

A Quick Tour of Python

Julien Roland

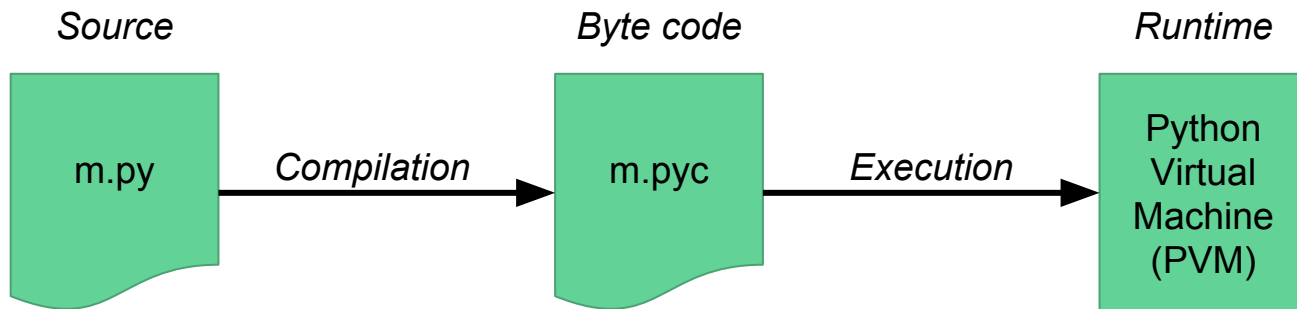
Outline

1. Python
2. Language Syntax
3. Types and Objects
4. Demo : Python REPL, Scripts, Notebooks

Python

Python

- Created by Guido Van Rossum in the late 1980s
- A multi-paradigm programming language (OOP, Procedural,...)
- Interpreted (Byte Compiled)



- Memory management by the interpreter

Language Syntax*

** Based on A Whirlwind Tour of Python by Jake VanderPlas, O'Reilly, 2016*

Language Syntax

```
midpoint = 5
```

```
# make two empty lists
```

```
lower = []; upper = []
```

```
# split the numbers into lower and upper
```

```
for i in range(10):
```

```
    if (i < midpoint):
```

```
        lower.append(i)
```

```
    else:
```

```
        upper.append(i)
```

```
print("lower:", lower)
```

```
print("upper:", upper)
```

Comments Are Marked by

```
midpoint = 5
```

```
# make two empty lists
```

```
lower = []; upper = []
```

```
# split the numbers into lower and upper
```

```
for i in range(10):  
    if (i < midpoint):  
        lower.append(i)  
    else:  
        upper.append(i)
```

```
print("lower:", lower)  
print("upper:", upper)
```

End-of-Line Terminates a Statement

```
midpoint = 5
```

```
# make two empty lists
```

```
lower = []; upper = []
```

```
# split the numbers into lower and upper
```

```
for i in range(10):
```

```
    if (i < midpoint):
```

```
        lower.append(i)
```

```
    else:
```

```
        upper.append(i)
```

```
print("lower:", lower)
```

```
print("upper:", upper)
```


Semicolon Can Optionally Terminate a Statement

```
midpoint = 5
```

```
# make two empty lists
```

```
lower = []; upper = []
```

```
# split the numbers into lower and upper
```

```
for i in range(10):
```

```
    if (i < midpoint):
```

```
        lower.append(i)
```

```
    else:
```

```
        upper.append(i)
```

```
print("lower:", lower)
```

```
print("upper:", upper)
```

Indentation: Whitespace Matters!

```
midpoint = 5
```

```
# make two empty lists
```

```
lower = []; upper = []
```

```
# split the numbers into lower and upper
```

```
for i in range(10):
```

```
    if (i < midpoint):  
        lower.append(i)  
    else:  
        upper.append(i)
```

Code blocks are
denoted by
indentation

```
print("lower:", lower)
```

```
print("upper:", upper)
```

Parentheses Are for Grouping or Calling

```
midpoint = 5
```

```
# make two empty lists
```

```
lower = []; upper = []
```

```
# split the numbers into lower and upper
```

```
for i in range(10):
```

```
    if (i < midpoint):  
        lower.append(i)
```

```
    else:  
        upper.append(i)
```

```
print("lower:", lower)
```

```
print("upper:", upper)
```

