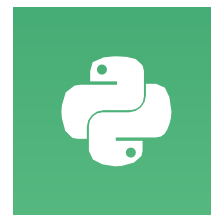


Python Strings and Data Structures

INTRODUCTION TO PYTHON



Reddy Prasad

Machine Learning Researcher at ClientoClarify.ai

Problem

- Data Science: many data points
- Height of entire family

```
height1 = 1.73
```

```
height2 = 1.68
```

```
height3 = 1.71
```

```
height4 = 1.89
```

- Inconvenient

Python Data Types

- Float - real numbers **e.g. : (10.5,-2.5)**
- Int – integer numbers **e.g.: (100,-255)**
- Str - string, text **e.g.: ('Python', "World")**
- Bool - True, False **e.g.: (True ,False)**
- Complex – Imaginary **e.g.: (2j+3, 3+2j)**
- Each variable represents single value

```
height = 1.73
```

```
tall = True
```

Strings

- String is Nothing But Collection Characters [A,B,C]
- Strings Can Create by using “Double quotes” and ‘Single quotes’
- `#` is Used For Comments
- ''' Single Quotes''' for Multiline Comment
- """ Double quotes """ for Multiline Comments

`S = “Hello”`

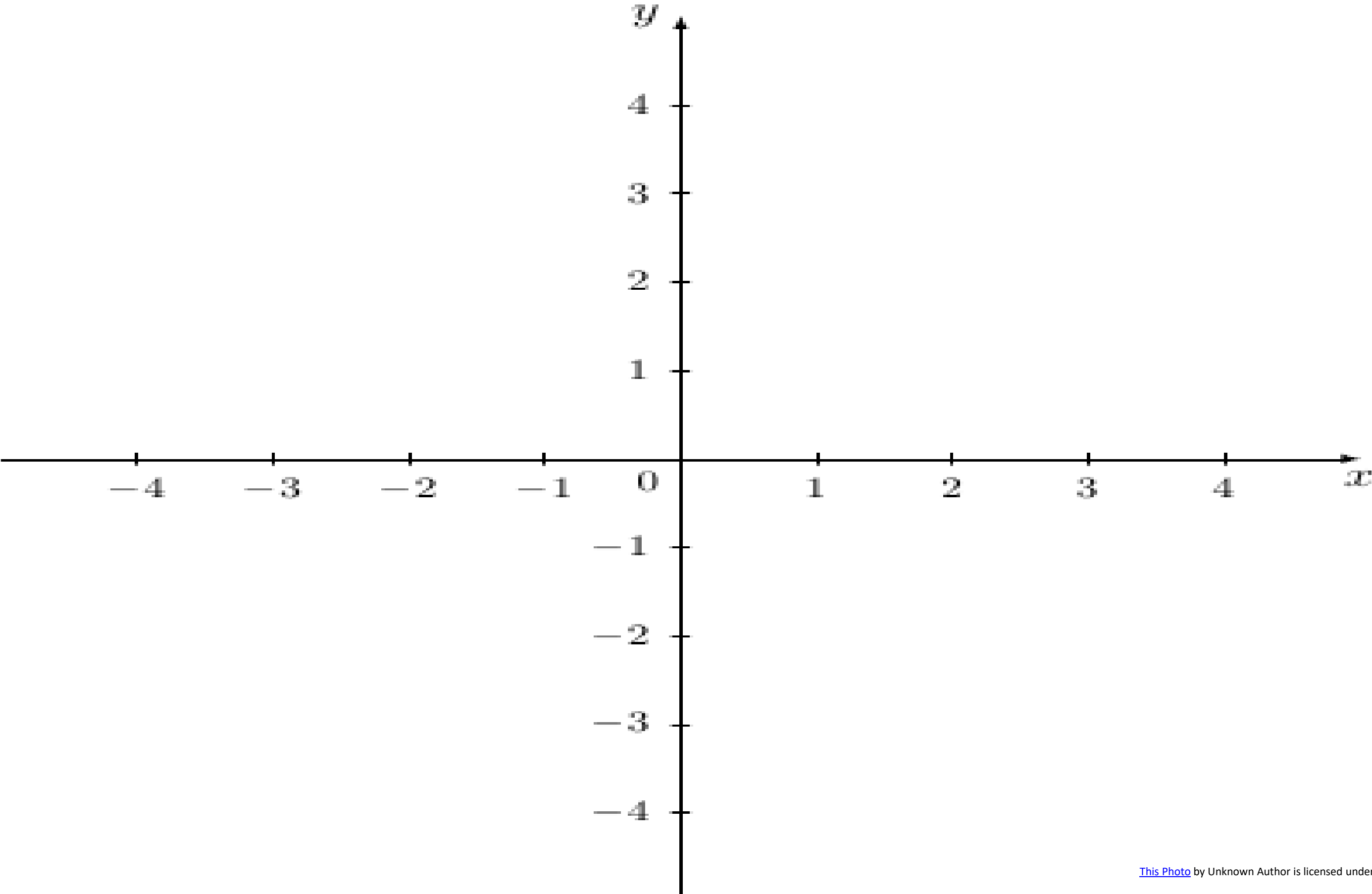
Or

`S_1 = ‘Hello’`

`S = “1025”`

Or

`S_1 = ‘10.25’`



Index

- Index is nothing but Position of particular Element
- Index is otherwise call it as Indexing to Select the Data.

