SDEV 1001

Programming Fundamentals

Making Decisions in Programming - 1 and 2

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS

Expectations - What I expect from you

- No Late Assignments

- No Cheating

- Be a good classmate

- Don’t waste your time

- Show up to class

Agenda

On the right is what we will cover today.

What is Making Decisions in Programming?

If Statement Fundamentals in Python

Why Use If Statements?

Basic If Statement Syntax

Comparison Operators

Logical Operators

If-Else Statements

Elif: Multiple Conditions

Indentation Matters!

Combining Conditions

Summary

What is Making Decisions in Programming?

Making decisions in programming refers to the ability to control the flow of a program based on certain conditions. This is typically done using `if`, `elif`, and `else` statements.
Analogy: This is like a traffic light that decides whether to stop or go based on the color of the light.

If Statement Fundamentals in Python

If statements are the building blocks of decision-making in programming.
They allow us to make decisions based on the state of our program.

Why Use If Statements?

Here’s some good reasons to use if statements:

- Control the flow of your program
- Make decisions based on user input or data
- Used in games, websites, data validation, and basically everything you’ll do in programming

Note: If statements are in every programming language and are a part of every project you will ever work on.

Basic If Statement Syntax

Here’s the basic syntax for an if statement in Python:

```python
if condition:
    # code to run if condition is True
```

Example:

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

Comparison Operators

Operators used to compare values in conditions of any data type. Note that you want to compare values of the same type (apples to apples, not apples to oranges).

- `==` : Equal to
- `!=` : Not equal to
- `>` : Greater than
- `<` : Less than
- `>=` : Greater than or equal to
- `<=` : Less than or equal to

Example:

```python
if score == 100:
    print("Perfect score!")
```

Logical Operators

Logical operators are used to combine multiple conditions in an if statement.

- `and` : Both conditions must be True
- `or` : At least one condition must be True
- `not` : Inverts the condition

Example:

```python
if age >= 18 and has_license:
    print("You can drive.")
```

If-Else Statements

Else statements allow you to define an alternative action if the condition is False.

```
if condition:
    # code if condition is True
else:
    # code if condition is False
```

Example:

```
if password == "python123":
    print("Access granted")
else:
    print("Access denied")
```

Elif: Multiple Conditions

Elif (short for "else if") allows you to check multiple conditions in a single if statement. Note this will only check the next condition if the previous one was False.

```
if condition1:
    # code if condition1 is True
elif condition2:
    # code if condition2 is True
else:
    # code if none are True
```

Example:

```
if color == "red":
    print("Stop")
elif color == "yellow":
    print("Caution")
else:
    print("Go")
```

Indentation Matters!

Python uses indentation to define code blocks.

Incorrect indentation will cause an `IndentationError`.

```
if x > 0:
    print("Positive")
     print("This line has too many spaces")  # Error!
```

Combining Conditions

You can combine multiple conditions using `and`, `or`, and `not`.

```python
if temperature > 20 and is_sunny:
    print("Great day for a walk!")
```

Summary

- Use `if`, `elif`, and `else` to control program flow
- Use comparison and logical operators to build conditions
- Indentation is critical in Python
- Practice writing your own decision-making code!

Example

Let’s go run a few examples together

A LEADING POLYTECHNIC COMMITTED TO YOUR SUCCESS