# Python Variables and Data Types

## Understanding Variables

### What are Variables?

Variables in Python are containers for storing data values. Think of them as labeled boxes that can hold different types of information.

### Variable Naming Rules

- Must start with a letter or underscore
- Can contain letters, numbers, and underscores
- Case-sensitive (age and Age are different variables)
- Cannot use Python keywords (like 'if', 'for', 'while')

Good examples:

```
user_name = "John"
age_1 = 25
_private = "hidden"
```

Bad examples:

```
1user = "John"    # Cannot start with number
user-name = "John" # Cannot use hyphen
if = "test"       # Cannot use Python keyword
```

## Data Types in Python

### Numeric Types

**Integers (int)** Whole numbers, positive or negative:

```
age = 25
temperature = -5
```

**Floating-Point Numbers (float)** Numbers with decimal points:

```
height = 1.75
pi = 3.14159
```

**Complex Numbers** Numbers with real and imaginary parts:

```
z = 3 + 4j
```

### Text Type

**Strings (str)** Text enclosed in single or double quotes:

```
name = "Alice"
message = 'Hello, World!'
multi_line = '''This is a
multi-line string'''
```

String operations:

```
# Concatenation
first = "Hello"
second = "World"
greeting = first + " " + second

# String methods
text = "Python"
print(text.upper())      # PYTHON
print(text.lower())      # python
print(len(text))         # 6
```

### Boolean Type (bool)

Represents True or False values:

```
is_student = True
is_working = False
```

### Sequence Types

**Lists** Ordered, mutable sequences:

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

**Tuples** Ordered, immutable sequences:

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

**Range** Sequence of numbers:

```
numbers = range(5)       # 0, 1, 2, 3, 4
even = range(0, 10, 2)   # 0, 2, 4, 6, 8
```

### Mapping Type

**Dictionaries (dict)** Key-value pairs:

```
person = {
    "name": "John",
    "age": 30,
```

```
    "city": "New York"
}
```

#### Set Types

**Set** Unordered collection of unique elements:

```
unique_numbers = {1, 2, 3, 3, 4}  # {1, 2, 3, 4}
```

### Type Conversion

#### Explicit Type Conversion (Type Casting)

```
# String to Integer
age_str = "25"
age_int = int(age_str)

# Integer to Float
num_int = 10
num_float = float(num_int)

# Number to String
number = 42
str_number = str(number)
```

### Variable Assignment

#### Multiple Assignment

```
# Multiple variables, same value
x = y = z = 0

# Multiple variables, different values
a, b, c = 1, 2, 3
```

#### Augmented Assignment

```
count = 0
count += 1  # Increment by 1
count -= 1  # Decrement by 1
count *= 2  # Multiply by 2
count /= 2  # Divide by 2
```

### Memory Management

Python handles memory management automatically through garbage collection.

## Best Practices

### Naming Conventions

- Use lowercase letters for variable names
- Use underscores for multi-word variables
- Choose descriptive names

### Type Hints (Python 3.5+)

Optional type hints for better code documentation:

```
age: int = 25
name: str = "Alice"
scores: list[int] = [85, 92, 78]
```

## Common Pitfalls

### Mutable vs Immutable Types

- Immutable: int, float, str, tuple
- Mutable: list, dict, set

### Variable Scope

Understanding local and global variables:

```
global_var = "I'm global"

def function():
    local_var = "I'm local"
    print(global_var)  # Accessible
    print(local_var)   # Accessible

print(global_var)      # Accessible
print(local_var)       # Error! Not accessible
```