Types and Objects*

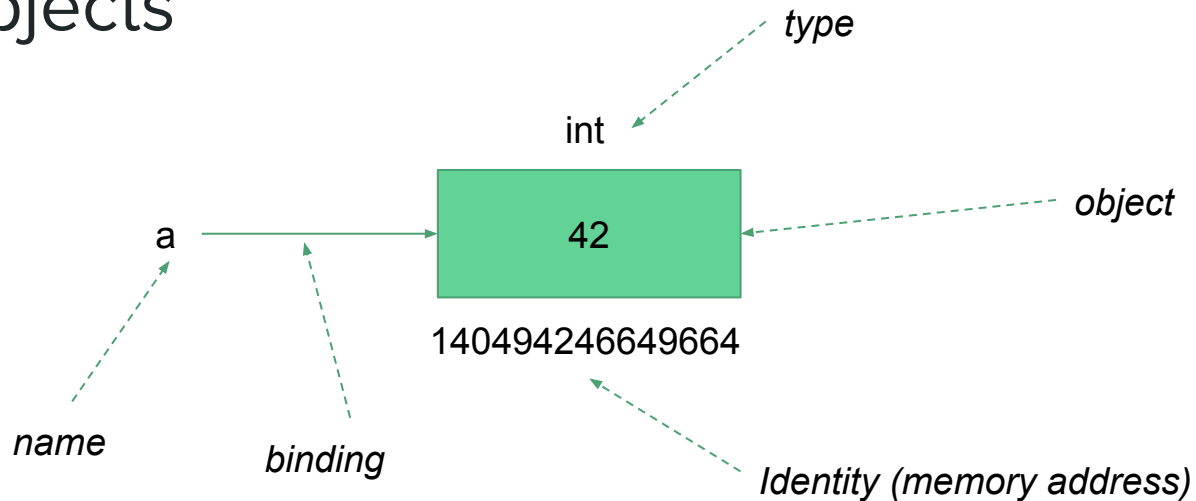
** Based on Python Essential Reference, by David Beazley, Addison-Wesley, 2009*

