

Chapter 2: Python Basics

2.1 Variables and Data Types

Variables in Python are used to store data. They do not require explicit declaration and their type is inferred based on the value assigned. Python supports several data types:

- String: A sequence of characters, e.g., "Hello, World!"
- Integer: Whole numbers, e.g., 123
- Float: Numbers with decimal points, e.g., 3.14
- Boolean: Represents True or False.
- List: A collection of ordered, changeable items, e.g., [1, 2, 3, 4, 5]
- Tuple: A collection of ordered, unchangeable items, e.g., (1, 2, 3)
- Dictionary: A collection of key-value pairs, e.g., {"name": "Alice", "age": 25}

Example of variable usage:

```
string = "Hello, World!"
integer = 123
float_num = 3.14
boolean = True
list_items = [1, 2, 3, 4, 5]
tuple_items = (1, 2, 3)
dictionary = {"name": "Alice", "age": 25}

print(string)      # Outputs: Hello, World!
print(integer)     # Outputs: 123
print(float_num)   # Outputs: 3.14
print(boolean)     # Outputs: True
print(list_items)  # Outputs: [1, 2, 3, 4, 5]
print(tuple_items) # Outputs: (1, 2, 3)
print(dictionary)  # Outputs: {'name': 'Alice', 'age': 25}
```

2.2 Operators and Expressions

Operators are used to perform operations on variables and values. Here are some common operators:

Arithmetic Operators: +, -, *, /, %

Assignment Operators: =, +=, -=, *=, /=

Comparison Operators: ==, !=, >, <, >=, <=

Logical Operators: and, or, not

Example:

```
a = 10
```

```
b = 5
```

```
# Arithmetic
```

```
print(a + b) # Outputs: 15
```

```
# Comparison
```

```
if a > b:
```

```
    print("a is greater than b")
```

```
# Logical
```

```
if a > 5 and b < 10:
```

```
    print("Both conditions are true")
```

2.3 Conditional Statements

Conditional statements are used to perform different actions based on different conditions. The most common are if, elif, and else.

Example:

```
num = 10
```

```
if num > 0:
```

```
    print("The number is positive.")
```

```
elif num < 0:
```

```
    print("The number is negative.")
```

```
else:
```

```
    print("The number is zero.")
```

2.4 Loops in Python

Loops are used to execute the same block of code repeatedly as long as a specified condition is met. Common loops include for and while.

Example of a for loop:

```
for i in range(5):
```

```
    print(f"Value of i: {i}")
```

Example of a while loop:

```
count = 0  
  
while count < 5:  
  
    print(f"Count: {count}")  
  
    count += 1
```

2.5 Functions

Functions are blocks of code that can be repeatedly called and executed whenever needed. They are defined using the `def` keyword.

Example:

```
def greet(name):  
  
    return f"Hello, {name}!"  
  
  
print(greet("Alice")) # Outputs: Hello, Alice!
```