

# **III B.Tech.**

## **Computer Science & Engineering**

**CSE304: PYTHON PROGRAMMING WITH  
WEB FRAMEWORKS**

**UNIT – I: String, Range Function, and control  
statements**

**By**  
**Mrs. S. KAMAKSHI, AP-III / CSE**  
**School of Computing**

# Strings

- Immutable
- Delimited by either single or double quote
- Multiline strings embedded with in 3 consecutive single / double quotes
- All relational operators can be used to compare strings
- To convert between upper and lower case:
  - `string.upper()`, `string.lower()`

# range function

- `range(stop)`
  - Returns integers from 0 to stop-1
- `range(start, stop, [, step])`
  - Returns integers from start to stop-step at the intervals of step
- Example:
- `range(5)`     # 0, 1, 2, 3, 4
- `range(1, 6)` # 1, 2, 3, 4, 5
- `range(2, 10, 2)` #2, 4, 6, 8
- `range(5, 0, -1)` # 5, 4, 3, 2, 1

# if ... else statement

```
if bool_exp:
    statements ...
[elif bool_exp:
    statements ... ]...
[else:
    statements ...]
```

# while statement

```
while (bool_expression):  
    statements ...  
else:  
    statements ...
```

# for statement

```
for int_var in range_function:  
    statements ...  
else:  
    statements ...
```

# break and continue



- break – To come out of a loop containing break
- continue – To skip the statements that are placed after this and continue to the next iteration of the loop

# Functions

## Defining Function

```
def fname(arguments):  
    statements...  
    return 1 or more values
```

## Calling the function

```
varlist = fname(actual arguments)
```

- **Argument Matching**
  - Positional argument
  - Named argument

## Example

```
def f1(a, b, c):  
    return a+b+c
```

```
f1(10, 20, 30)
```

```
f1(b=2, a=1, c=3)
```

```
f1(1, c=3, b=2)
```



# main() function

- Defining main() function

```
def main():  
    statements ...
```

```
def main():  
    print("I am main()")
```

- Calling main() function

- Simple way is to place main() outside all functions in a module
- If module is reused (imported) into another module then main() function should not be called. In that case it may be placed as:

```
if __name__ == "__main__":  
    main()
```

- Or simply place the code to be executed within this if statement itself

```
if __name__ == "__main__":  
    print("I am main()")
```

# import statement

- `import module_name [as namespace]`
- `from module_name import function1 [, function2 ...]`
- `from module_name import *`
- Few standard modules:
  - `math` for mathematical operations
  - `random` for generating random nos.
  - `decimal` for working with decimal nos.
  - `csv` for working with csv files
  - `pickle` for persistent data storage
  - `tkinter` for building GUI applications

# random module



- `random()`
  - Returns random value between 0.0 and less than 1.0
- `randint(min, max)`
  - Returns a random int value between min and max-1
- `randrange([start,] stop [,step])` returns a random value  $\geq$  start, less than stop and a multiple of step