Conditional Statements in Python

Distributed Network and Systems Laboratory,
Chonnam National University
Presenter : Shivani Kolekar

Recap

- ✓ **Computer memory**
- ✓ **Error and its types**
- ✓ **Understanding Variables and Data Types**
- ✓ **Python Keywords**
- ✓ **Basic Arithmetic Operations**

Today's goals

- ✓ **Decision Structures**
- ✓ **Boolean Expressions and Relational Operators**
- ✓ **If-else statements**
- ✓ **Activity**
- ✓ **Q & A**

Control Structure

- ✓ A control structure is a logical design that controls the order in which a set of statements execute.

- So far, we have used only the simplest type of control structure:
  - ✓ the sequence structure. A sequence structure is a set of statements th at execute in the order in which they appear.

Sequence structure example

```
1  # This program displays a person's
2  # name and address.
3  print('Kate Austen')
4  print('123 Full Circle Drive')
5  print('Asheville, NC 28899')
```

Program Output

```
Kate Austen
123 Full Circle Drive
Asheville, NC 28899
```

Decision Structure

- ✓ In this case, we are determining whether the condition Cold outside is true or false.
- ✓ If this condition is true, the action ‘Wear a coat’ is performed.
- ✓ If the condition is false, the action is skipped.

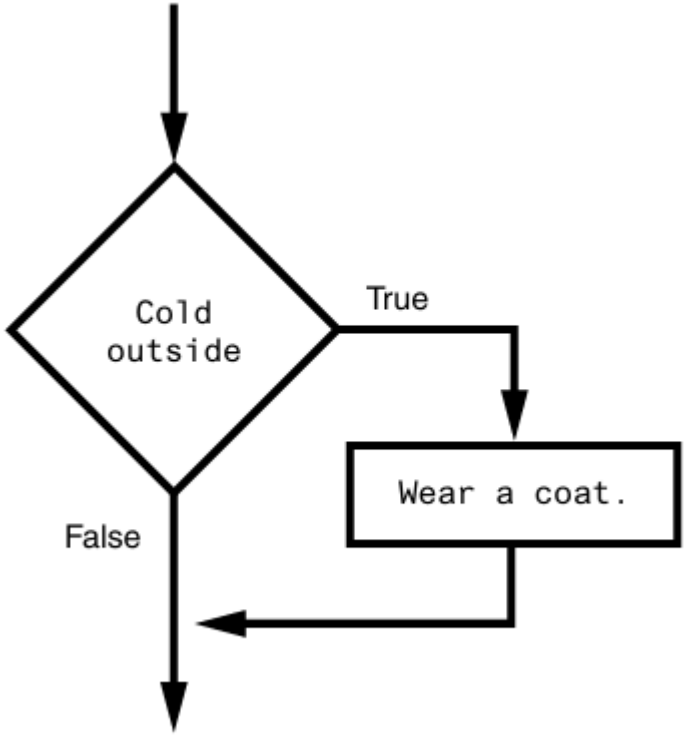


Fig: In the flowchart, the diamond symbol indicates some condition that must be tested.

if statements

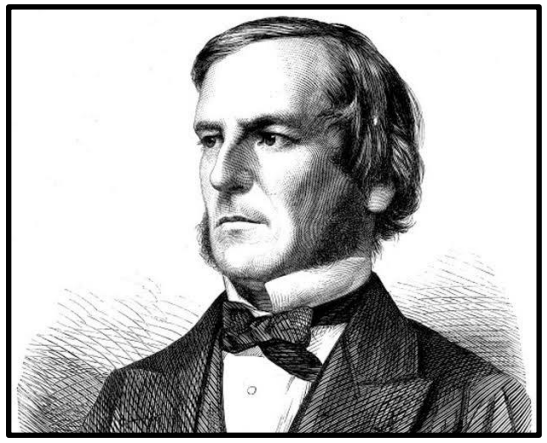
- ✓ The if statement is used to create a decision structure, which allows a program to have more than one path of execution.

- ✓ The if statement causes one or more statements to execute only when a Boolean expression is true.

Booleans

- So far we've been considering straight line code, meaning executing one statement after another.
  a.k.a. sequential flow of control.

