Data Types

Chapter No.	1
Status	Completed

- A variable is a name created in a program to represent something

▼ In Python, every "thing" is considered to be an object

- Each object has an identity, some attributes and some (possibly zero) names

▼ id( )

- Used to retrieve an object's ID number, which indicates the specific location in the computer's memory where the object is stored

▼ Assigning Values to Variables

```
a = 1
x, y = 50, 60
```

▼ Data Types

Data Types

Type	Description	Examples
int (integer)	This corresponds to our mathematical definition of an integer. They can be positive or negative.	1, 23 -100
float (floating-point number)	This roughly corresponds to real numbers.	2.71, -3.5

Type	Description	Examples
bool (Boolean value)	True or False.	True, False
str (string)	A collection of characters in a sequence, delimited by single quotes (') or double quotes (")	"i am george"
list	A collection of objects in a sequence, delimited by square brackets ([and]). The objects do not need to be all of the same type.	[1, 2.5, "abc"]
set	A collection of unique elements, delimited by curly braces ({ and }). The order does not matter.	{1, 2.5, "abc"}
dict (dictionary)	A set of key-value pairs. The first element in each pair is the key and the second element is the value. The key can be used to look up the value.	{"Jack":1, "Jones":2}

▼ type( )

- Returns an object's type

▼ Typecasting

- It is also possible to change the type of an object through typecasting
- Most of the time, Python will make a new object of the specified type which is most similar to the old object

▼ Example

```
x = int(1.6)
y = str(2000)
```

▼ Operators

Operators

Operator	Description
+	Addition
-	Subtraction

Operator	Description
*	Multiplication
/	Division
//	Floor Division (quotient)
%	Modulo (remainder)
**	Exponentiation

- Mixing integers and floats always produces a float as the result
- The result of dividing an integer by another is always a float, even if the division does not have a remainder
- Parentheses and order of operations also work as they do in mathematics

▼ Reassigning Variables

```
x = 1

x = x + 1

x += 1

a = 3, b = 5
a, b = b, a
```

▼ Rules for Naming Variables

1. Every name must begin with a letter or an underscore (_)
    - A number is not allowed as the first character
    - Multiple-word names can be linked together using underscores
    - A name that actually starts with an underscore is usually used to denote a variable with special characteristics
2. After the first letter, the name may contain any combination of letters, numbers and underscores
    - The name cannot be a keyword used by Python
    - The name cannot contain any delimiters, punctuation or operators

- The name can be of any length
- The name is case sensitive