# Python Basics Reference Guide

## Table of Contents

---

## 1. Python Syntax Basics

### Comments

```
# This is a single-line comment

"""
This is a multi-line comment
or docstring. Used for longer explanations
or function documentation.
"""
```

### Code Structure

- Python uses **indentation** instead of braces `{}`
- Standard indentation is **4 spaces**
- Statements end with a new line (no semicolons needed)

```
# Correct indentation
if age >= 18:
    print("You are an adult")
    print("You can vote")
else:
    print("You are a minor")
```

### Case Sensitivity

```
name = "Alice"
Name = "Bob"     # Different variable!
NAME = "Charlie" # Also different!
```

---

# 2. Variables and Assignment

## Variable Naming Rules

- Must start with letter (a-z, A-Z) or underscore (_)
- Can contain letters, numbers, underscores
- Cannot contain spaces or special characters
- Case sensitive
- Cannot use Python keywords

## Good Variable Names

```
first_name = "John"        # Snake case (recommended)
last_name = "Doe"
user_age = 25
is_student = True
MAX_ATTEMPTS = 3           # Constants in UPPERCASE
```

## Multiple Assignment

```
# Assign same value to multiple variables
x = y = z = 0

# Assign different values
name, age, city = "Alice", 25, "Boston"

# Swap variables
a = 10
b = 20
a, b = b, a  # Now a=20, b=10
```

---

# 3. Data Types

## Numeric Types

### Integer (int)

```
age = 25
negative_number = -10
large_number = 1000000
```

### Float

```
price = 19.99
pi = 3.14159
scientific = 1.5e-4  # Scientific notation: 0.00015
```

### Complex (advanced)

```
complex_num = 3 + 4j
```

## String (str)

```
# Single quotes
name = 'Alice'

# Double quotes
message = "Hello, World!"

# Triple quotes (multiline)
paragraph = """This is a long
paragraph that spans
multiple lines."""

# Raw strings (ignore escape characters)
file_path = r"C:\Users\Documents\file.txt"
```

## Boolean (bool)

```
is_active = True
is_complete = False

# Boolean from expressions
is_adult = age >= 18
has_permission = user_type == "admin"
```

## Collections

### List

```
fruits = ["apple", "banana", "orange"]
numbers = [1, 2, 3, 4, 5]
mixed = [1, "hello", True, 3.14]
empty_list = []
```

### Dictionary

```
person = {
    "name": "Alice",
    "age": 30,
    "city": "Boston"
}

# Accessing values
print(person["name"])      # "Alice"
print(person.get("age"))   # 30
```

### Tuple (immutable)

```
coordinates = (10, 20)
rgb_color = (255, 128, 0)
```

### Type Checking

```
name = "Alice"
age = 25

print(type(name))    # <class 'str'>
print(type(age))     # <class 'int'>

# Check specific type
print(isinstance(name, str))    # True
print(isinstance(age, int))     # True
```

---

## 4. Input and Output

### Output with print()

```
# Basic output
print("Hello, World!")

# Multiple values
print("Name:", "Alice", "Age:", 25)

# Custom separator
print("apple", "banana", "orange", sep=", ")
# Output: apple, banana, orange

# Custom end character
print("Loading", end="...")
print("Done!")
# Output: Loading...Done!

# Formatted output
name = "Alice"
age = 25
print(f"My name is {name} and I am {age} years old")
```

### Input with input()

```
# Get string input
name = input("Enter your name: ")

# Get numeric input (convert from string)
age = int(input("Enter your age: "))
price = float(input("Enter price: "))

# Input with validation
while True:
    try:
        age = int(input("Enter your age: "))
        break
    except ValueError:
        print("Please enter a valid number")
```

### String Formatting

```
name = "Alice"
age = 25
score = 87.5

# f-strings (recommended, Python 3.6+)
message = f"Hello {name}, you scored {score:.1f}%"

# .format() method
message = "Hello {}, you scored {:.1f}%".format(name, score)

# % formatting (older style)
message = "Hello %s, you scored %.1f%%" % (name, score)
```

---

## 5. Basic Operations

### Arithmetic Operators

```
a = 10
b = 3

addition = a + b        # 13
subtraction = a - b     # 7
multiplication = a * b  # 30
division = a / b        # 3.333...
floor_division = a // b # 3 (integer division)
modulo = a % b         # 1 (remainder)
exponentiation = a ** b # 1000 (10^3)
```