- But often in programming,
  you need to ask a question, and do different things based on the answer.
- Boolean values are a useful way to refer to the answer to a yes/no question.



George Boole (1815-1864)

Using Booleans: Example

- ✓ The Boolean literal values are : True, False.
- ✓ Python uses 0 to represent False and anything not 0 to represent True.

```
In [21]: x=1  #statement

In [22]: x<0  #write a boolean expression

Out[22]: False
```

This is a boolean expression that evaluates to either True or False. In this case, it's checking if x is less than 0. Since x is 1, the expression x<0 evaluates to False.

Boolean Expressions and Relational Operators

✓ the if statement : tests an expression to determine whether it is true or false.

- The expressions that are tested by the if statement are called Boolean expressions.

Relational Operators:

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

Using Booleans: Example

- ✓ The Boolean literal values are : True, False.
- ✓ Python uses 0 to represent False and anything not 0 to represent True.

```
In [21]: x=1  #statement

In [22]: x<0  #write a boolean expression

Out[22]: False

In [23]: b = (x==0) #statement containing boolean expression
         print(b)

         False
```

This block first evaluates the boolean expression (x==0), which checks whether x is equal to 0.
It then assigns the result of that expression to the variable b.

Since x is 1, the expression x==0 evaluates to False.

Try this!

✓ You can use the Python interpreter in interactive mode to experiment with these operators.

Relational Operators:

Operator	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

Try this!

```
1  >>> x = 1 Enter
2  >>> y = 0 Enter
3  >>> x > y Enter
4  True
5  >>> y > x
6  False
7  >>>
```

(python interactive mode)

Difference between = and ==

❑ In programming, = and == are used for different purposes:

✓ = is the assignment operator.

- It is used to assign a value to a variable. For example, x = 2 means that the variable x is being assigned the value of 1.

✓ == is the equality comparison operator.

- It is used to evaluate whether two values are equal.
- If the values are equal, the expression returns True; otherwise, it returns False.

Activity!

- Lets practice the relational operators in google colab or using interactive mode in pycharm.

1) define x,y,z variables with different values of your choice and use all relational operations (**>, <, >=, <=, ==, !=**).

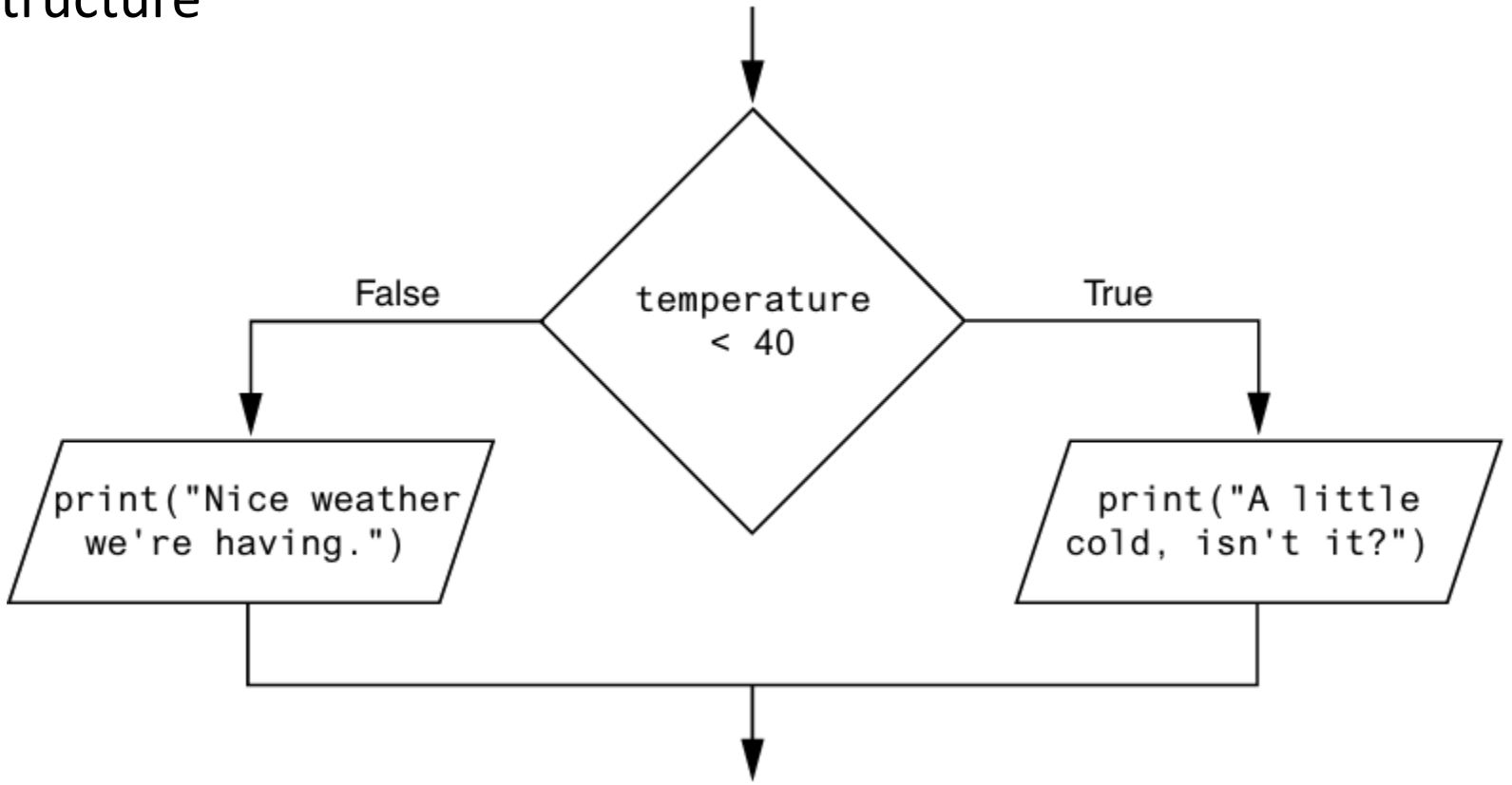
2) Share the screenshot.

