

Demo 1

Demo 2

```
>>> Ab123 = 6
>>> _b123 = 7
>>> 123ABC = 8
SyntaxError: invalid syntax
```

verbal explanation: A digit is not allowed as a first character of an identifier

```
>>> $eer = 9
SyntaxError: invalid syntax
```

caption : Must begin with a letter or _

a symbol, except for underscore cannot be used as a part of the name.

```
>>> this_is_an_identifier_123 = 234
>>> T_ddf__33__ = 22
```

- **caption :** May contain letters, digits, and underscores

```
>>> break = 44
SyntaxError: invalid syntax
>>> if = 55
SyntaxError: invalid syntax
... |
```

- **caption :** Should not use keywords.

Why syntax error? Because break and if are two keywords, we should not use keywords. What are keywords?

Demo 3:

```
>>> break = 44
SyntaxError: invalid syntax
>>> if = 55
SyntaxError: invalid syntax
>>> Break = 99
>>> If = 88
```

caption beside Python is case sensitive

Change the first letter of `break` to capital B and the first letter of `if` to capital I, and both are no longer keywords, and thus they can now be used as valid identifiers.

```
>>> LengthOfRope = 11
>>> lengthofrope = 33
>>> print(LengthOfRope)
11
```

Two `lengthofRopes` are different

An identifier can be of any length, short or long, it is not important whether the name is short or long, for as long as it is meaningful and with reasonable length

- **Caption: Can be of any length ,**
-
- **Caption: Names starting with `_` have special meaning.**

Script:

Names starting with `_` have special meaning. You are not encouraged to start your identifier with `_` at this stage.

Demo 4

After reading through the code quickly, we know that our expected average of the Math scores is 80, and that of the English scores is 60.

Let's run it.

Caption: ► Can we interpret and run this program? ► Yes.

► Is the result correct? ► No

What went wrong?

► **Caption: Be careful! Python is case sensitive!**

The variable “total”, which is the total of Math scores is different from variable “Total”, which is total of English scores. If we use lower-case total to calculate the average of English scores, of course the output is the average Math score.

We change the lowercase t to capital T, run it again. The result is what we expect.

Unlike syntax error, this kind of error is difficult to spot. It is called the Logic error . **Caption: Logic error**

Caption: A program, that can run doesn't mean that it is correct. Do remember, you need to design some test cases and let your program pass all the test cases to make sure that it is error free before you submit your assignments.

Demo 5

```
>>> x = 9 Let's assign value 9 to variable x
```

```
>>> type(x) Then check the data type of the variable x
```

```
<class 'int'> it's int
```

```
>>> x = 7.8 Let's assign another value 7.8 to the same variable x
```

```
>>> type(x) Then check the type
```

```
<class 'float'> it changes to float
```

```
>>> x = "Welcome" assign another value
```

```
>>> type(x) check type
<class 'str'> it changes to str
```

Caption: Python does not have variable declaration like Java or C to announce or create a variable.

Caption: A variable is created by just assigning a value to it and the type of the value defines the type of the variable.

Caption: If another value is re-assigned to the variable, its type can change.

We can also check the data type of a literal

```
>>> type(8.9)
<class 'float'>
```

Besides integers and floats, we find a common data type, which to us, is a word or a sentence. In programming, it is called a string.

Demo 6--- code typing, no explanation, fast forward, until my voice starts again

`hori_dist` we have two inputs, `hori_dist` and `vert_dist`

`vert_dist`

`travel_dist = hori_dist + vert_dist` we have the formula to calculate the total travel distance

`print(travel_dist)` then display the output

We have not learnt how to read inputs from user. Then just hardcode the values first

```
hori_dist = 4
```

```
vert_dist = 3
```

```
travel_dist = hori_dist + vert_dist
```

```
print (travel_dist)
```

Let's run it.

A single number output 7. Not so user-friendly. So, let's add more information to the output.

```
print("distance from A to B is ", travel_dist)
```

run it again. distance from A to B is 7. It is more meaningful.

In order to make the program flexible, we should be able to read in the inputs from user's keyboard. The built-in function `input()` is the easiest solution. The details will be given in next Python lesson, now you just follow my demo, type in the following two statements to replace the hardcode statements.

```
horizon_dist = (int)(input("Read horizonDist"))
```

```
vertical_dist = (int)(input("Read vertDist"))
```

Let's run it again.

```
Read horizonDist 4
```

```
Read vertDist 3
```

```
distance from A to B is 7
```

For another user, the location is different

Without changing the code, we can run the same program again with different inputs,

```
Read horizonDist 8
```

```
Read vertDist 5
```

```
distance from A to B is 13
```