Newton School of Technology

2

Variables and Datatypes

by Kunal Goyal

CSA101 : Problem Solving with Programming

Newton School of Technology

2

Join the lecture online on your dashboard.

Introduction to Variable :

Newton School of Technology

- Just like **water needs a container (glass, bowl, or bottle)** to be stored, values need variables.
- **A variable stores data, and its type** (int, float, string, etc.) is like the container shape.

Normal Water

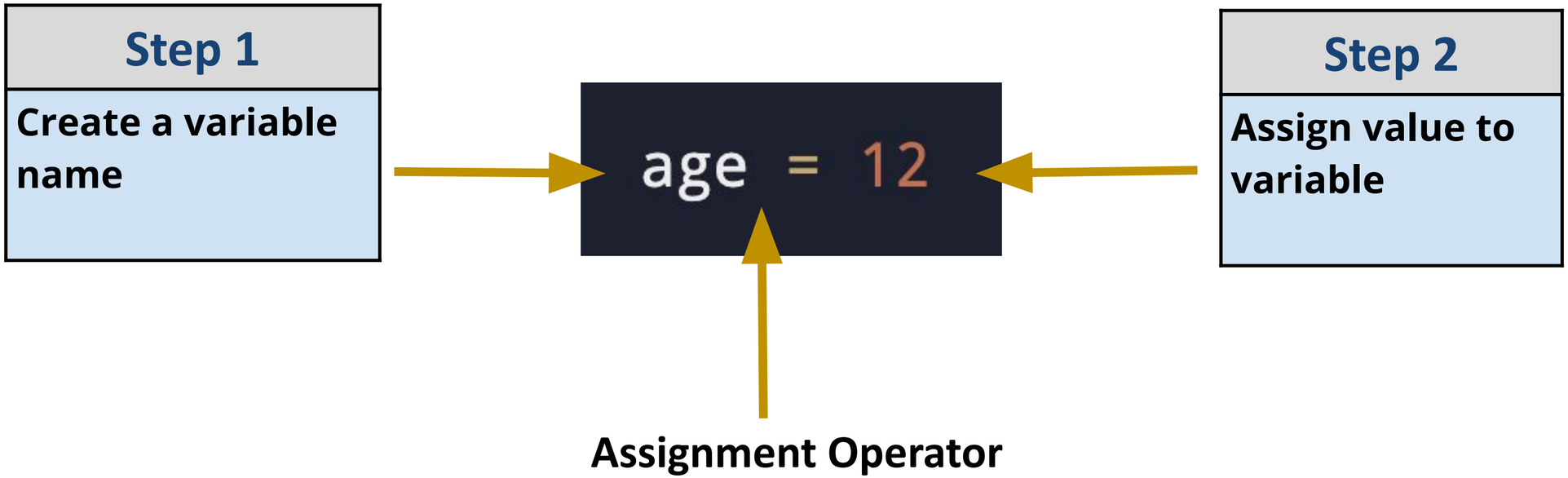
Warm Water

Frozen Water

Newton School of Technology

Variables in Programming

Variables in Python :



Variables in Python :

- **Step 1 : Choose a variable name**
- **Step 2 : Assign a value**
- **Step 3 : Let’s print it in your playground!**

```
num_1 = 57
num_2 = 65
print(num_1)
print(num_2)
```

```
Output
57
65
```

Newton School of Technology

Variable Naming Rules :

Rules (Strict — must follow) :

1. Must start with a letter (a–z, A–Z) or underscore (_)

✅ **valid :** name, _temp ❌ **invalid:** 1name, @value

2. Can contain letters, digits, and underscores only

✅ **valid :** total_1, data_2024 ❌ **invalid:** total-price, data%count

3. Case-sensitive : Name, name, and NAME are all different variables.

4. Variable name cannot be a Python keyword (like if, else, for, class, True, None, etc.)

Newton School of Technology

Is this a correct name ?

```
1st_value = 23
```

Newton School of Technology

Is this a correct name ?

```
book-name = “Python Programming”
```

Newton School of Technology

Is this a correct name ?

```
if = 123
```

Newton School of Technology

Is this a correct name ?