Types and Objects

`a = 42`

Types and Objects

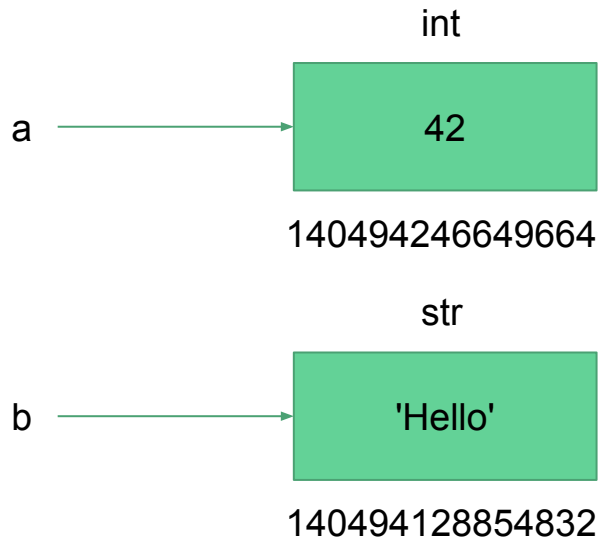
a = 42



Types and Objects

```
a = 42
```

```
b = 'Hello'
```

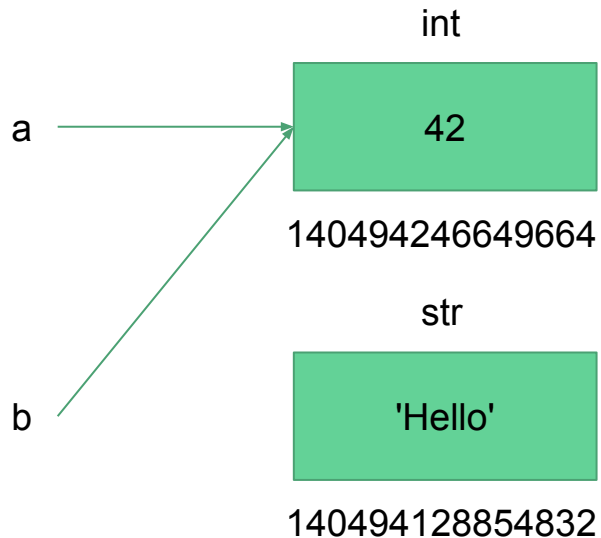


Types and Objects

```
a = 42
```

```
b = 'Hello'
```

```
b = a
```

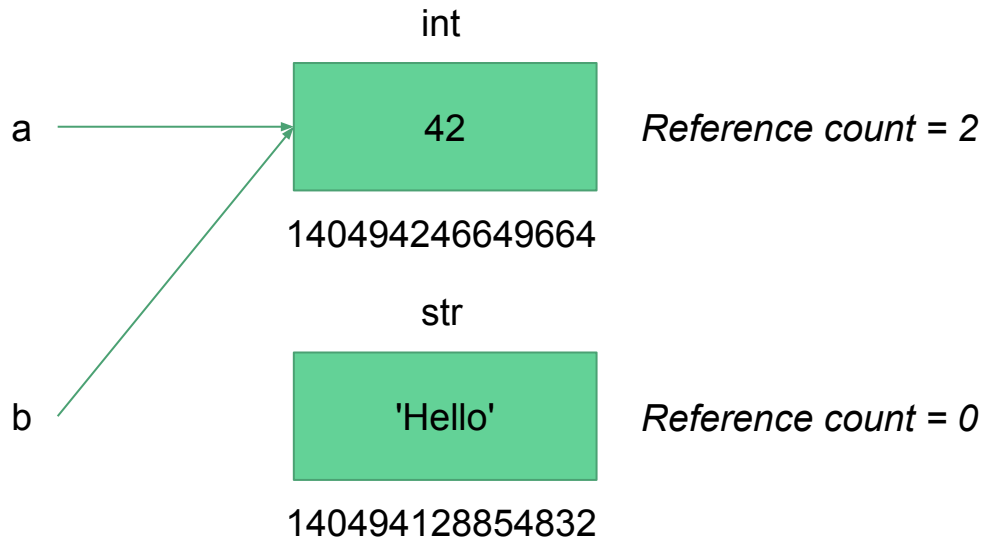


Types and Objects

```
a = 42
```

```
b = 'Hello'
```

```
b = a
```

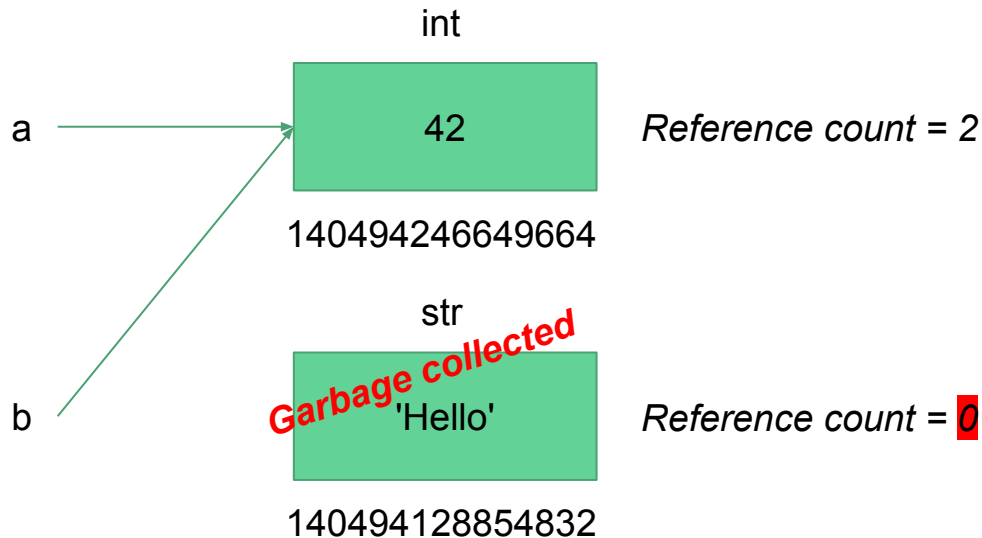


Types and Objects

```
a = 42
```

```
b = 'Hello'
```

```
b = a
```



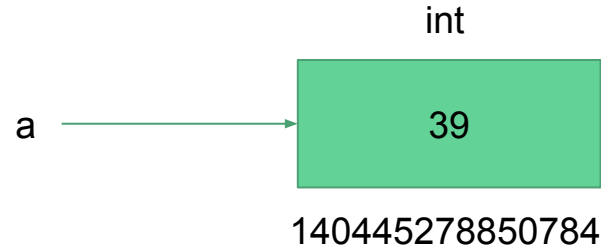
Terminology

- **Type** (the object's class) : describes object's internal representation and methods/operations it supports
- **Instance** : an object of a particular type
- **Mutable** object : its value can be modified
- **Immutable** object : its value cannot be modified
- **Container** (or collection) : an object that contains references to other objects

