PYTHON FOR COMPUTATIONAL PROBLEM SOLVING

Control Structures

Prof. Sindhu R Pai
PCPS Theory Anchor - 2024
Department of Computer Science and Engineering

- Control flow is the order that instructions are executed in a program.
- A control statement is a statement that determines control flow of a set of instructions.

Three fundamental forms of control that programming languages provides,

Sequential Control

Selection Control

Iterative Control

PES UNIVERSITY

Sequential Control

Sequential statements are a set of statements whose execution process happens in a sequence.

Statement 1
Statement 2
Statement 3
....

Indentation in Python

- One fairly unique aspect of Python is that the amount of indentation of each program line is significant.

- In most programming languages, indentation has no affect on program logic—it is simply used to align program lines to aid readability.

- In Python, however, indentation is used to associate and group statements.

PES UNIVERSITY

Selection Control

- The selection statements are also known as Decision control statements or branching statements.
- The selection statement allows a program to test several conditions and execute instructions based on the Truth value of the condition.
- The set of statements following a header in Python is called a **suite** (commonly called a **block**).
- The statements of a given suite **must** all be indented the same amount.
- A header/leader and it' s associated suite are together referred to as a **clause**.
- Below are some of the decision control statements.

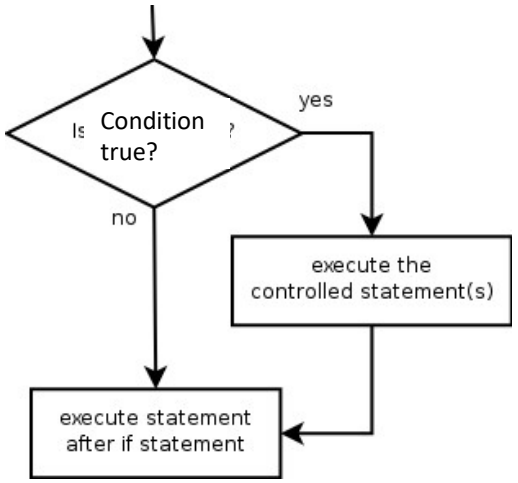
  If, if-else,if-elif-else

PES UNIVERSITY

Selection Control - simple if

- Control structure runs a particular block of code only if certain condition is met.
- Expression can be any expression that evaluates to a Truth value
- Truth value means True of False
- Non zero number, non empty string, non empty list, True, etc are evaluated as True
- Number zero, None, False , empty string, empty list, etc are evaluated as False.

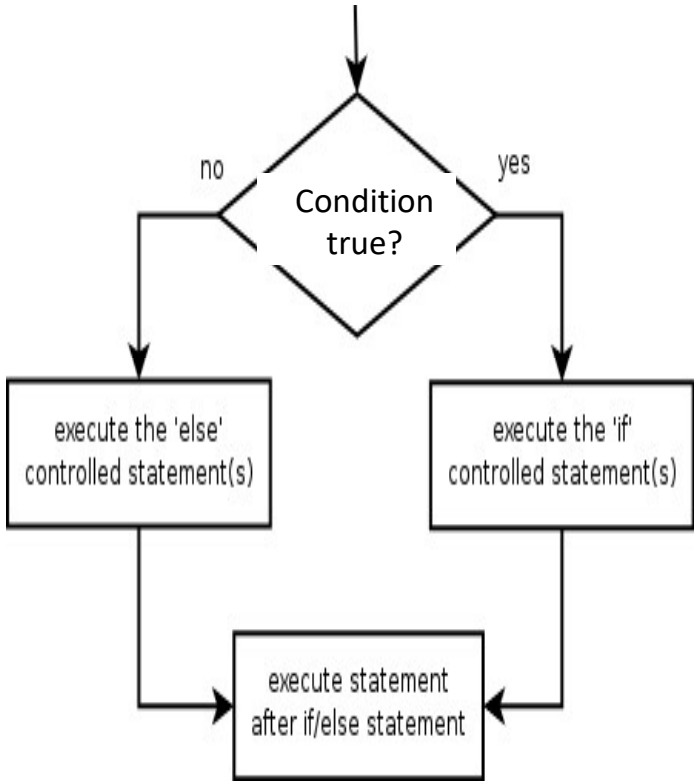
```
if <expression>:
        <block of code>
<statements after if statement>
```



