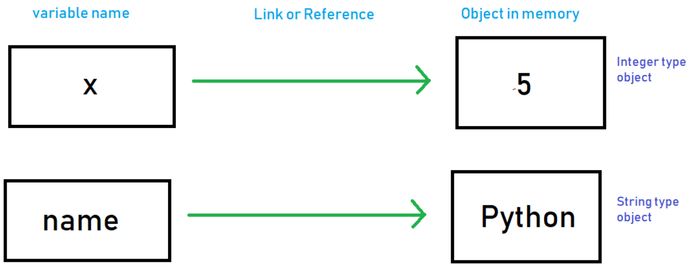
Python Objects

- Everything in Python is object.
- A uniquie object ID is assigned to an object when it is created.
- A variable is a reference to an object.
- More than one variable can refer to the same object.

- In Python, object is either mutable or immutable.
- Object can be changed only if it is mutable. (Immutable object cannot be changed.)

- Built-in types objects (e.g., int, float, bool, str, tuple and unicode) are immutable.
- Built-in types objects (e.g.,list, set and dict) are mutable.



(image from : https://www.pythoneasy.com/python-programming-tutorial/variables-datatypes)

Lists

let's create one list variable

```
In [1]: beatles = ["John", "Paul", "George", "Ringo"]
```

```
In [2]: print('beatles', beatles)
        print(id(beatles))
```

```
beatles ['John', 'Paul', 'George', 'Ringo']
2336871918080
```

let's create another list variable from this list

```
In [3]: beatles2 = beatles
```

now, we have two list variables

```
In [4]: print('beatles', beatles)
        print('beatles2', beatles2)
```

```
beatles ['John', 'Paul', 'George', 'Ringo']
beatles2 ['John', 'Paul', 'George', 'Ringo']
```

let's change value in one of the lists

```
In [5]: beatles[2] = "Alonzo"
```

ummm .. how do the two list varibles look now ?

```
In [6]: print('beatles', beatles)
        print('beatles2', beatles2)
```

```
beatles ['John', 'Paul', 'Alonzo', 'Ringo']
beatles2 ['John', 'Paul', 'Alonzo', 'Ringo']
```

why ?

```
In [7]: print(id(beatles))
        print(id(beatles2))
```

```
2336871918080
2336871918080
```

what about creating a list with 'copy' ?

```
In [8]: beatles3 = beatles[:]
```

```
In [9]: print('beatles', beatles)
        print('beatles3', beatles3)
```

```
beatles ['John', 'Paul', 'Alonzo', 'Ringo']
beatles3 ['John', 'Paul', 'Alonzo', 'Ringo']
```

now, assign a value to one list

```
In [10]: beatles[2] = "Pete"
```

let's see how the two lists look now

```
In [12]: print('beatles', beatles)
         print('beatles2', beatles2)
         print('beatles3', beatles3)
```

```
beatles ['John', 'Paul', 'Pete', 'Ringo']
beatles2 ['John', 'Paul', 'Pete', 'Ringo']
beatles3 ['John', 'Paul', 'Alonzo', 'Ringo']
```

why ?

```
In [13]: print(id(beatles))
         print(id(beatles2))
         print(id(beatles3))
```

```
2336871918080
2336871918080
2336871953088
```

are Lists mutable ?

Strings

let's create one String variable

```
In [14]: str1 = "potato"
```

let's create another String varible from this String

```
In [15]: str2 = str1
```

let's see how the two String variables look

```
In [16]: print(str1)
         print(str2)
```

```
potato
potato
```

```
In [17]: print(id(str1))
         print(id(str2))
```

```
2336872241648
2336872241648
```

now, let's change the value of one of them

```
In [18]: str1 += " corner"
```

let's see how the two Strings look

```
In [20]: print(str1)
         print(str2)
```

```
potato corner
potato
```

why ?

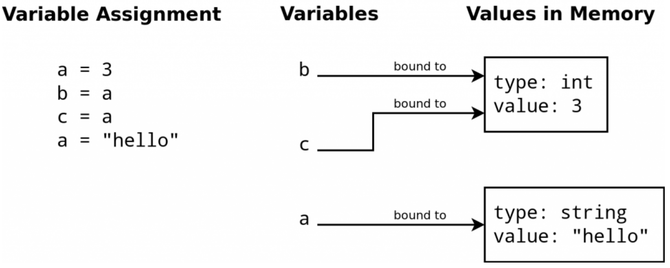
```
In [21]: print(id(str1))
         print(id(str2))
```

```
2336872252784
2336872241648
```

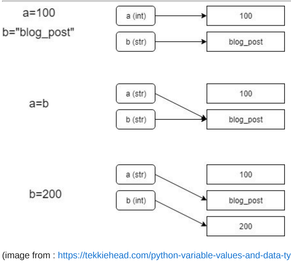
are Strings mutable ?

---

now, can you explain object, variable, reference and mutability in Python ?



(image from : http://swcarpentry.github.io/training-course/2013/05/python-variables-a-diagram-that-is-not-a-concept-map/)



(image from : https://tekkiehead.com/python-variable-values-and-data-types/)