



Skills
Network

Module 1 Cheatsheet: Python Basics

Package/Method	Description	Code Example
Comments	Comments are lines of text that are ignored by the Python interpreter when executing the code	<pre>1. 1 1. # This is a comment</pre> <div>Copied!</div> <p>Syntax:</p> <pre>1. 1 1. variable_name = value</pre> <div>Copied!</div>
Variable Assignment	Assigns a value to a variable	<p>Example:</p> <pre>1. 1 2. 2 1. name="John" # assigning John to variable name 2. x = 5 # assigning 5 to variable x</pre> <div>Copied!</div>
Data Types	<ul style="list-style-type: none">• Integer• Float• Boolean• String	<p>Example:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 1. x=7 # Integer Value 2. y=12.4 # Float Value 3. is_valid = True # Boolean Value 4. is_valid = False # Boolean Value</pre>

`print()` Prints the message or variable inside `()`

Concatenation Combines (concatenates) strings

`len()` Returns the length of a string

`upper()` Converts string to uppercase

```
5. F_Name = "John" # String Value
```

Copied!

Example:

```
1. 1
2. 2
```

```
1. print("Hello, world")
2. print(a+b)
```

Copied!

Syntax:

```
1. 1
```

```
1. concatenated_string = string1 + string2
```

Copied!

Example:

```
1. 1
```

```
1. result = "Hello" + " John"
```

Copied!

Syntax:

```
1. 1
```

```
1. len(string_name)
```

Copied!

Example:

```
1. 1
2. 2
```

```
1. my_string="Hello"
2. length = len(my_string)
```

Copied!

Example:

lower()

Converts string to lowercase

```
1. 1
2. 2
```

```
1. my_string="Hello"
2. uppercase_text = my_string.upper()
```

Copied!

Example:

```
1. 1
2. 2
```

```
1. my_string="Hello"
2. uppercase_text = my_string.lower()
```

Copied!

Syntax:

```
1. 1
```

```
1. substring = string_name[start:end]
```

Copied!

Slicing

Extracts a portion of the string

```
1. 1
2. 2
```

```
1. my_string="Hello"
2. substring = my_string[0:5]
```

Copied!

Example:

```
1. 1
2. 2
```

```
1. my_string="Hello"
2. char = my_string[0]
```

Copied!

Example:

```
1. 1
```

Indexing

Accesses character at a specific index

strip()

Removes leading/trailing whitespace

replace()

Replaces substrings

2. 2

```
1. my_string="Hello"
2. trimmed = my_string.strip()
```

Copied!

Example:

```
1. 1
2. 2
```

```
1. my_string="Hello"
2. new_text = my_string.replace("Hello", "Hi")
```

Copied!

Example:

```
1. 1
2. 2
```

```
1. my_string="Hello"
2. split_text = my_string.split(",")
```

Copied!

Example:

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
```

```
1. x = 9
2. y = 4
3. result_add= x + y    # Addition
4. result_sub= x - y    # Subtraction
5. result_mul= x * y    # Multiplication
6. result_div= x / y    # Division
7. result_fdiv= x // y  # Floor Division
8. result_mod= x % y    # Modulo
```

Copied!

split()

Splits string into a list based on a delimiter

Python Operators

- Addition (+): Adds two values together.
- Subtraction (-): Subtracts one value from another.
- Multiplication (*): Multiplies two values.
- Division (/): Divides one value by another, returns a float.
- Floor Division (//): Divides one value by another, returns the quotient as an integer.
- Modulo (%): Returns the remainder after division.

Author(s)

Pooja Patel

Contributor(s)

[Malika Singla](#)

Changelog

Date	Version	Changed by	Change Description
2023-08-10	0.1	Pooja Patel	Initial version created