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>	```# This is a comment```

Concatenation	Combines (concatenates) strings.	Syntax:
		```concatenated_string = string1 + string2```
		Example:
		```result = "Hello" + " John"</td>```

Data Types	- Integer - Float - Boolean - String	Example:
		```x=7```
		```# Integer Value```
		```y=12.4```
		```# Float Value```
		```is_valid = True```
		```# Boolean Value```
		```is_valid = False```
		```# Boolean Value```
		```F_Name = "John"```
		```# String Value```

Indexing	Accesses character at a specific index.	Example:
		```my_string="Hello"```
		```char = my_string[0]```

len()	Returns the length of a string.	Syntax:
		```len(string_name)```
		Example:
		```my_string="Hello"```
		```length = len(my_string)```

lower()	Converts string to lowercase.	Example:
		```my_string="Hello"```
		```uppercase_text = my_string.lower()```

print()	Prints the message or variable inside `()`.	Example:
		```print("Hello, world")```
		```print(a+b)```

Python Operators	- Addition (+): Adds two values together.
	- Subtraction (-): Subtracts one value from another.
		Example:
		```x = 9 y = 4```
		```result_add= x + y # Addition```
		```result_sub= x - y # Subtraction```

	- 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.
		```result_mul= x * y # Multiplication```
		 ```result_div= x / y # Division```
		 ```result_fdiv= x // y # Floor Division```
		 ```result_mod= x % y # Modulo</td>```

replace()	Replaces substrings.	Example:
		
		```my_string="Hello"```
		 ```new_text = my_string.replace("Hello", "Hi")```

Slicing	Extracts a portion of the string.	Syntax:
		
		```substring = string_name[start:end]```
		
		Example:
		
		```my_string="Hello" substring = my_string[0:5]```

split()	Splits string into a list based on a delimiter.	Example:
		
		```my_string="Hello"```
		 ```split_text = my_string.split(",")```

strip()	Removes leading/trailing whitespace.	Example:
		
		```my_string="Hello"```
		 ```trimmed = my_string.strip()```

upper()	Converts string to uppercase.	Example:
		
		```my_string="Hello"```
		 ```uppercase_text = my_string.upper()```

Variable Assignment	Assigns a value to a variable.	Syntax:
		
		```variable_name = value```
		
		Example:
		
		```name="John" # assigning John to variable name```
		 ```x = 5 # assigning 5 to variable x```

Skills Network