Chapter 2 - Variables and Datatypes.

A variable is the name given to a memory location in a Program For example.

a = 30 → variables = Container to Store a v
b = "Kaif" → keywords = Reserved words in Pytho
c = 71.22 → Identifiers = class/function/variab
name.

Data Types.

Primarly there are following data types in Pytho

1- Integers
2- Floating Point numbers
3- strings
4- Booleans
5- None.

Python is a Fantastic language that automatically indentifies the types of data For us.

a = 71 ⇒ Identifies ~~as~~ as class<int>

b=88.44 ⇒ Identifies b as Class <Float>

name="kaif" ⇒ Identifies C as Class <str>

Rules for defining a variable name → Also applies to other Identifiers.

→ A variable name can contain alphabets, digits and underscores.

→ A variable name can only start with an alphabet and underscore.

→ A variable name cant start with a digit.

→ No while space is allowed to be used inside a variable name.

Example of a few variable names are :-

kaif, one8, seven_, _seven etc.

Operators in Python

Following are some Common operators in Python

2) Arithmetic operators ⇒ +, −, *, / etc.
2) Assignmetic operators ⇒ =, +=, −= etc.
3) Comparison operators ⇒ ==, >, >=, <, != etc.
4) Logical operators ⇒ and, or, not.

type () Function and typecasting
type Function is used to find the data type of a given variable in Python.

```
a = 31
type(a) => class <int>

b = "31"
type(b) => Class <str>
```

A number can be converted into a string and vice verse (If Possible)
There are many functions to convert one data type in another.

str(31) ⇒ "31" → Integer to string convers
int("32") ⇒ 32 → string to integer con
Float(32) ⇒ 32.0 ⇒ Integer to Float co

DATE / /

... and so on

Here "31" is a string literal and 31 a numeric literal.

input () Function

This function allows the user to take input from the keywords as a string.

```
a = input ("Enter name")
```

It a is "kaif" the user entered

Kaif

It is important to note that the output of input is always a string. (even if the number is entered)

It a is "34" user entered 34.

* * *

on

rors -

versions