PES UNIVERSITY

Selection Control – if else

A control structure that executes one block of statements if a certain condition is True, and a second block of statements if condition is False.




```
if expression:
        < block of code>
else:
        <block of code>
< statements after the control structure>
```

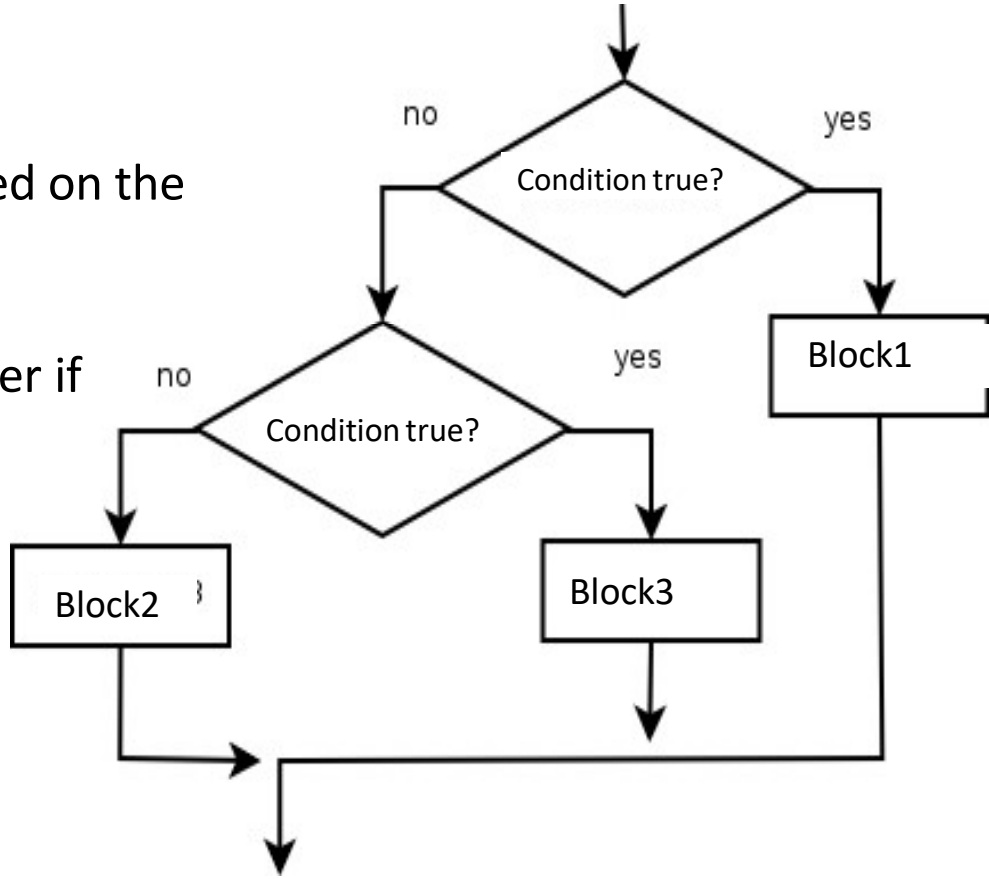
PES UNIVERSITY

Selection Control – if elif else

- Any one block of the code will be executed based on the Truth value of the expression.
- Any number of elif constructs can be written after if statement and else construct is optional.

```
if <expression>:
        < block of code>
elif < expression>:
        < block of code>
else:
        < block of code>

<statements after the control structure>
```



Valid and invalid indentation

A **compound statement** in Python may consist of one or more clauses. While four spaces is commonly used for each level of indentation, any number of spaces may be used, as shown below.

