

Python Basics Cheat Sheet

Table of Contents



- [Python Basics Cheat Sheet](#)
 - [1. Python Syntax and Structure](#)
 - [Hello World \(First Program\)](#)
 - [Variables and Data Types](#)
 - [Comments](#)
 - [2. Data Structures](#)
 - [Lists](#)
 - [Tuples \(Immutable Lists\)](#)
 - [Dictionaries \(Key-Value Pairs\)](#)
 - [Sets \(Unique Values\)](#)
 - [3. Conditional Statements](#)
 - [Multiple Conditions](#)
 - [4. Loops](#)
 - [For Loop](#)
 - [While Loop](#)
 - [5. Functions](#)
 - [Default Parameters](#)
 - [6. Classes and Objects](#)
 - [7. Exception Handling](#)
 - [8. File Handling](#)
 - [Read File](#)
 - [Write to File](#)
 - [9. Importing Modules](#)
 - [10. List Comprehensions](#)
 - [11. Useful Built-in Functions](#)
 - [12. Useful Tips](#)
-

1. Python Syntax and Structure

Hello World (First Program)

```
print("Hello, World!")
```

Variables and Data Types

```
name = "Alice"           # String
age = 25                  # Integer
height = 5.8              # Float
is_student = True         # Boolean
```

Comments

```
# This is a single-line comment
```

```
"""
This is a
multi-line comment
"""
```

2. Data Structures

Lists

```
fruits = ["apple", "banana", "cherry"]
fruits.append("orange") # Add item
fruits.remove("banana") # Remove item
print(fruits[0])        # Access first item
```

Tuples (Immutable Lists)

```
coordinates = (10, 20)
print(coordinates[1])    # Access second item
```

Dictionaries (Key-Value Pairs)

```
person = {
    "name": "Bob",
    "age": 30
}
print(person["name"])    # Access value by key
person["age"] = 31       # Update value
```

Sets (Unique Values)

```
unique_numbers = {1, 2, 3, 3, 2}
unique_numbers.add(4)
```

3. Conditional Statements

```
age = 18
if age >= 18:
    print("Adult")
else:
    print("Minor")
```

Multiple Conditions

```
score = 85
if score >= 90:
```

```
        print("Grade A")
elif score >= 75:
    print("Grade B")
else:
    print("Grade C")
```

4. Loops

For Loop

```
for i in range(1, 6):
    print(i)
```

While Loop

```
count = 0
while count < 5:
    print(count)
    count += 1
```

5. Functions

```
def greet(name):
    return f"Hello, {name}!"

print(greet("Alice"))
```

Default Parameters

```
def power(base, exponent=2):
    return base ** exponent
```

```
print(power(3))      # 9
print(power(3, 3))   # 27
```

6. Classes and Objects

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

    def bark(self):
        return "Woof!"

dog1 = Dog("Rex")
print(dog1.bark())
```

7. Exception Handling

```
try:
    x = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero!")
finally:
    print("End of operation")
```

8. File Handling

Read File

```
with open("example.txt", "r") as file:
    content = file.read()
    print(content)
```

Write to File

```
with open("example.txt", "w") as file:  
    file.write("Hello, World!")
```

9. Importing Modules

```
import math  
print(math.sqrt(16))  # 4.0
```

10. List Comprehensions

```
numbers = [x for x in range(10) if x % 2 == 0]  
print(numbers)  # [0, 2, 4, 6, 8]
```

11. Useful Built-in Functions

```
len([1, 2, 3])          # Length of list  
max([10, 5, 7])         # Max value  
min([10, 5, 7])         # Min value  
sum([1, 2, 3, 4])       # Sum of list  
sorted([3, 1, 2])       # Sort list
```

12. Useful Tips

- Use f-strings for easier string formatting:

```
name = "John"
```

```
print(f"My name is {name}")
```

- Use `enumerate()` to get index in loops:

```
for index, value in enumerate(["a", "b", "c"]):  
    print(index, value)
```

- Use `zip()` to iterate over multiple lists:

```
names = ["Alice", "Bob"]  
scores = [90, 80]  
for name, score in zip(names, scores):  
    print(f"{name}: {score}")
```

This cheat sheet provides a quick overview of Python's essential features.