

4. PYTHON – NAMING CONVENTIONS

Table of Contents

1. Identifier.....	2
2. Why should we follow naming conventions?	2
3. Points to follow for identifiers in Python	3
# Point 1	3
# Point 2	4
# Point 3	5
# Point 4	6
# Point 5	6
Examples.....	7
Common error.....	7
4. Python identifiers table.....	8
5. Comments in program.....	10
Purpose of comments.....	10
1. Single line comments.....	10
2. Multi line comments.....	11

4. PYTHON - NAMING CONVENTIONS

1. Identifier

- ✓ A name in a python program is called **identifier**.
- ✓ This name can be,

- Package name
- Module name
- Variable name
- Function name
- Class name
- Method name

- ✓ Python creator made some suggestions to the programmers regarding how to write identifiers in a program.

2. Why should we follow naming conventions?

- ✓ While writing the program if we follow the naming conventions then the written code is,
 - Easy to understand.
 - Easy to read.
 - Easy to debug.

3. Points to follow for identifiers in Python

- ✓ We need to follow few points to define an identifiers,

Point 1

- ✓ While writing an identifier we can use,
 - Alphabets, either upper case or lower case
 - Numbers from 0 to 9
 - Underscore symbol (_)
- ✓ If we are using any other symbol then we will get syntax error.

Program Name Creating a valid identifier
demo1.py

```
student_id = 101  
print(student_id)
```

Output
101

Program Name Creating an invalid identifier
demo2.py

```
$student_id = 101  
print($student_id)
```

Error

SyntaxError: invalid syntax

Point 2

- ✓ We can write an identifier with number but identifier should not start with digit.

Program Name Creating a valid identifier
demo3.py

```
student_id123 = 101  
print(student_id123)
```

Output
101

Program Name Creating an invalid identifier
demo4.py

```
123tudent_id = 101  
print(123tudent_id)
```

Error
SyntaxError: invalid decimal literal

Point 3

- ✓ Identifiers are case sensitive.

Program Name Creating a valid identifier
demo5.py

```
value = 10  
print(value)
```

Error 10

Program Name Identifier is a case sensitive
demo5.py

```
value = 10  
print(VALUE)
```

Error **NameError**: name 'VALUE' is not defined

Point 4

- ✓ We cannot use keywords as identifiers.

Program Name We should not use keywords to create an identifiers
demo6.py

```
if = 10  
print(if)
```

Error
SyntaxError: invalid syntax

Point 5

- ✓ Spaces are not allowed in between the identifier.

Program Name Spaces not allowed between identifier
demo7.py

```
student id = 101  
print(student id)
```

Error
SyntaxError: invalid syntax

Examples

✓ 435student	#	invalid
✓ student564	#	valid
✓ student565info	#	valid
✓ \$tudent	#	invalid
✓ _student_info	#	valid
✓ class	#	invalid
✓ def	#	invalid

Common error

✓ **SyntaxError**: invalid syntax

4. Python identifiers table

✓ This table we can understand while studying upcoming topics.

Identifier	Conventions to follow for identifiers
1. class	<ul style="list-style-type: none">✓ In python, a class name should start with upper case and remaining letters are in lower case.✓ If name having multiple words, then every nested word should start with upper case letter.<ul style="list-style-type: none">○ Example: StudentInfo✓ Info: This rule is applicable for classes created by users only; the in-built class names used all are in lower-case.
2. package 3. module 4. variable 5. function 6. method	<ul style="list-style-type: none">✓ Names should be in lower case.✓ If name having multiple words, then separating words with underscore (_) is good practice.<ul style="list-style-type: none">○ Example: student_id
7. Non-public instance variables	<ul style="list-style-type: none">✓ Non-public instance variables should begin with underscore (_), we can say private data.<ul style="list-style-type: none">○ Example: _balance
8. constants	<ul style="list-style-type: none">✓ Names should be in upper case.✓ If name having multiple words, then separating words with underscore (_) is good practice.

	<ul style="list-style-type: none">○ Example: IN_PROGRESS
9. Non-accessible entities	<ul style="list-style-type: none">✓ Few variables, class constructors (topic in object oriented programming) names having two underscores symbols starting and ending<ul style="list-style-type: none">○ Example: __init__(self)

5. Comments in program

✓ There are two types of comments

1. Single line comments
2. Multi line comments

Purpose of comments

- ✓ Comments are useful to describe about the code in an easy way.
- ✓ Python ignores comments while running the program.

1. Single line comments

- ✓ By using single line comment, we can comment only a single line.
- ✓ To comment single line, we need to use hash symbol **#**

Program Name	A program with single line comment demo8.py
	<pre>#This is Basic program in python print("Welcome to python programming")</pre>
output	Welcome to python programming

2. Multi line comments

- ✓ By using multi line comment we can comment multiple lines.
- ✓ To comment multiple lines, we need to use triple double quotes symbol.

Program A program with multi line comments
Name demo9.py

```
"""Author Daniel  
Project Python project  
Location Bengaluru"""
```

```
print("Welcome to python programming")
```

output Welcome to python programming