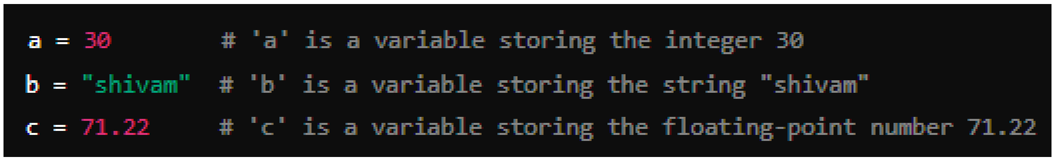
Chapter 2 – Variables and Data Types

Variables

A variable is a name given to a memory location in a program. It acts as a container to store a value.

Examples:



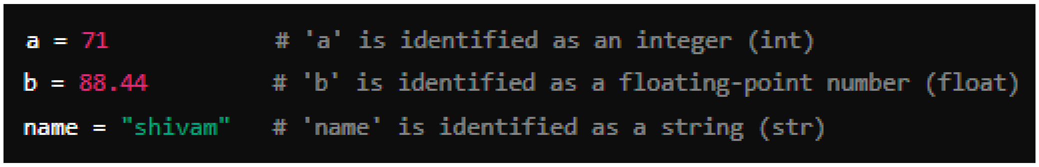
```
a = 30        # 'a' is a variable storing the integer 30
b = "shivam"  # 'b' is a variable storing the string "shivam"
c = 71.22     # 'c' is a variable storing the floating-point number 71.22
```

Data Types

In Python, data types specify the type of data that a variable can hold. The primary data types in Python are:

1. **Integers**: Whole numbers without a fractional part.
2. **Floating point numbers**: Numbers with a decimal point.
3. **Strings**: Sequence of characters.
4. **Booleans**: Represents `True` or `False`.
5. **None**: Represents a null value or no value at all.

Examples:



```
a = 71          # 'a' is identified as an integer (int)
b = 88.44       # 'b' is identified as a floating-point number (float)
name = "shivam" # 'name' is identified as a string (str)
```

Rules for Choosing an Identifier

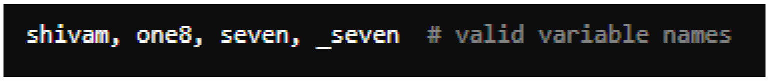
An identifier is a name given to a variable, function, class, etc.

Rules:

- An identifier can contain alphabets, digits, and underscores.
- An identifier must start with an alphabet or an underscore.
- An identifier cannot start with a digit.

- No whitespace is allowed in an identifier.

Examples:



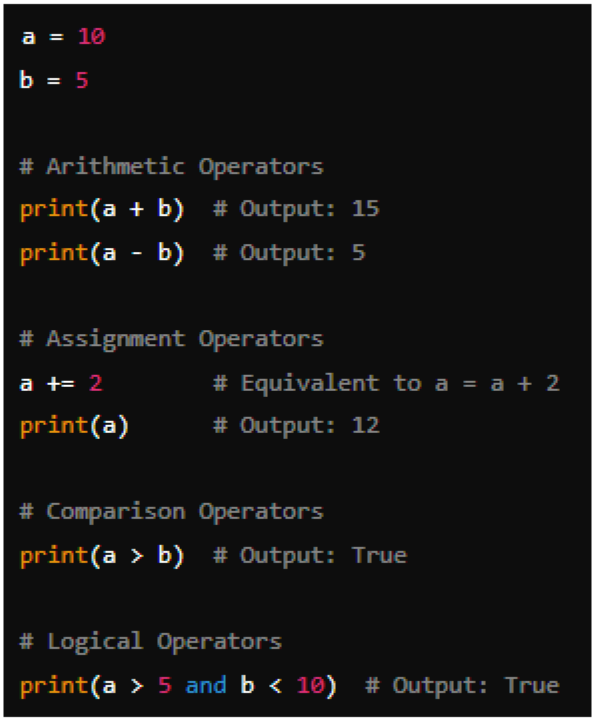
```
shivam, one8, seven, _seven  # valid variable names
```

Operators in Python

Common Operators:

1. **Arithmetic operators**: `+`, `-`, `*`, `/`, etc.
2. **Assignment operators**: `=`, `+=`, `-=`, etc.
3. **Comparison operators**: `==`, `>`, `>=`, `<`, `!=`, etc.
4. **Logical operators**: `and`, `or`, `not`.

Examples:

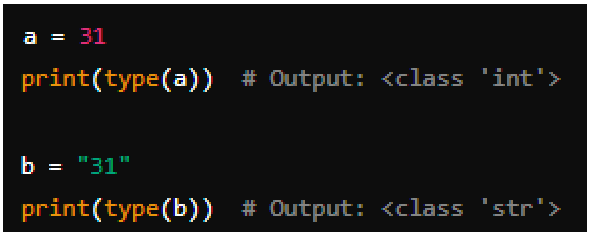


```
a = 10
b = 5

# Arithmetic Operators
print(a + b)  # Output: 15
print(a - b)  # Output: 5

# Assignment Operators
a += 2        # Equivalent to a = a + 2
print(a)      # Output: 12

# Comparison Operators
print(a > b)  # Output: True

# Logical Operators
print(a > 5 and b < 10)  # Output: True
```

`type()` Function and Typecasting

The `type()` function is used to find the data type of a given variable.

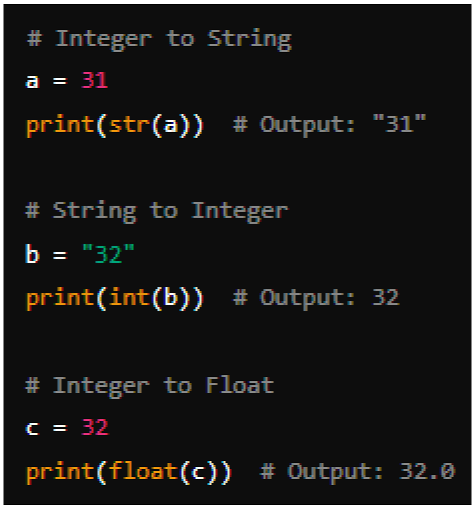
Examples:



```python
a = 31
print(type(a))  # Output: <class 'int'>

b = "31"
print(type(b))  # Output: <class 'str'>
```

Typecasting is the process of converting one data type to another.

Examples:

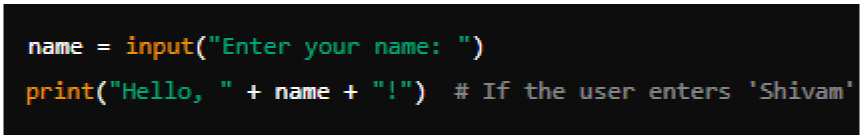


```python
# Integer to String
a = 31
print(str(a))  # Output: "31"

# String to Integer
b = "32"
print(int(b))  # Output: 32

# Integer to Float
c = 32
print(float(c))  # Output: 32.0
```

`input()` Function

The `input()` function allows the user to take input from the keyboard as a string.

Example:



```python
name = input("Enter your name: ")
print("Hello, " + name + "!")  # If the user enters 'Shivam'
```