

# DYNAMIC TYPING vs STATIC TYPING

© 2018 MemoryToAcademy



Some languages (Java, C++, Swift) are **statically** typed

```
String myVar = "hello";
```

data  
type

variable  
name

value



```
myVar = 10;
```

**Does not work!**

myVar has been **declared** as a String, and cannot be assigned the integer value **10** later.

```
myVar = "abc";
```

**This is OK!**

"abc" is a String – so compatible type and assignment works.



Python, in contrast, is **dynamically** typed.

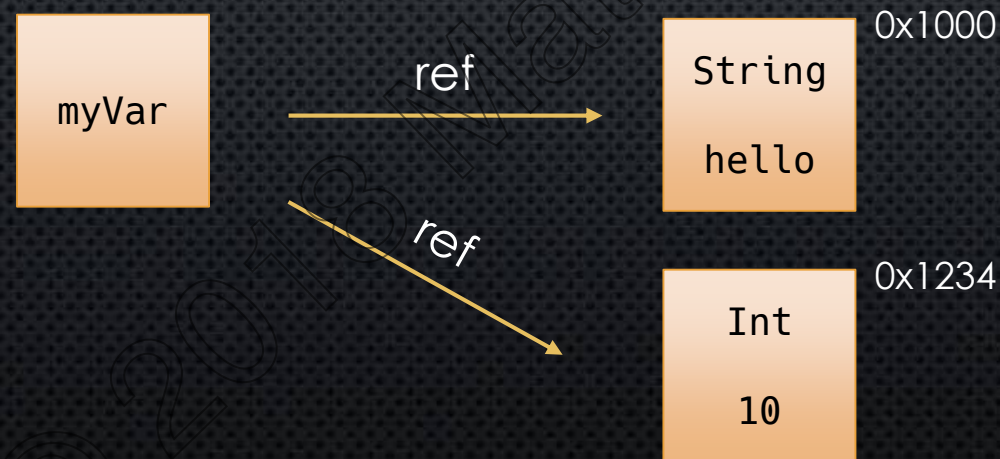
```
my_var = 'hello';
```

The variable `my_var` is purely a **reference** to a string object with value `hello`.

No type is “attached” to `my_var`.

```
my_var = 10;
```

The variable `my_var` is now pointing to an integer object with value 10.





We can use the built-in `type( )` function to determine the type of the object `currently referenced` by a variable.

Remember: variables in Python do not have an inherent static type.

Instead, when we call `type(my_var)`

Python looks up the object `my_var` is `referencing` (pointing to), and returns the `type of the object` at that memory location.