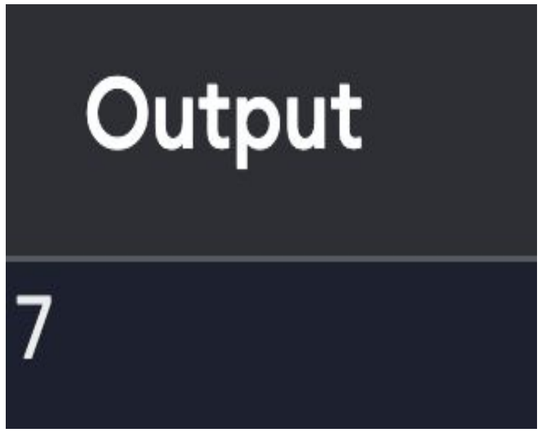
```
book_name = “Python Programming”
```

Re Assigning Variables :

Variable can be **assigned a new value**.

```
x = 5
x = 7

print(x)
```

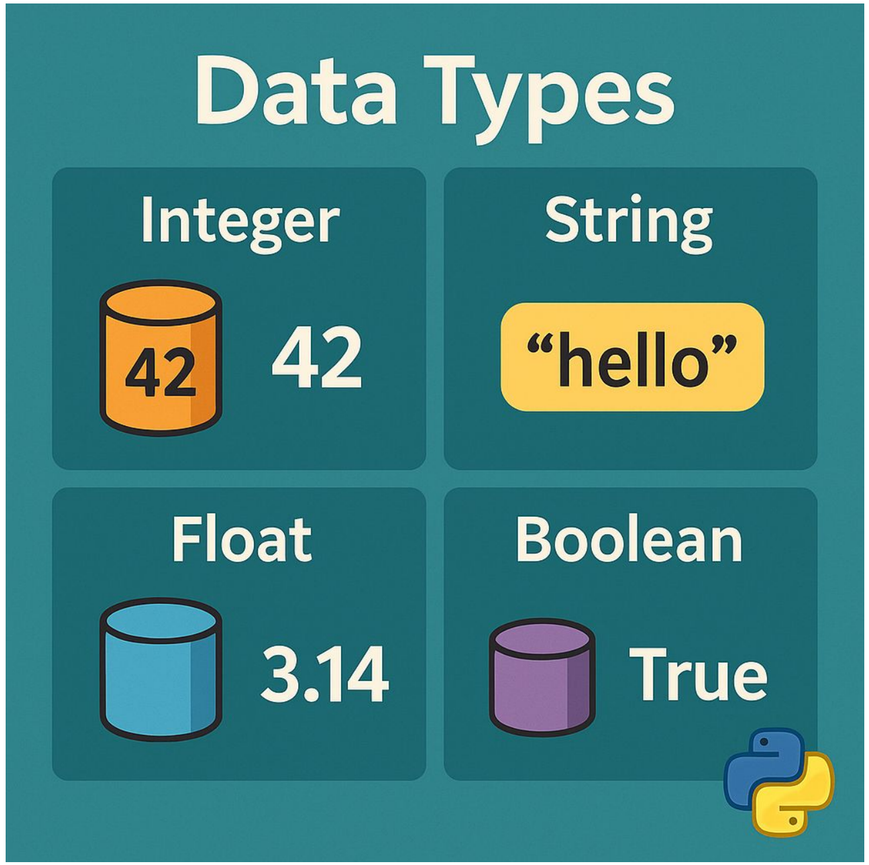


Newton School of Technology

Data Types