### Assignment Operators

```
x = 10

x += 5    # x = x + 5, now x = 15
x -= 3    # x = x - 3, now x = 12
x *= 2    # x = x * 2, now x = 24
x /= 4    # x = x / 4, now x = 6.0
x //= 2   # x = x // 2, now x = 3.0
x %= 2    # x = x % 2, now x = 1.0
x **= 3   # x = x ** 3, now x = 1.0
```

### Comparison Operators

```
a = 10
b = 5

print(a > b)      # True
print(a < b)      # False
print(a >= 10)    # True
print(a <= b)     # False
print(a == 10)   # True
print(a != b)    # True
```

### Logical Operators

```
age = 25
has_license = True

# and operator
can_drive = age >= 18 and has_license

# or operator
is_weekend = day == "Saturday" or day == "Sunday"

# not operator
is_not_adult = not (age >= 18)
```

---

## 6. String Manipulation

### String Methods

```
text = "  Hello, World!  "

# Case manipulation
print(text.upper())        # "  HELLO, WORLD!  "
print(text.lower())        # "  hello, world!  "
print(text.title())        # "  Hello, World!  "
print(text.capitalize())   # "  hello, world!  "

# Whitespace removal
print(text.strip())        # "Hello, World!"
print(text.lstrip())       # "Hello, World!  "
print(text.rstrip())       # "  Hello, World!"

# Replacement
print(text.replace("World", "Python"))  # "  Hello, Python!  "

# Splitting and joining
words = "apple,banana,orange".split(",")  # ["apple", "banana", "orange"]
joined = "-".join(words)                  # "apple-banana-orange"
```

### String Information

```
text = "Hello, World!"

print(len(text))              # 13 (length)
print(text.count("l"))        # 3 (count of character)
print(text.find("World"))     # 7 (index of substring)
print(text.startswith("Hi"))  # False
print(text.endswith("!"))     # True
print("World" in text)        # True (membership test)
```

### String Slicing

```
text = "Python Programming"

print(text[0])       # "P" (first character)
print(text[-1])      # "g" (last character)
print(text[0:6])     # "Python" (slice from 0 to 6, exclusive)
print(text[7:])      # "Programming" (from index 7 to end)
print(text[:6])      # "Python" (from start to index 6)
print(text[::2])     # "Pto rgamn" (every 2nd character)
print(text[::-1])    # "gnimmargorP nohtyP" (reverse)
```

---

## 7. Common Patterns

### Input Validation Pattern

```
def get_positive_number(prompt):
    while True:
        try:
            value = float(input(prompt))
            if value > 0:
                return value
            else:
                print("Please enter a positive number.")
        except ValueError:
            print("Please enter a valid number.")

# Usage
age = get_positive_number("Enter your age: ")
```

### Menu Selection Pattern

```
def show_menu():
    print("\n=== Main Menu ===")
    print("1. Option 1")
    print("2. Option 2")
    print("3. Exit")

while True:
    show_menu()
    choice = input("Enter your choice (1-3): ")

    if choice == "1":
        print("You selected Option 1")
    elif choice == "2":
        print("You selected Option 2")
    elif choice == "3":
        print("Goodbye!")
        break
    else:
        print("Invalid choice. Please try again.")
```

## Data Processing Pattern

```
# Read data
data = input("Enter numbers separated by commas: ")
numbers = [float(x.strip()) for x in data.split(",")]

# Process data
total = sum(numbers)
average = total / len(numbers)
maximum = max(numbers)
minimum = min(numbers)

# Display results
print(f"Total: {total}")
print(f"Average: {average:.2f}")
print(f"Maximum: {maximum}")
print(f"Minimum: {minimum}")
```

## Calculator Pattern

```
def simple_calculator():
    print("Simple Calculator")
    print("Operations: +, -, *, /, ** (power), % (modulo)")

    while True:
        try:
            num1 = float(input("Enter first number: "))
            operator = input("Enter operator (+, -, *, /, **, %): ")
            num2 = float(input("Enter second number: "))

            if operator == "+":
                result = num1 + num2
            elif operator == "-":
                result = num1 - num2
            elif operator == "*":
                result = num1 * num2
            elif operator == "/":
                if num2 == 0:
                    print("Error: Cannot divide by zero!")
                    continue
                result = num1 / num2
            elif operator == "**":
                result = num1 ** num2
            elif operator == "%":
                if num2 == 0:
                    print("Error: Cannot divide by zero!")
                    continue
                result = num1 % num2
            else:
                print("Invalid operator!")
                continue

            print(f"Result: {num1} {operator} {num2} = {result}")

            if input("Continue? (y/n): ").lower() != 'y':
                break

        except ValueError:
            print("Invalid input! Please enter valid numbers.")

# Usage
simple_calculator()
```