
# What is Keyword or reserved word?

# Keyword/Reserved Word

What is Keyword?

Keywords are also called as reserved words these are having special meaning in python language. The words are defined in the python interpreter hence these cant be used as programming identifiers.

# Some Keywords of Python Language

| | |
|---|---|
| and | assert |
| break | class |
| continue | def |
| del | elif |
| else | except |
| exec | finally |
| for | from |

# Some Keywords of Python Language

| | |
|---|---|
| global | if |
| import | in |
| is | lambda |
| not | or |
| pass | print |
| raise | return |
| try | while |
| with | yield |

# What is an identifier?

# IDENTIFIERS


What is an identifier?

A Python Identifier is a name given to a function, class, variable, module, or other objects that you'll be using in your Python program.

In short, its a name appeared in the program.

For example: a, b, c

a b and c are the identifiers and

a b & c and , are the tokens


# PYTHON NAMING CONVENTIONS

# PYTHON NAMING CONVENTIONS


What are the python naming conventions?

- An identifier can be a combination of uppercase letters, lowercase letters, underscores, and digits (0-9).
- Hence, the following are valid identifiers: myClass, my_variable, var_1, and print_hello_world.

# PYTHON NAMING CONVENTIONS

What are the python naming conventions?

- The first character must be letter.
- Special characters such as %, @, and $ are not allowed within identifiers.
- An identifier should not begin with a number. Hence, 2variable is not valid, but variable2 is acceptable.

# PYTHON NAMING CONVENTIONS


What are the python naming conventions?

- Python is a case-sensitive language and this behaviour extends to identifiers. Thus, Labour and labour are two distinct identifiers in Python.
- You cannot use Python keywords as identifiers.

# PYTHON NAMING CONVENTIONS


What are the python naming conventions?

- You cannot use Python keywords as identifiers.
- You can use underscores to separate multiple words in your identifier.

# PYTHON NAMING CONVENTIONS


## SOME VALID IDENTIFIERS:

| | |
|---|---|
| Myfile1 | DATE9_7_8 |
| y3m9d3 | _xs |
| MYFILE | _FXd |

## SOME INVALID IDENTIFIERS:

| | | |
|---|---|---|
| MY-REC | 28dre | break |
| elif | false | del |

# Things to Remember

- Python is a case-sensitive language. This means, Variable and variable are not the same.
- Always give the identifiers a name that makes sense. While c = 10 is a valid name, writing count = 10 would make more sense, and it would be easier to figure out what it represents when you look at your code after a long gap.
- Multiple words can be separated using an underscore, like this_is_a_long_variable.

# Swap two numbers without using third variable

x = 5.4

y = 10.3

After swapping X and Y, we get :

x = 10.3

y = 5.4

# Swap: Using simple built-in method

```
x = 5.4
y = 10.3

# Swap Code
x, y = y, x
```

# Swap: Using Addition and Subtraction Operators


```
x = 5.4
y = 10.3


# Swap Code
x = x + y
y = x - y
x = x - y
```

# Swap: Using Division and Multiplication Operators


```
x = 5.4
y = 10.3


# Swap code
x = x * y
y = x / y
x = x / y
```

# Reading Input from user

**input ():**

- This function first takes the input from the user and converts it into a string.
- The type of the returned object always will be <type ‘str’>. It does not evaluate the expression it just returns the complete statement as String.
  - For example, Python provides a built-in function called input which takes the input from the user. When the input function is called it stops the program and waits for the user’s input. When the user presses enter, the program resumes and returns what the user typed.

Syntax:

```
inp = input('STATEMENT')
```

# Example

```
num = input ("Enter number :")
print(num)
name1 = input("Enter name : ")
print(name1)

# Printing type of input value
print ("type of number", type(num))
print ("type of name", type(name1))
```