PES UNIVERSITY

CELEBRATING 50 YEARS

Computing Fundamentals using Python

SUBJECT CODE : UQ25CA151A

Samyukta D Kumta

Computer Applications

Formatted output in Python

- Formatted output in Python means displaying data in a well-structured, readable, and specific format instead of just printing raw values.
- It allows you to control alignment, width, decimal precision, padding, signs, commas, etc. when printing.

Output without formatting

```
name = “Priya"
age = 25
pi = 3.14159
print("Name:", name, "Age:", age, "Pi:", pi)
```

Output: Name: Priya Age: 25 Pi: 3.14159 #pi has too many decimals

Output with formatting

```
name = “Priya"
age = 25
pi = 3.14159
print(f"Name: {name}, Age: {age}, Pi: {pi:.2f}")
```

Output:

```
Name: Priya, Age: 25, Pi: 3.14
```

Ways to Do Formatted Output in Python

1.f-strings (recommended, Python 3.6+)

Example:

```
>>>pi = 3.14159
>>>print(f"{pi:.3f}")
```

Output:

```
3.142
```

Ways to Do Formatted Output in Python

2. format() method

Example:

```
name = “Priya"
age = 25
pi = 3.14159

print("Name: {}, Age: {}, Pi: {:.2f}".format(name, age, pi))
```

Output:

```
Name: Priya, Age: 25, Pi: 3.14
```

Ways to Do Formatted Output in Python

3. Old % formatting (C-style)

Example:

```
name = “Priya"
age = 25
pi = 3.14159

print("Name: %s, Age: %d, Pi: %.2f" % (name, age, pi))
```

Output:

```
Name: Priya, Age: 25, Pi: 3.14
```

Computing Fundamentals using Python

Ways to align Output in Python

1. print(f"{'Apple':<10} | {'Banana':>10} | {'Mango':^10}")

THANK YOU

Samyukta D Kumta
Department of Computer Applications
samyuktad@pes.edu