Cheat sheet

Syntax

The print function is used to display or print output.

```
>>> print("Hello, Welcome to Simplilearn!")
Hello, Welcome to Simplilearn!
```

Indentation in Python

Indentation means leaving a TAB space from margin and in python it is used to show that statements which are indented belong to the upper statement.

Comments

Single Line Comments	#Comment Code
Multi-Line Comments	‘’’ Comment Code ‘’’

Variables

Variables are containers that are used to store values.

Rules for defining a variable in Python :-

- Variable name can contain alphabets, digits, and underscore (_). Eg: Si_mpli1 = 65.85
- Variable name can start with an alphabet and underscore only. Eg: _simpli = “Welcome”
- It can’t start with a digit.
- No whitespace and reserved keywords are allowed to be used as a variable name.
- Variables are case-sensitive.

Data types

Variables can store data of different types, and different types can do different things.

Text Type:	a = "Hello" #strings
Numeric Types:	x = 7 # int y x = 7.8 # float z z = 7j # complex
Sequence Types:	list, tuple, range
Set Types:	set

#To verify the type of any object in Python, use the type() function:

```
#x = 1    # int
#y = 2.8  # float
#z = 1j   # complex

print(type(x))
print(type(y))
print(type(z))
```

Python Casting

Casting is done when you want to specify a type on a variable.

- x = int(1)
  print(x)
- x = float(1)
  print(x)
- x = str(1)
  print(x)

Operators

Operators are symbols that operate on variables and values.

Arithmetic operators

+	Adding two values	x + y
-	Subtract one value from another	x - y
*	Multiplies two values	x * y
/	Divides one value to another	x / y
%	Returns the division remainder	x % y
**	Exponentiation	x**y
//	Floor division rounds the result down to the nearest whole number	x // y

Assignment operator

=	x=2	x=2
+=	x+=2	x=x+2
-=	x-=2	x=x-2
*=	x*=2	x=x*

Comparison operator

Comparison operators are used to compare two values:

==	Equal to	x == y
!=	Not equal to	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than equal to	x >= y
<=	Less than equal to	x <= y

Logical operator

Logical operators are used to combine conditional statements:

and	Logical and	Returns True if both statements are true
or	Logical or	Returns True if one of the statements is true
not	Logical not	Reverse the result, returns False if the result is true

Input Method

The input function is used to take input as string or character from the user as follows:

```
var1 = input("Enter your name: ")
print("My name is: ", var1)
```

To take input in form of other data types we need to typecast them as follows:
To take input as an integer:

```
var1=int(input("enter the integer value"))
print(var1)
```

Introduction to Python

Python is regarded as one of the most simple, powerful, and versatile programming languages available. According to the most recent figures, Python is the most used coding language for about 80% of developers. The syntax of this high-level programming language is simple to understand.

Strings

Python string is a sequence of characters, and each character can be individually accessed using its index.

```
a = "Hello"
print(a)
```

Indexing

The position of every character placed in the string starts from 0th position and step by step it ends at length-1 position.

Slicing

Slicing refers to obtaining a sub-string from the given string.

```
var_name[1 : 5]
```

Lists

A List in Python represents a list of comma-separated values of any data type between square brackets.

var_name = [element1, element2, ...]

- These elements can be of different data types
- List is ordered,indexed,mutable and the most flexible and dynamic collection of elements in python.
- Duplicate elements are allowed in lists.

Tuples

Tuples are represented as comma-separated values of any data type within parentheses.

variable_name = (element1, element2, ...)

- These elements can be of different data types
- Tuples are ordered,indexing,immutable and most secured collection of elements
- Duplicate elements are allowed in tuples.

tuple.count(value)

It returns the number of times a specified value occurs in a tuple

Sets

A set is a collection of multiple values which is both unordered and unindexed. It is written in curly brackets.

Set Creation: Way 1

var_name = {element1, element2, ...}

Set Creation: Way 2

var_name = set([element1, element2, ...])

- Set is an unordered,immutable,non-indexed type of collection.
- Duplicate elements are not allowed in sets.

Dictionaries

The dictionary is an unordered set of comma-separated key:value pairs, within {}, with the requirement that within a dictionary, no two keys can be the same.

Syntax:

<dictionary-name> = {<key>: value, <key>: value ...}

- Dictionary is an ordered and mutable collection of elements.
- Dictionary allows duplicate values but not duplicate keys.

Control Statements

The if, elif and else statements are the conditional statements in Python, and these implement selection constructs (decision constructs).

if Statement

```
if(conditional expression):
    statements
```

if-else Statement

```
if(conditional expression):
    statements
else:
    Statements
```

if-elif Statement

```
if (conditional expression):
    statements
elif (conditional expression):
    statements
else:
    statements
```

Nested if-else Statement

```
if (conditional expression):
    if (conditional expression):
        statements
    else:
        statements
else:
    Statements
```

Loops

A loop or iteration statement repeatedly executes a statement, known as the loop body, until the controlling expression is false (0).

For Loop

The for loop of Python is designed to process the items of any sequence, such as a list or a string, one by one.

Syntax:

```
for <variable> in <sequence>:
    statements_to_repeat
```

While Loop

A while loop is a conditional loop that will repeat the instructions within itself as long as a conditional remains true.

Syntax:

```
while <logical-expression>:
    Loop-body
```

Functions

A function is a block of code that performs a specific task. You can pass parameters into a function. It helps us to make our code more organized and manageable.

Function Definition:

```
def my_function():
        #statements
```

def keyword is used before defining the function.

Function Call

```
my_function()
```

File Handling

File handling refers to reading or writing data from files. Python provides some functions that allow us to manipulate data in the files.

open() function

```
var_name = open("file name", " mode")
```

Modes	Functionality description
r	To read the content from file
w	To write the content into file
a	To append the existing content into file
r+	To read and write data into the file. The previous data in the file will be overridden.
w+	To write and read data. It will override existing data.
a+	To append and read data from the file. It won’t override existing data.

(OOPs)

It is a programming approach that primarily focuses on using objects and classes. The objects can be any real-world entities.

class

Syntax:

```
class class_name:
    pass #statements
```

Creating an object

Instantiating an object can be done as follows:

Syntax:

```
<object-name> = <class-name>(<arguments>)
```

All classes have a function called __init__(), which is always executed when the class is being initiated.

Eg:

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

Fibonacci sequence

```
n = int(input("How many terms? "))

# first two terms
n1, n2 = 0, 1
count = 0

# check if the number of terms is valid
if n <= 0:
   print("Please enter a positive integer")
# if there is only one term, return n1
elif n== 1:
   print("Fibonacci sequence upto",n,":")
   print(n1)
# generate fibonacci sequence
else:
   print("Fibonacci sequence:")
   while count < n:
       print(n1)
       nth = n1 + n2
       # update values
       n1 = n2
       n2 = nth
       count += 1
```

Palindrome

```
num = int(input("Enter a value:"))
temp = num
rev = 0
while(num > 0):
    dig = num % 10
    rev = rev * 10 + dig
    num = num // 10
if(temp == rev):
    print("This value is a palindrome number!")
else:
    print("This value is not a palindrome number!")
```

simplilearn

Data Scientist Master's Program

http://bitly.ws/x73H