

## Lesson 2: Input/Output and Conditional Logic

### Objectives:

- Learn to use input and output functions in Python.
- Understand and apply conditional statements to make decisions in code related to space science.

### Lesson Plan

#### 1. Introduction (10 minutes)

##### Engagement:

- Begin with a discussion on how data from space missions is gathered (e.g., through telescopes, satellites, rovers) and how scientists use that data to make informed decisions.

##### For example:

- Mars Rovers (Curiosity, Perseverance)
  - Description: Rovers are robotic explorers sent to planets like Mars to gather data directly from the surface.
  - Data Collected: High-resolution images, soil and rock samples, atmospheric readings, and chemical analysis.
  - Usage: Helps scientists understand the geology and climate of Mars, search for signs of past life, and prepare for future human missions.
- Discussion Point:
  - Explain the mechanics of how rovers operate and navigate on other planets.
  - Emphasize the significance of rovers in uncovering evidence of water on Mars and other key discoveries.

- Show a simple example of a Python program that asks for user input and provides an output.

### Example code

```
# This program takes user input for the name of a planet and its distance from the
Sun, then prints the information back to the user.

# Prompt the user to enter the name of a planet
planet_name = input("Enter the name of a planet: ")

# Prompt the user to enter the distance of the planet from the Sun in millions of
kilometers
distance_from_sun_km = input("Enter the distance of the planet from the Sun (in
millions of kilometers): ")

# Display the collected information in a readable format
print("The distance from the Sun to", planet_name, "is", distance_from_sun_km,
"million kilometers.")
```

## 2. Explanation (15 minutes)

### Input and Output Functions:

- Introduce the `input()` function to get user input.
  - The `input()` function is used to get input from the user. When you use `input()` in your code, it prompts the user to type something. The value entered by the user is then stored in a variable for further use in the program.
- Explain the `print()` function to display output.
  - It just is the output of your code.
  - HELPFUL TIP: When writing your code, it is often useful to include multiple print statements to check if your code is working as expected. These print statements can help you identify where an error might be occurring. You can comment out or delete these print statements once your code is complete.

#### Example code:

```
# Prompt the user to enter the name of a planet
planet_name = input("Enter the name of a planet: ")

# The input() function displays the message within the parentheses and waits for the
user to type a response.
# Once the user presses Enter, the response is stored in the variable planet_name.

# Display the collected information
print("You entered:", planet_name)
```

- You can even ask for multiple inputs.

#### Example code:

```
# Prompt the user to enter the name of a planet
planet_name = input("Enter the name of a planet: ")

# Prompt the user to enter the distance of the planet from the Sun in millions of
kilometers
distance_from_sun_km = input("Enter the distance of the planet from the Sun (in
millions of kilometers): ")

# Prompt the user to enter the number of moons the planet has
number_of_moons = input("Enter the number of moons: ")

# Display the collected information in a readable format
print(f"The planet {planet_name} is {distance_from_sun_km} million kilometers away
from the Sun and has {number_of_moons} moons.")
```

### Conditional Statements:

- Conditional statements are a fundamental concept in programming that allow you to control the flow of your code based on certain conditions. They enable your program to make decisions and execute specific blocks of code depending on whether a condition is true or false. In Python, the primary conditional statements are `if`, `elif`, and `else`.

#### 1. `if` Statement

The `if` statement evaluates a condition (an expression that returns a Boolean value `True` or `False`). If the condition is `True`, the block of code indented under the `if` statement is executed. If the condition is `False`, the code block is skipped.

Syntax:

```
if condition:
    # Block of code to be executed if the condition is true
```

Example Code:

```
distance_from_sun_km = int(input("Enter the distance of the planet from the Sun (in
millions of kilometers): "))

if distance_from_sun_km < 150:
    print("The planet is relatively close to the Sun.")
```

Explanation:

- The condition `distance_from_sun_km < 150` is checked.
- If the condition is true, the message "The planet is relatively close to the Sun." is printed.

## 2. elif Statement

The `elif` (short for else if) statement allows you to check multiple conditions after the initial `if` statement. If the initial `if` condition is `False`, the program checks the `elif` condition. If the `elif` condition is `True`, the corresponding block of code is executed.

Syntax:

```
if condition1:
    # Block of code to be executed if condition1 is true
elif condition2:
    # Block of code to be executed if condition1 is false and condition2 is true
```

Example:

```
distance_from_sun_km = float(input("Enter the distance of the planet from the Sun (in
millions of kilometers): "))

if distance_from_sun_km < 150:
    print("The planet is relatively close to the Sun.")
elif distance_from_sun_km > 450:
    print("The planet is far from the Sun.")
```

Explanation:

- If `distance_from_sun_km < 150` is false, the program checks `distance_from_sun_km > 450`.
- If `distance_from_sun_km > 450` is true, the message "The planet is far from the Sun." is printed.

## 3. else Statement

The `else` statement is used to define a block of code to be executed if none of the preceding conditions are true. It is optional but provides a catch-all block that executes when all other conditions fail.

Syntax:

```
if condition1:
    # Block of code to be executed if condition1 is true
elif condition2:
    # Block of code to be executed if condition1 is false and condition2 is true
else:
    # Block of code to be executed if none of the above conditions are true
```

Example:

```
distance_from_sun_km = float(input("Enter the distance of the planet from the Sun (in
millions of kilometers): "))

if distance_from_sun_km < 150:
    print("The planet is relatively close to the Sun.")
elif distance_from_sun_km > 450:
```

```
        print("The planet is far from the Sun.")
    else:
        print("The planet is at a moderate distance from the Sun.")
```

Explanation:

- If both `distance_from_sun_km < 150` and `distance_from_sun_km > 450` are false, the message "The planet is at a moderate distance from the Sun." is printed.

