

Input and Output - Making Interactive Programs

Introduction

Hello, future programmers! Today, we're learning about input and output in Python - the key to making interactive programs. These are the skills that turn your code from something that just runs automatically into something that can have a conversation with users. Let's dive in!

What are Input and Output?

- **Output:** Information that your program shows to the user (like text, numbers, or graphics)
- **Input:** Information that the user gives to your program (like typing answers or clicking buttons)

Think of it like a conversation:

- When you speak to someone, that's output
- When you listen to their response, that's input

Basic Output with `print()`

The `print()` function is the simplest way to display output:

```
print("Hello, world!")
print("My name is Python.")
print("I am a programming language.")
```

You can print multiple items by separating them with commas:

```
name = "Alex"
age = 10
print("Name:", name, "Age:", age)
```

Basic Input with `input()`

The `input()` function lets your program get information from the user:

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

When your program runs this code:

1. It shows the message "What is your name? "
2. It waits for the user to type something and press Enter
3. It stores what the user typed in the variable `name`
4. It then continues to the next line of code

Important: input() Always Returns a String

The `input()` function always gives you a string, even if the user types a number:

```
age_string = input("How old are you? ")
print(type(age_string)) # Will show <class 'str'>
```

If you need a number, you must convert the input:

```
age_string = input("How old are you? ")
age = int(age_string) # Convert to integer
next_year = age + 1
print("Next year, you'll be", next_year)
```

You can also do this in one line:

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

Handling Different Types of Input

For integers (whole numbers):

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

For floats (decimal numbers):

```
height = float(input("How tall are you in meters? "))
```

For strings (text):

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

For yes/no questions:

```
answer = input("Do you like pizza? (yes/no) ")
if answer.lower() == "yes":
    print("Me too!")
else:
    print("That's okay!")
```

Formatting Your Output

You can make your output look nicer in several ways:

Using f-strings (Python 3.6+)

```
name = "Emma"
age = 11
print(f"Hello, my name is {name} and I am {age} years old.")
```

Using multiple print statements

```
print(""" * 30)
print(""" WELCOME TO MY PROGRAM """)
print(""" * 30)
```

Adding spaces and new lines

```
# \n creates a new line
print("Line 1\nLine 2\nLine 3")

# \t adds a tab
print("Name:\tAlex\nAge:\t10")
```

Making Your Program Pause

Sometimes you want your program to wait for the user before continuing:

```
print("Welcome to the Adventure Game!")
input("Press Enter to continue...")
print("You find yourself in a dark forest...")
```

Error Handling

When getting input, things can go wrong. For example, what if the user types "ten" instead of "10"? Let's handle that:

```
try:
    age = int(input("How old are you? "))
    print("Next year, you'll be", age + 1)
except ValueError:
    print("That's not a valid number!")
```

Hands-on Activities

Activity 1: Simple Calculator

```
print("Simple Calculator")
print("-----")

# Get input from the user
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
operation = input("Enter operation (+, -, *, /): ")

# Perform calculation based on the operation
if operation == "+":
    result = num1 + num2
    print(f"{num1} + {num2} = {result}")
elif operation == "-":
    result = num1 - num2
    print(f"{num1} - {num2} = {result}")
elif operation == "*":
```

```

    result = num1 * num2
    print(f"{num1} * {num2} = {result}")
elif operation == "/":
    if num2 != 0: # Check for division by zero
        result = num1 / num2
        print(f"{num1} / {num2} = {result}")
    else:
        print("Error: Cannot divide by zero!")
else:
    print("Error: Invalid operation!")

```

Activity 2: Mad Libs Story Generator

```

print("MAD LIBS STORY GENERATOR")
print("-----")

# Get input from the user
name = input("Enter a name: ")
animal = input("Enter an animal: ")
place = input("Enter a place: ")
adjective = input("Enter an adjective (describing word): ")
verb = input("Enter a verb ending in 'ing': ")

# Create the story
story = f"""
Once upon a time, there was a {adjective} {animal} named {name}.
{name} loved {verb} in {place}.
Everyone in {place} thought {name} was the most {adjective} {animal} ever!
"""

# Display the story
print("\nHere's your story:")
print("-----")
print(story)

```

Activity 3: Quiz Game

```

print("PYTHON QUIZ GAME")
print("-----")
print("Answer these questions to test your Python knowledge!")

score = 0

# Question 1
answer1 = input("What symbol is used for comments in Python? ")
if answer1 == "#":
    print("Correct!")
    score += 1
else:
    print("Incorrect. The answer is #")

# Question 2
answer2 = input("What function do you use to get input from the user? ")
if answer2 == "input" or answer2 == "input()":
    print("Correct!")
    score += 1
else:
    print("Incorrect. The answer is input()")

```

```

# Question 3
answer3 = int(input("How does Python count the first character in a string? (Enter a number) "))
if answer3 == 0:
    print("Correct!")
    score += 1
else:
    print("Incorrect. The answer is 0 (zero-indexed)")

# Final score
print(f"\nYour final score: {score}/3")
if score == 3:
    print("Perfect score! You're a Python wizard!")
elif score >= 1:
    print("Good job! Keep practicing!")
else:
    print("Keep studying! You'll get better!")

```

Wrap-up

Congratulations! You now know how to create interactive programs that can get information from users and display customized output. This opens up a whole new world of possibilities for your programs.

Challenge: Adventure Game

Create a simple text adventure game that:

1. Asks the user for their name and greets them
2. Presents them with a scenario (e.g., "You're in a cave with two tunnels")
3. Gives them choices (e.g., "Do you go left or right?")
4. Has different outcomes based on their choice
5. Includes at least one numerical input (e.g., "How many steps do you take?")

Remember to use appropriate input validation and make your game fun with descriptive text!