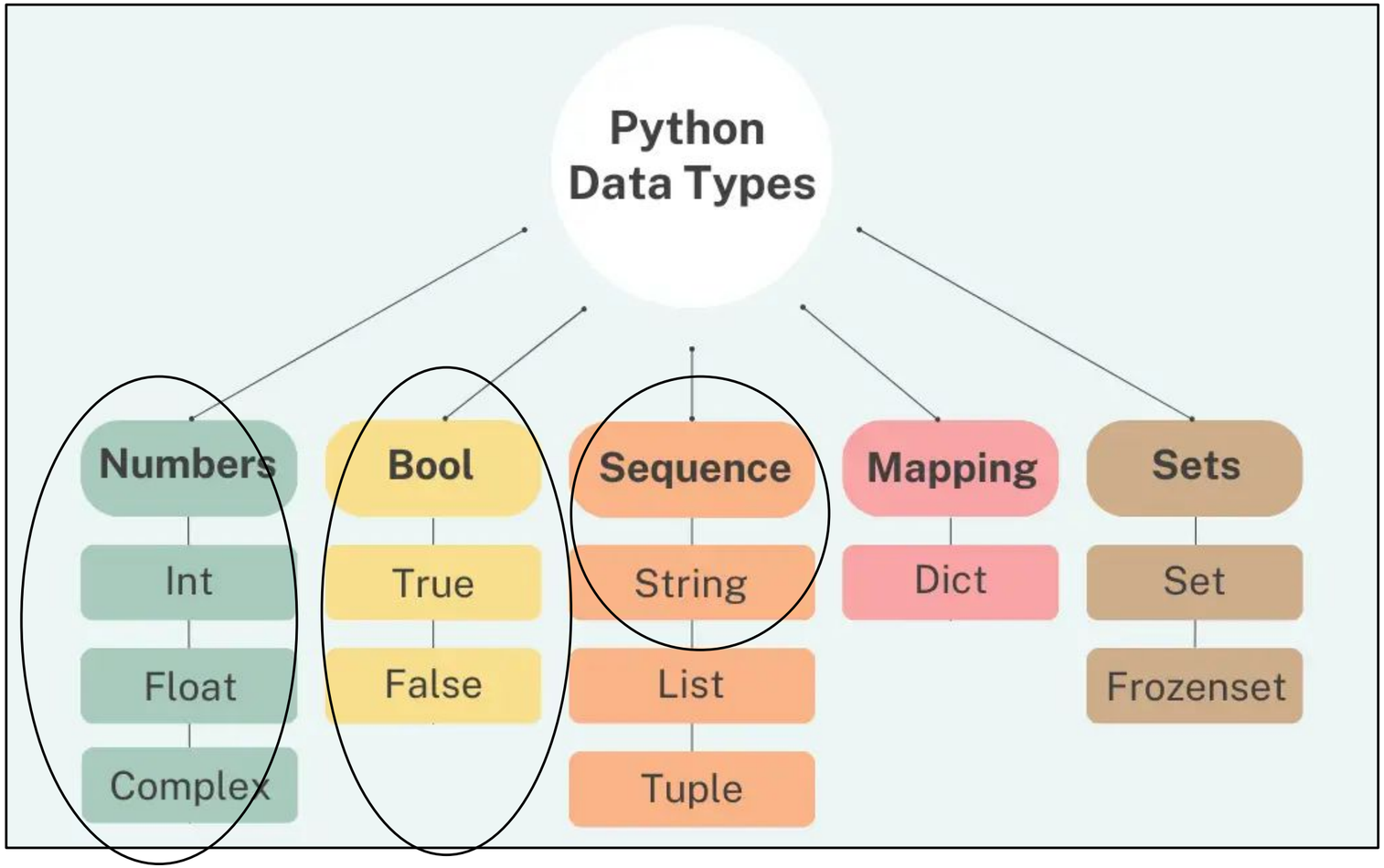
Exploring Types of Data :

Similarly, in programming, data is categorized into different types to keep it organized and manageable. Data types (like **integers, strings, floats, and booleans**) help manage various kinds of data in your programs.



