Data type in Python...

★ Objects and Mutability.

Everything in python is an object.

When we say anything as an object, then it must have some properties.

- So every Object will have a unique identity.
- Every object will also have a unique type
- Each object will have a value

★ Mutable and Immutable

Mutable → that is changeable.
Immutable → that is not changeable.

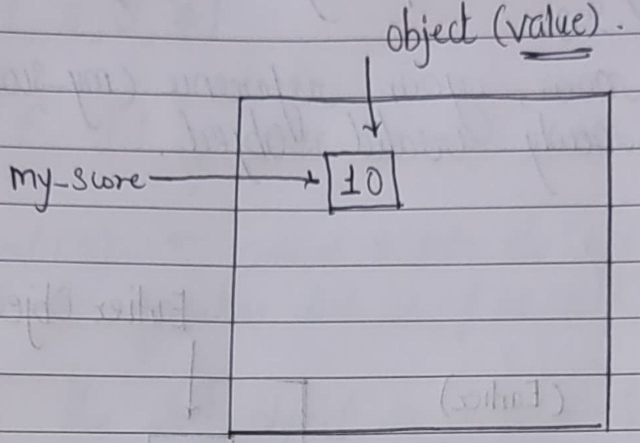
Mutability is checked with <u>ID</u> and <u>not</u> the <u>Value</u>.

If identity is same the value didn't got changed.
If the identity is different, the value got changed.

For Example:

```
my_score = 10 ← a var holding value 10.
print (f"Shaill your Score is: {my_score}");
```

Inside Memory...



Now, if I change the value of my_score, What you think will happen?
will this my_score = 10 will get overwrite?
Let's experiment with it...

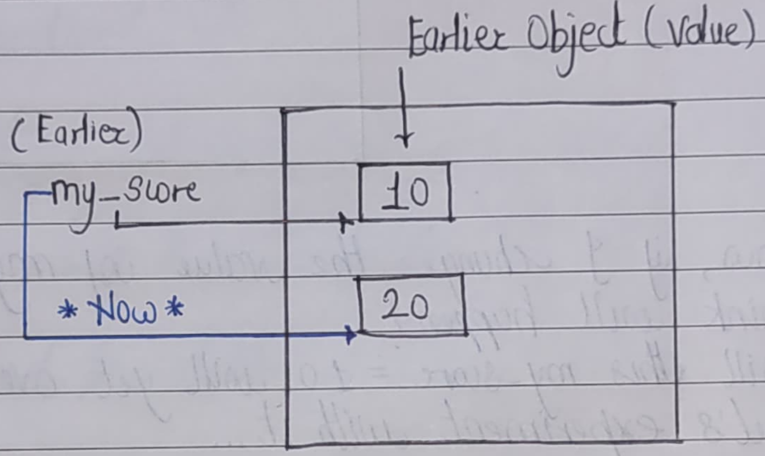
```
my_score = 20   ← same var (reference) holding 20
print (f"Shaill your second initial score is: {my_score}");
```

Let's See output:

```
Shaill your score is: 10.
Shaill your second initial score is: 20.
```

Now you might think that, value of my_score just got overwrite!
But wait that's not really the case.
In Python Numbers are "Inmutable" i.e. they cannot be changed!
So how my_score got changed here?

What happened actually behind the scene is, inside memory a completely new object is created with value 20.
And now, your reference (my-score) points to that newly created object.



If you want a proof that a brand new object is created, then use id() which takes an object as it's parameter and returns the unique id alloted for that specific object.

So you can compare the unique Id (memory address) of both the objects.

If Id's are different, it means a completely new object is created.
If It's the same, it updated the existing object (Mutable)...

Now, lets take an example of Mutating object.

```
ls = [1, 2, 3, 4]

print(F "Id of this list is": {id(ls)}");

ls.append(5); ← added 5 into the list...
print(F "Id of this list is : {id(ls)}");
```

Now as the list object is a mutable entity, which updates existing object, this list will get updated with 5, and when we print it's Id, It will print the same Id.