, and int means whole number. It points at the exact problem.

A friendlier print: commas

```
                    print("Hello,", "Sam")
print("Sam is", 12, "years old")

Output:
Hello, Sam
Sam is 12 years old
```

Commas put a space between pieces and mix text and numbers freely. Use commas to show things; use + to build one piece of text to store.

Turning text into a number

```
            age_text = input("How old are you? ")
age = int(age_text)
print("Next year you will be", age + 1)

Output:
How old are you? 12
Next year you will be 13
```

Now age holds the number 12, so age + 1 is 13.

One line, and when it fails

```
age = int(input("How old are you? "))

Output (type twelve):
How old are you? twelve
ValueError: invalid literal for int() with base 10:
'twelve'
```

Read it from the inside out: ask first, then convert. int() can only convert text that looks like a whole number — a decimal like 12.5 fails too.

Putting it together

```
name = input("What is your name? ")
age = int(input("How old are you? "))
print("Hello,", name)
print("In 10 years you will be", age + 10)

Output:
What is your name? Sam
How old are you? 12
Hello, Sam
In 10 years you will be 22
```

Run it again with different answers — the output changes while the program stays exactly the same.

Hands-on exercise

- Ask for your favorite food and print it followed by "is a great choice!"
- Ask for your name and print how many letters it has, using len() — like "Your name has 3 letters"
- Ask for the current year and your birth year, convert both with int(), and print your age in years
- Run a program and type twelve where it wants a number — read the error and explain it in one sentence

Key takeaways

- input("question ") shows a prompt, waits for Enter, and returns what was typed
- Store the answer in a variable, like any other value
- Everything input() returns is text, even digits, so + joins instead of adds
- print() with commas shows text and numbers together, with a space between them
- int() turns digit text into a whole number; int(input(...)) asks and converts in one line
- A TypeError means text and a number were mixed
- A ValueError from int() means the text didn't look like a whole number

Take the quiz

Take the Lesson 7 quiz — 5 quick questions to check what you learned.

Next lesson

Booleans and Comparisons