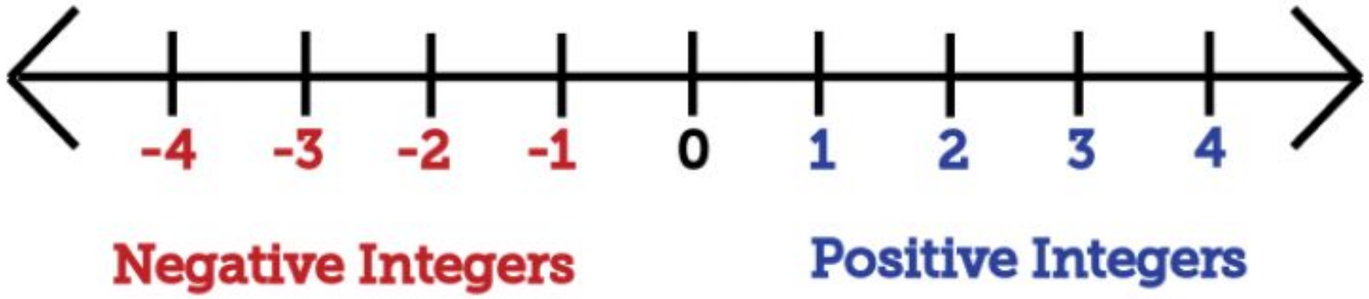
Data Types in Python :

Newton School of Technology



Integers in Python :

- Integers are numbers without any fractional part.
- They can be 0, positive or negative.
- There is no limit to how long an integer can be in Python.



Integers in Python :

```
number = 20
print(number)
```

```
Output
20
```

Code Snippet

Floating Point Numbers :

Newton School of Technology

- **Floating point numbers (float) are real numbers with floating point representation, they are specified by a decimal point.**
- **Floating point numbers are of float type.**
- **It is written as a number with a decimal point.**

Floating Point Numbers :

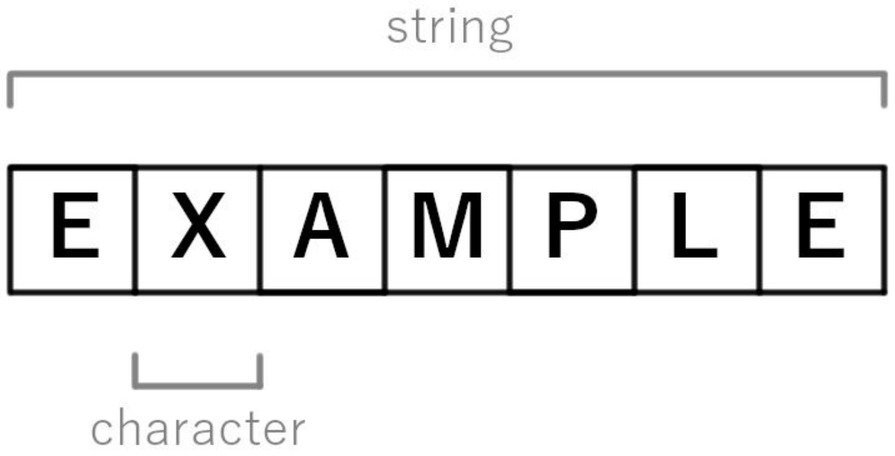
Newton School of Technology

```
pi = 3.14159
print(pi)
```

```
Output
3.14159
```

Text Type in Python – String :

- A string can be defined as a sequence of characters.
- We use single and double quotation marks to represent strings
- Triple quotation marks are used for multi-line strings.



Text Type in Python – String :

Newton School of Technology

python Copy code

```python
# Creating a string variable
my_string = "Hello, World!"

# Printing the string
print(my_string)
```

Output:

Copy code

```
Hello, World!
```

Code Snippet

Boolean Type :

- **Boolean is a data type that has one of two possible values, True or False.**
- **They are mostly used in creating the control flow of a program using conditional statements.**

Boolean Type :

```
a = True
b = False

#printing boolean value
print(a)
print(b)
```

```
Output
True
False
```

Re Assigning Variables :

Variable can be **assigned a new value**.