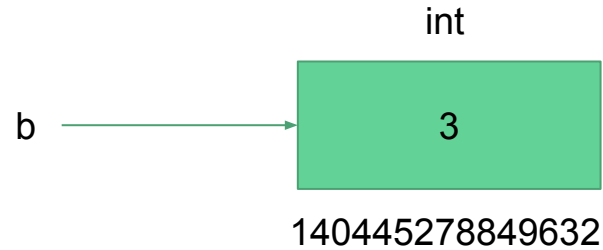
```
filename = 'key.txt'  
words = ['contains', 'references', 'to', 'other', 'objects']
```

Immutable Objects

a = 39



b = 3

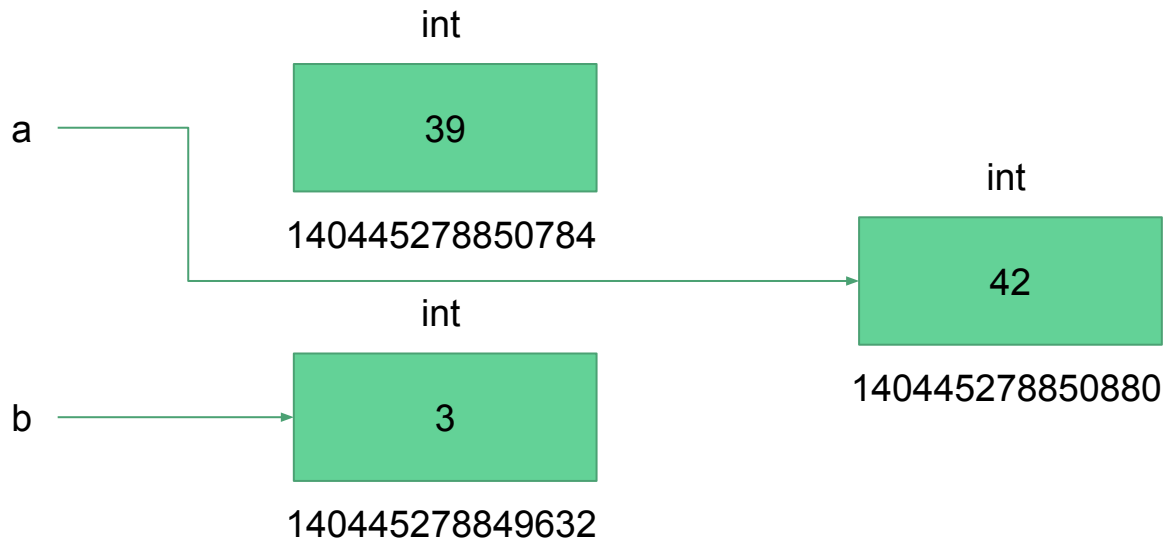


Immutable Objects

```
a = 39
```

```
b = 3
```

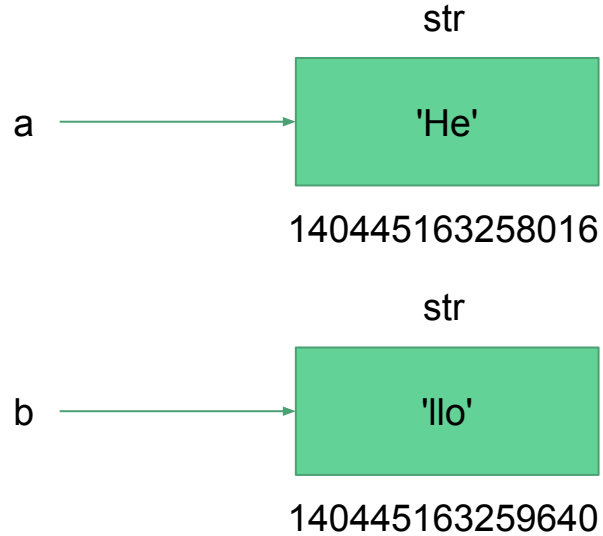
```
a = a + b
```



Immutable Objects

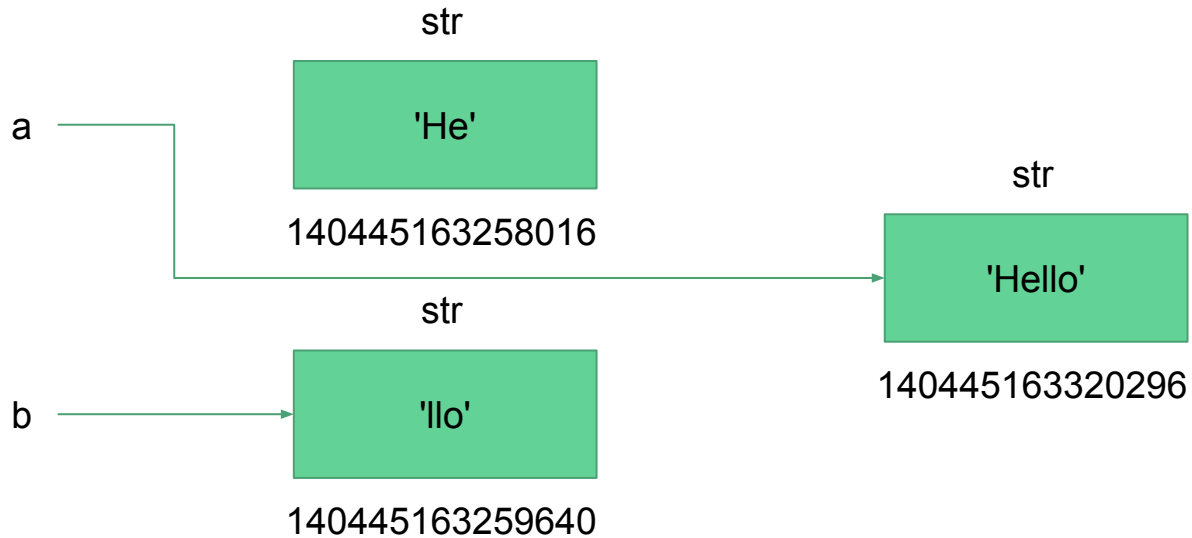
