

Module 1 Cheat Sheet: Python Basics

Package/Method	Description	Code Example
Comments	Comments are lines of text that are ignored by the Python interpreter when executing the code<./td>	<pre>1. 1 1. # This is a comment</pre> <div>Copied!</div> <p>Syntax:</p> <pre>1. 1 1. concatenated_string = string1 + string2</pre> <div>Copied!</div> <p>Example:</p> <pre>1. 1 1. result = "Hello" + " John"</td></pre> <div>Copied!</div> <p>Example:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 1. x=7 2. # Integer Value 3. y=12.4 4. # Float Value 5. is_valid = True 6. # Boolean Value 7. is_valid = False 8. # Boolean Value 9. F_Name = "John" 10. # String Value</pre> <div>Copied!</div> <p>Example:</p> <pre>1. 1 2. 2 1. my_string="Hello" 2. char = my_string[0]</pre> <div>Copied!</div> <p>Syntax:</p> <pre>1. 1 1. len(string_name)</pre> <div>Copied!</div> <p>Example:</p>
Concatenation	Combines (concatenates) strings.	
Data Types	- Integer - Float - Boolean - String	
Indexing	Accesses character at a specific index.	
len()	Returns the length of a string.	

`lower()` Converts string to lowercase.

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

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.

`replace()` Replaces substrings.

Slicing Extracts a portion of the string.

about:blank

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

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

Copied!

Example:

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

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

Copied!

Example:

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

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

Copied!

Example:

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

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

Copied!

Example:

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

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

Copied!

Syntax:

```
1. 1
```

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

Copied!

Example:

```
1. 1
```

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

Copied!

`split()` Splits string into a list based on a delimiter.

`strip()` Removes leading/trailing whitespace.

`upper()` Converts string to uppercase.

Variable Assignment Assigns a value to a variable.



Skills Network

© IBM Corporation. All rights reserved.

about:blank

Example:

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

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

Copied!

Example:

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

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

Copied!

Example:

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

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

Copied!

Syntax:

```
1. 1
```

```
1. variable_name = value
```

Copied!

Example:

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

```
1. name="John" # assigning John to variable name
2. x = 5 # assigning 5 to variable x
```

Copied!