

2

Day 2: Variables and Data Types

Comments

1. Single line: #
2. Multi line: """ This is comment."""

Variables

1. Containers for storing values

```
x = 5

y = "Rohit"

print(x)

print(y)
```

2. Casting

Variables need not to be declared, but if you want you can specify the data type of a variable.

```
x = str(3)    # x will be '3'
y = int(3)    # y will be 3
z = float(3)  # z will be 3.0

# Get the type
x = 5
y = "John"
```

```
print(type(x))
print(type(y))
```

3. Single or double quotes for string

You can use either.

4. Variables are case sensitive

```
a = 4
A = "Sally"
print(a)
print(A)
#A will not overwrite a
```

5. Variable Names: snake_case and camelCase

6. Assign multiple values

```
# Many value to multiple variables
x, y, z = "Orange", "Banana", "Cherry"

# One value to multiple variables
x = y = z = "Orange"

# If you have a collection of values in list, tuple, etc.
fruits = ["apple", "banana", "cherry"]
x, y, z = fruits
print(x)
print(y)
print(z)
```

Data Types

Text Type:	<code>str</code>
Numeric Types:	<code>int</code> , <code>float</code>

Boolean Type

bool

```
x = 5
print(type(x))
```

Numbers are of 2 types - int and float.

Float can also be scientific numbers with an "e" to indicate the power of 10.

```
x = 35e3
y = 12E4
z = -87.7e100

print(type(x))
print(type(y))
print(type(z))
```

Type Conversion

```
x = 1      # int
y = 2.8    # float

#convert from int to float:
a = float(x)

#convert from float to int:
b = int(y)
```

Random Number

Python does not have a `random()` function to make a random number, but Python has a built-in module called `random` that can be used to make random numbers:

```
import random
print(random.randrange(1, 10))
```

Type Casting

Type casting is manual and type conversion is automatic.

Operators

Types of operators:

- Arithmetic operators (+, -, *, /, %, **, //)
- Assignment operators(=, +=, -=, *=, /=)
- Comparison operators(==, !=, >, <, >=, <=)
- Logical operators(and, or, not)
- Identity operators
- Membership operators
- Bitwise operators