If-else statements

- ✓ An if-else statement will execute one block of statements if its condition is true, or another block if its condition is false.
- ✓ A 'dual alternative' decision structure

General format :

```
if condition:
     statement
     statement
     etc.
else:
     statement
     statement
     etc.
```



Conditional execution in an if-else statement

What do you think will be the output of the following codes ?

example 1

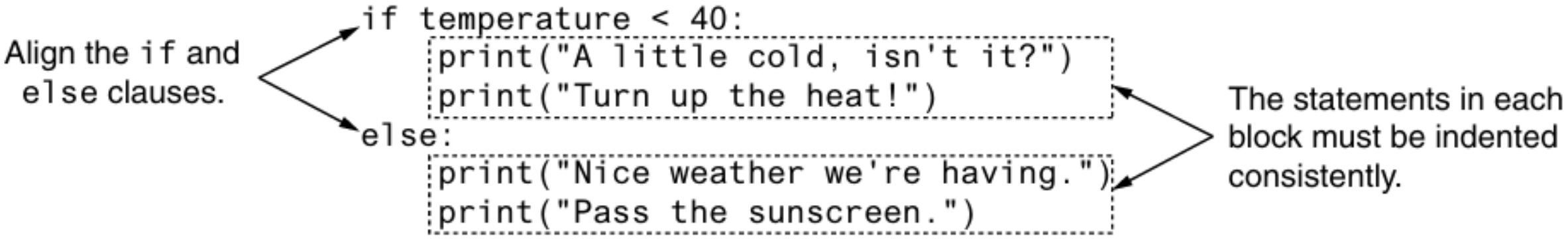
```
>>> temperature = 34
... if temperature<40:
...     print("A little cold, isn't it?")
... else:
...     print("Nice weather we're having.")
```

example 2

```
>>> temperature = 43
... if temperature<40:
...     print("A little cold, isn't it?")
... else:
...     print("Nice weather we're having.")
```

Indentation rules in the if-else statement

- Make sure the ‘if clause’ and the ‘else clause’ are aligned.
- The if clause and the else clause are each followed by a block of statements.
- Make sure the statements in the blocks are consistently indented.



Activity!

1) Follow the if-else statement exercise in google colab.

2) Share the screenshot.

Revision

- ✓Decision Structures
- ✓Boolean Expressions and Relational Operators
- ✓If-else statements

Thank You