```
a = 'He'
```

```
b = 'llo'
```



Immutable Objects

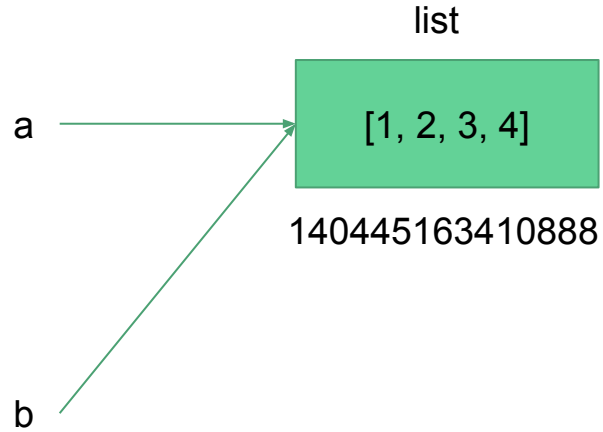
```
a = 'He'  
b = 'llo'  
a += b
```



Mutable Objects

a = [1, 2, 3, 4]

b = a

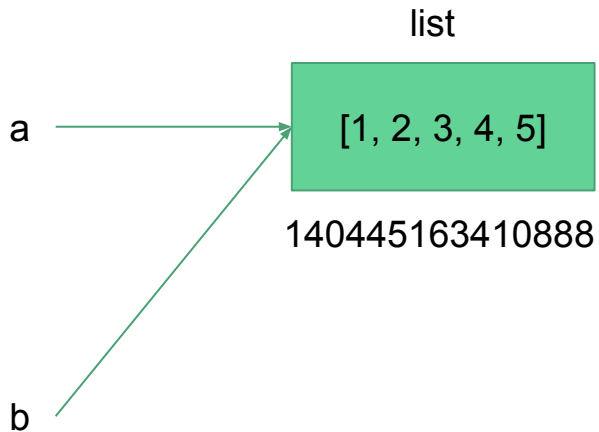


Mutable Objects

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

```
b = a
```

```
a.append(5)
```



Demo : REPL, Scripts, Notebooks