Valid indentation		Invalid indentation	
(a) ```if condition: statement statement else: statement statement```	(b) ```if condition: statement statement else: statement statement```	(c) ```if condition: statement statement else: statement statement```	(d) ```if condition: statement statement else: statement statement```

Iterative Control

- A control statement providing the repeated execution of a set of instructions.
- A set of instructions and the iterative control statement(s) controlling their execution.
- Because of their repeated execution, iterative control structures are commonly referred to as “loops”.
- **for** and **while**

PES UNIVERSITY

Iterative control- while loop

- Repeatedly executes a set of statements based on the provided expression (condition).
- As long as the condition of a while statement is true, the statements within the loop are (re)executed.
- Once the condition becomes false, the iteration terminates and control continues with the first statement after the while loop.
- The condition may be false the first time a loop is reached and therefore the loop would never be executed.
- There are 3 ingredients to any iterative statement: Initial value of the iterative counter, Iterative condition and Updating the iterative counter.

PES UNIVERSITY

Iterative control- while loop

Syntax:

```
while <condition>:
        <suite>
```

Example:

```
i=0
while i<5:#header or leader
        print("Hello ")
        i+=1
```

PES UNIVERSITY

Iterative control- while loop

Infinite Loops

- An **infinite loop** is an iterative control structure that never terminates (or eventually terminates with a system error).
- Infinite loops are generally the result of programming errors.
- For example, if the condition of a while loop can never be false, an infinite loop will result when executed.

range()

- range is a built-in function that returns an iterable object that can be used in a for loop
- range is lazy in 3.x and above
- range **conceptually generates an arithmetic progression**. The values given by range in a for loop can by used as indices for any sequence type.

PES UNIVERSITY

range()

range(start , stop , step)

- It returns a range object that produces a sequence of integers from start(inclusive) to stop(exclusive) by step.
  For example, range(0,4) produces 0,1,2,3
- range(i, j) produces i, i+1, i+2, ..., j-1.
- When step is given, it specifies the increment (or decrement).
- It is a Lazy function - The values come into existence only when we explicitly ask for it.

PES UNIVERSITY

range()

```
>>> range(3)#lazy function
range(0, 3)
>>> list(range(3)) # start defaults to 0, and stop is omitted!
[0, 1, 2]
>>> list(range(3,6))
[3, 4, 5]
>>> list(range(3,3)) #does not give any element
[]
>>> list(range(-6,7,4))
[-6, -2, 2, 6]
>>> list(range(-16,-32,-4))
[-16, -20, -24, -28]
```

PES UNIVERSITY

Iterative control- for loop:

- Python has a statement that works exclusively on collections.
- examines each element and performs any action set by the programmer until there are no more elements left in the collection.
- for statement is a looping structure
- The number of times we execute the body or the suite is normally determinable

Semantics of for :

1. Start the iteration(walking through) of the iterable.
2. Get the element to the variable.
3. Execute the suite or the body.
4. Repeat steps 2 and 3 until the iterable signals that it has no more elements.
5. Move to the next statement and exit the for loop.

PES UNIVERSITY

Iterative control-for loop

Syntax:

```
for <target_variable> in <iterable>:
        <suite/body of for>
```

- An iterable is a collection of elements physically or conceptually.
- It has a built-in mechanism to give an element each time we ask
- It has a built-in mechanism to signal when there are no more elements.

Example:

```
for i in range(5): #header or leader
        print(i)
```

Demo of Control Structures

THANK YOU

Department of Computer Science and Engineering

Dr. Shylaja S S, Director, CCBD & CDSAML, PESU
Prof. Sindhu R Pai – sindhurpai@pes.edu
Prof. Chitra G M
Prof. Gayatri S