Rules for Declaring a variable

- Although we can use any name for declaring a variable but we must follow certain rules, so that it is easy to understand by us and others too

- The variable names can be taken as:

```
a= 10
b = 12.5
c= ‘ John ‘
```

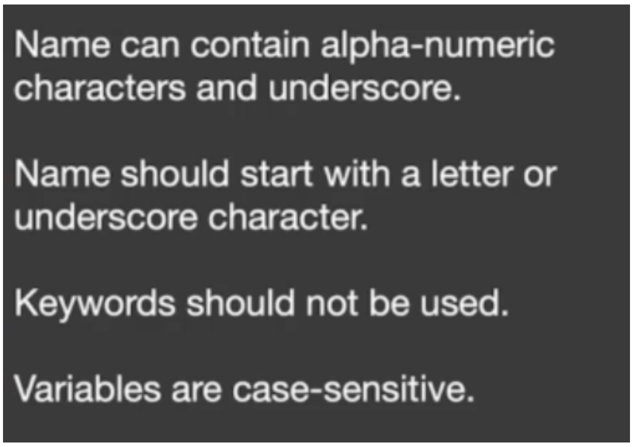
- Even though the above declaration is correct but we cannot extract the exact meaning of the variables.

- To make the above variables more meaningful we can declare them as   :

```
roll_no = 10
price = 12.5
cust_name= ‘ John ‘
```

- Now the above variables are more descriptive and understandable .

- The rules for declaring a variables is as follows



Name can contain alpha-numeric characters and underscore.

Name should start with a letter or underscore character.

Keywords should not be used.

Variables are case-sensitive.

- The **first rule** says that we can mix alphabet and numbers while declaring the variables, we can even use an underscore.

- However we cannot use any special symbol like $, &,@,#,- etc…
- Example :

x1 = 10 ✓

cust_name= ‘ John ✓

address1 ✓

address#1 ✗

address-1 ✗

- The **second rule** says that the variable must start with a letter or underscore character only.
- Although we can use alphabet and numbers in a variable but numbers cannot be used at the beginning of the variable name.
- Example :

x2= 10 ✓

_x = 10 ✓

1x = 10 ✗

- The **third rule** states we cannot use keywords to declare a variable.
- In python program we use words, numbers or symbols.
- here the words can be categorise into two, that is identifiers and keywords .

Identifiers : these are the words given by the programmer, It is used for identifying

something that is define exclusively by the programmer .

price = 12.5

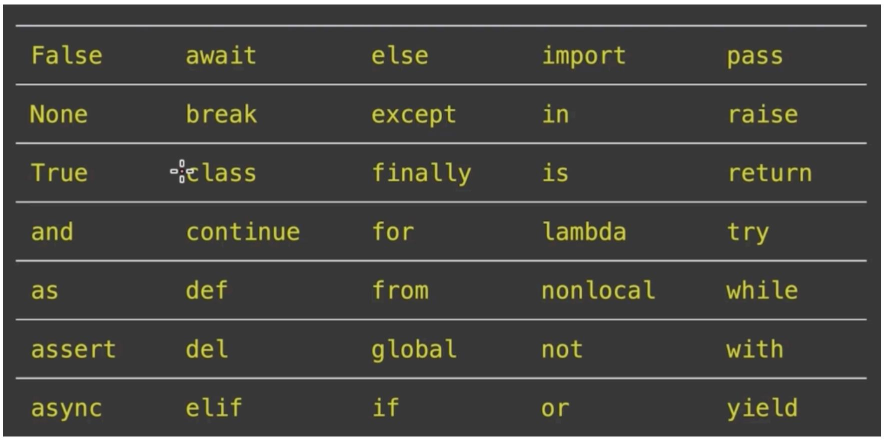
- Variable name, function name , class name and even module name are all identifiers.

Keywords : the words which are predefined in the language are called keywords or reserved words.

while = 10 ✗

pass = 20 ✗

- The list of the keywords are given below .
- These word cannot be used while declaring the variable.



False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

- The last rule states that variables are case sensitive,

a = 10

A = 10

- The above variables are not same , ‘A’ is not same as ‘a’.