This can even change the variable's type, due to Python's dynamic typing.

```
x = 5
x = "Sara"
print(x)
```

Output:

```
Sara
```

Newton School of Technology

The type() Function

The type() function :

- **The type() function is an inbuilt function in Python.**
- **It returns the class type of the argument passed to it.**

01

integer

integers are whole numbers that can be positive, negative, or zero

02

float

float represent real numbers and include those with decimal points.

03

boolean

booleans represent true or false values

04

strings

strings are sequences of characters used to represent text data.

Think of a Student Record :

Newton School of Technology

- **Name** needs to store text → **str**
- **Age** is always a whole number → **int**
- **Grade** can be decimal (like 89.5) → **float**
- **Pass/Fail** is either yes or no → **bool**

The type() function – Student Record :

Newton School of Technology

```
name = "Richa"
age = 21
grade = 85.7
passed = True

print("Student record is :")
print(type(name))
print(type(age))
print(type(grade))
print(type(passed))
```

Output

```
Student record is :
<class 'str'>
<class 'int'>
<class 'float'>
<class 'bool'>
```

Code Snippet

Newton School of Technology

Names


JA
AR 4
JAR 1
JAR 3


PE
SIL
CUMIN
TURMERIC

Confusing Names – Confuse Coders :

Newton School of Technology

```
x = "Richa"
y = 21
z = 85.7
p = True

print("Student record is :")
print(type(x))
print(type(y))
print(type(z))
print(type(p))
```

Code Snippet

Newton School of Technology

Typecasting

Newton School of Technology

Typecasting :

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

Example: It is like you **freeze water into ice** before making a mango shake,

in programming, **you change a variable's type** to use it in a different way.

- **Water = Liquid data type**

- **Ice = Solid Data type**

- **Mango Shake**

Typecasting from string to integer :

Newton School of Technology

```
str_1 = "111"
str_2 = "222"
num_1 = int(str_1)
num_2 = int(str_2)

print(type(str_1))
print(type(str_2))
print(type(num_1))
print(type(num_2))
```

Output

```
<class 'str'>
<class 'str'>
<class 'int'>
<class 'int'>
```

Code Snippet

Typecasting :

- Float to Integer
- Integer to Float
- Integer to String
- Float to String
- String to String
- Boolean to Integer
- Integer to Boolean

Typecasting :

```
print(int(58.4))#float to int
print(float(17))#integer to float
print(str(276)) #integer to string
print(str(355.7))#float to string
print(str("abc"))# string to string
```

Output

```
58
17.0
276
355.7
abc
```

Invalid Typecasting from string to integer :

```
str_1 = "abc"
num_1 = int(str_1)
print(type(str_1))
print(type(num_1))
```

OR

```
print(int("abc"))
```

Output:

ValueError : You're trying to convert non-numeric strings ("abc") into integer using int().

bool() Function :

```
print(bool(1))    # map 1 to True
print(bool(0))    # map 0 to 1
print(int(True))  # map True value to 1
print(int(False)) #map False value to 0
```

```
True
False
1
0
```

Basic Typecasting Functions :

Newton School of Technology

Function	Converts to	Example
int()	integer	int(“10”) → 10
float()	float	float(“10”) → 10.0
str()	string	str(125) → “125”
bool()	boolean	bool(0) → False

Newton School of Technology

print() with Multiple Values

print() with Multiple Values :

Newton School of Technology

- **The print() function in Python can display multiple values by separating them with commas (,).**
- **Python will automatically add spaces between them.**

```
name    = "Richa"
age     = 21
grade   = 85.7
passed = True
print(name,age,grade,passed)
```

Output

```
Richa 21 85.7 True
```

Code Snippet

Newton School of Technology

print() – Understanding end and sep Parameters

print() – Understanding sep Parameters :

Newton School of Technology

- **sep** stands for **separator.**
- It's an **optional keyword parameter** used in the print() function
- It defines the **separator value between multiple values** passed to print().
- **By default, sep is a single space (" "),** but you **can change** it to any string.