S = "Clientoclarify.ai"

Negative Index -ve

-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
C	L	I	E	N	T	O	C	L	A	R	I	F	Y	.	A	I
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Positive Index +ve

Let's practice!

INTRODUCTION TO PYTHON

List

- A list is a sequenced collection of different objects such as integers, strings, Bool, Float, complex and other lists as well.
- The address of each element within a list is called an **index**.

Index	
0	Element 1
1	Element 2
2	Element 3
3	Element 4
4	Element 5
Element	

[Element 1 , Element 2 , Element 3 , Element 4 , Element 5]

Index 0 1 2 3 4

List

- A list is a sequenced collection of different objects such as integers, strings, Bool, Float, complex and other lists as well.
- The address of each element within a list is called an **index**.
- An index (Data Selection) is used to access and refer to Element/items within a list.

$L = [\text{"Michael Jackson"}, 10.1, 1982]$

-3	0	"Michael Jackson"
-2	1	10.1
-1	2	1982

$L[-3]: \text{"Michael Jackson"}$

$L[-2]: 10.1$

$L[-1]: 1982$

List

- A list is a sequenced collection of different objects such as integers, strings, Bool, Float, complex and other lists as well.
- The address of each element within a list is called an **index**.
- An index (Data Selection) is used to access and refer to Element/items within a list.
- List will allow us to Perform *index, Slice, Extended Slice* (Strid) and we assign a Element to it as well.

`L = ["Michael Jackson", 10.1, 1982, "MJ", 1]`

0	1	2	3	4
---	---	---	---	---

`L[3:5]`

List

- A list is a sequenced collection of different objects such as integers, strings, Bool, Float, complex and other lists as well.
- The address of each element within a list is called an **index**.
- An index (Data Selection) is used to access and refer to Element/items within a list.
- List will allow us to Perform *index, Slice, Extended Slice* (Strid) and we assign a Element to it as well.
- List is Mutable(Which we can Change at any time, add, modify,delete).
- List is having different Methods
- Index Based item assignment will be they.
- List Can Create by using Square Brackets [].

Python List

- `[a, b, c]`

```
[1.73, 1.68, 1.71, 1.89]
```

```
[1.73, 1.68, 1.71, 1.89]
```

```
fam = [1.73, 1.68, 1.71, 1.89]  
fam
```

```
[1.73, 1.68, 1.71, 1.89]
```

- Name a collection of values
- Contain any type
- List Can Create by is using Square Bracket []

Python List

- `[a, b, c]`

```
fam = ["liz", 1.73, "emma", 1.68, "mom", 1.71, "dad", 1.89]
```

```
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam2 = [{"liz", 1.73},  
        {"emma", 1.68},  
        {"mom", 1.71},  
        {"dad", 1.89}]
```

```
fam2
```

```
[['liz', 1.73], ['emma', 1.68], ['mom', 1.71], ['dad', 1.89]]
```

List Inside a List Is called
Nested list

List type

```
type(fam)
```

```
list
```

```
type(fam2)
```

```
list
```

- Specific functionality
- Specific behavior

Let's practice!

INTRODUCTION TO PYTHON

Subsetting Lists

INTRODUCTION TO PYTHON



Reddy Prasad
Machine Learning Researcher at ClientoClarify.ai

Subsetting lists

```
fam = ["liz", 1.73, "emma", 1.68, "mom", 1.71, "dad", 1.89]  
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam[3]
```

```
1.68
```

Subsetting lists

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam[6]
```

```
'dad'
```

```
fam[-1]
```

```
1.89
```

```
fam[7]
```

```
1.89
```

Subsetting lists

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam[6]
```

```
'dad'
```

```
fam[-1] # <-
```

```
1.89
```

```
fam[7] # <-
```

```
1.89
```

List slicing

```
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam[3:5]
```

```
[1.68, 'mom']
```

```
fam[1:4]
```

```
[1.73, 'emma', 1.68]
```

[**start** : **end**]

inclusive

exclusive

List slicing

```
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam[:4]
```

```
['liz', 1.73, 'emma', 1.68]
```

```
fam[5:]
```

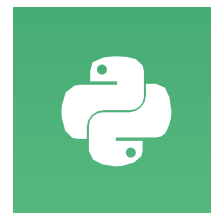
```
[1.71, 'dad', 1.89]
```


Let's practice!