## Importance of Conditional Statements in Decision-Making

### Control Flow:

Conditional statements control the flow of a program by allowing certain parts of the code to be executed only when specific conditions are met. This helps in creating dynamic and responsive programs that can handle a variety of inputs and situations.

### Decision Making:

They enable the program to make decisions. For example, determining if a user is eligible for a service based on age, calculating discounts based on purchase amount, or deciding the outcome of a game based on player actions.

### Error Handling:

Conditional statements can be used to check for errors or invalid input and handle them appropriately. For example, checking if a user has entered a valid email address before proceeding with registration.

### Efficiency:

By executing only the necessary code blocks, conditional statements can make programs more efficient. They help avoid unnecessary computations and operations.

### Flexibility:

They provide flexibility in programming. You can add as many conditions as needed using multiple `elif` statements and handle various scenarios in a structured manner.

## 3. Hands-On Activity (20 minutes)

### Task:

- Students will create a Python program that asks for input about a planet's characteristics and uses conditional statements to provide feedback.

### Worksheet (below):

- **Practice with using `input()` and `print()` functions.**
- **Exercises on writing conditional statements to classify celestial objects.**

## Example Worksheet Tasks:

```
# Define variables for user input
planet_name = input("Enter the name of a planet: ")
distance_from_sun_km = float(input("Enter the distance of the planet from the Sun (in
millions of kilometers): "))
number_of_moons = int(input("Enter the number of moons: "))

# Conditional statements to classify the planet
if distance_from_sun_km < 150:
    proximity = "close"
elif distance_from_sun_km > 450:
    proximity = "far"
else:
    proximity = "moderate"

print(f"The planet {planet_name} is {proximity} to the Sun and has {number_of_moons}
moons.")
```

## 4. Review (10 minutes)

### Q&A:

- Address any questions students might have about variables, data types, or the example code.

### Exit Ticket:

1. What function do you use in Python to get input from the user?

- A. `print()`
- B. `input()`
- C. `def()`
- D. `for()`

Answer: B

2. What is a conditional statement in Python?

- A. A loop that repeats actions
- B. A storage location for data
- C. A statement that performs different actions based on whether a condition is true or false
- D. A function that outputs data

Answer: C

3. Which of the following Python programs classifies a planet based on the number of moons it has?

A. 

```
number_of_moons = int(input("Enter the number of moons: "))
```

```
if number_of_moons == 0:
    print("The planet has no moons.")
elif number_of_moons <= 2:
    print("The planet has a few moons.")
elif number_of_moons <= 10:
    print("The planet has several moons.")
else:
    print("The planet has many moons.")
```

B. 

```
number_of_moons = int(input("Enter the number of moons: "))
```

```
if number_of_moons == 0:
    print("The planet has many moons.")
elif number_of_moons <= 2:
    print("The planet has several moons.")
elif number_of_moons <= 10:
    print("The planet has a few moons.")
else:
    print("The planet has no moons.")
```

C. 

```
number_of_moons = int(input("Enter the number of moons: "))
```

```
if number_of_moons == 0:
    print("The planet has no moons.")
elif number_of_moons <= 10:
    print("The planet has many moons.")
elif number_of_moons <= 2:
```

```
        print("The planet has a few moons.")
else:
    print("The planet has several moons.")
```

D. `number_of_moons = int(input("Enter the number of moons: "))`

```
if number_of_moons == 0:
    print("The planet has several moons.")
elif number_of_moons <= 2:
    print("The planet has a few moons.")
elif number_of_moons <= 10:
    print("The planet has many moons.")
else:
    print("The planet has no moons.")
```

**Answer: A**



# Worksheet 2: Input/Output and Conditional Logic

## Section 1: Using Input and Output Functions

*Task: Write a program that asks for the name of a planet and its distance from the Sun, then prints this information.*

## Section 2: Conditional Statements

*Task 1: Write a program that classifies a planet as "close," "moderate," or "far" from the Sun based on its distance.*

*Task 2: Write a program that asks for the number of moons a planet has and classifies it as having "few," "several," or "many" moons.*

## ANSWER KEY:

### Worksheet 2: Input/Output and Conditional Logic

#### Section 1: Using Input and Output Functions

*Task: Write a program that asks for the name of a planet and its distance from the Sun, then prints this information.*

```
planet_name = input("Enter the name of a planet: ")
distance_from_sun_km = input("Enter the distance of the planet from the Sun (in
millions of kilometers): ")
print("The distance from the Sun to", planet_name, "is", distance_from_sun_km,
"million kilometers.")
```

#### Section 2: Conditional Statements

*Task 1: Write a program that classifies a planet as "close," "moderate," or "far" from the Sun based on its distance.*

```
distance_from_sun_km = float(input("Enter the distance of the planet from the Sun (in
millions of kilometers): "))

if distance_from_sun_km < 150:
    print("The planet is relatively close to the Sun.")
elif distance_from_sun_km > 450:
    print("The planet is far from the Sun.")
else:
    print("The planet is at a moderate distance from the Sun.")
```

*Task 2: Write a program that asks for the number of moons a planet has and classifies it as having "few," "several," or "many" moons.*

```
number_of_moons = int(input("Enter the number of moons: "))

if number_of_moons == 0:
    print("The planet has no moons.")
elif number_of_moons <= 2:
    print("The planet has a few moons.")
elif number_of_moons <= 10:
    print("The planet has several moons.")
else:
    print("The planet has many moons.")
```