Syntax :

```
print(value1, value2, ..., sep="separator")
```

https://my.newtonschool.co/playground/code/1hlhq3mxa7zk

print() – Understanding sep Parameters :

Newton School of Technology

	Code statement	Output
Default Separator: A single space (" ")	`print("Newton","School")`	`Newton School`

OR

Explicitly define Default Separator: A single space (" ")	`print("Newton","School",sep=" ")`	`Newton School`
Empty String: "" (no separation)	`print("Newton","School",sep="")`	`NewtonSchool`

Code Snippet

print() – Understanding sep Parameters :

Newton School of Technology

	Code statement	Output
Custom Strings: (e.g., ",", "-", "…", etc.)	`print("Hour","Minute",sep=":")`	`Hour:Minute`
Slash /	`print("Sum of values","No of values",sep="/")`	`Sum of values/No. of values`
Special Characters: ("\n")	`print("Newton","School",sep="\n")`	`Newton` `School`

Use of \n :

- The **\n is called a newline character** in Python.
- It tells the program **to move to the next line** when printing text.

Expected Output

```
Newton
School
```

```
print("Newton")
print("School")
```

Method -1

```
print("Newton\nSchool")
```

Method -2

print() – Understanding end Parameters :

- end is **an optional keyword parameter** in the print() function.
- It defines **what is printed at the end of the print statement.**
- **By default, end="\n"** → which means Python adds a newline after each print().

Syntax :

```
print(value1, value2, ..., end='ending_text')
```

https://my.newtonschool.co/playground/code/xekq4jcnwmz4

Newton School of Technology

print() – Understanding end Parameters :

	Code statement	Output
Default (newline)	`print("hello")` `print("World!")`	`hello` `World!`
	OR	
Explicitly define Default end: newline	`print("hello",end="\n")` `print("World!")`	`hello` `World!`
Same line output	`print("hello",end=" ")` `print("World!")`	`hello World!`

Code Snippet

print() – Understanding end Parameters :

Newton School of Technology

	Code statement	Output
Dot separator	`print("hello",end=".")` `print("World!")`	`hello.World!`
Comma	`print("hello",end=",")` `print("World!")`	`hello,World!`
No space	`print("hello",end="")` `print("World!")`	`helloWorld!`

Question - 1 : Print Variables

Newton School of Technology

Write a Python program that assigns values to variables using correct naming conventions:

- **Create a variable named first_Name with the value "Harsh"**
- **Create a variable named last_Name with the value "Gupta"**
- **Create a variable named _age with the value 25**

Print each variable on a separate line using the print() function.

Example : You do not need to take any input.

Output :

```
Harsh
Gupta
25
```

Question - 2 : What's the type

Newton School of Technology

Write a Python program that creates the following variables:

- **A string variable first_Name with the value "Harsh"**
- **A string variable last_Name with the value "Gupta"**
- **An integer variable _age with the value 25**
- **A float variable height with the value 5.9**

Your task is to print the data type of each variable using the type() function. Each output should be printed on a separate line.

Example : You do not need to take any input.

Output :

```
<class 'str'>
<class 'str'>
<class 'int'>
<class 'float'>
```

Question - 3 : Print in Single Line

Newton School of Technology

Write a Python program that creates the following variables:

- **A string variable first_Name with the value "Harsh"**
- **A string variable last_Name with the value "Gupta"**
- **An integer variable _age with the value 25**
- **A float variable height with the value 5.9**

Your task is to print the data type of each variable using the type() function. Each output should be printed on a separate line.

Example : You do not need to take any input.

Output :

```
Harsh Gupta 25
```

References :

Book_1 : Intro to Python by Paul Deitel		
S.No.	Topic Name	Page No.
1.	variables, types of variables, type()	50 - 51
2.	print function	56 - 57

Quiz Time!

Newton School of Technology

Please fill the feedback!

Thank You!