INTRODUCTION TO PYTHON

Manipulating Lists

INTRODUCTION TO PYTHON



Reddy Prasad

Machine Learning Researcher at ClientoClarify.ai

List Manipulation

- Change list elements
- Add list elements
- Remove list elements

Changing list elements

```
fam = ["liz", 1.73, "emma", 1.68, "mom", 1.71, "dad", 1.89]  
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam[7] = 1.86  
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.86]
```

```
fam[0:2] = ["lisa", 1.74]  
fam
```

```
['lisa', 1.74, 'emma', 1.68, 'mom', 1.71, 'dad', 1.86]
```

Adding and removing elements

```
fam + ["me", 1.79]
```

```
['lisa', 1.74, 'emma', 1.68, 'mom', 1.71, 'dad', 1.86, 'me', 1.79]
```

```
fam_ext = fam + ["me", 1.79]
```

```
del(fam[2])
```

```
fam
```

```
['lisa', 1.74, 1.68, 'mom', 1.71, 'dad', 1.86]
```

Behind the scenes (I)

```
x = ["a", "b", "c"]
```



Behind the scenes (I)

```
x = ["a", "b", "c"]
```

```
y = x
```

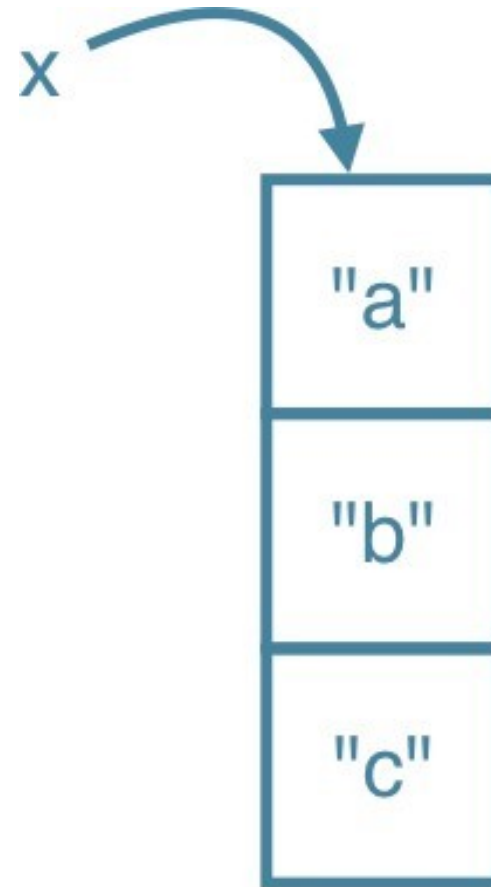
```
y[1] = "z"
```

```
y
```

```
['a', 'z', 'c']
```

```
x
```

```
['a', 'z', 'c']
```



Behind the scenes (I)

```
x = ["a", "b", "c"]
```

```
y = x
```

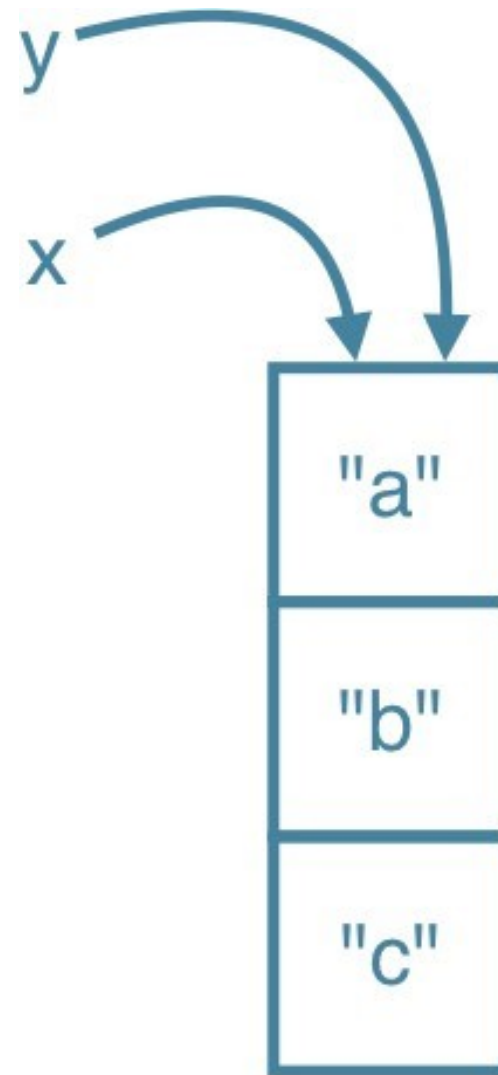
```
y[1] = "z"
```

```
y
```

```
['a', 'z', 'c']
```

```
x
```

```
['a', 'z', 'c']
```



Behind the scenes (I)

```
x = ["a", "b", "c"]
```

```
y = x
```

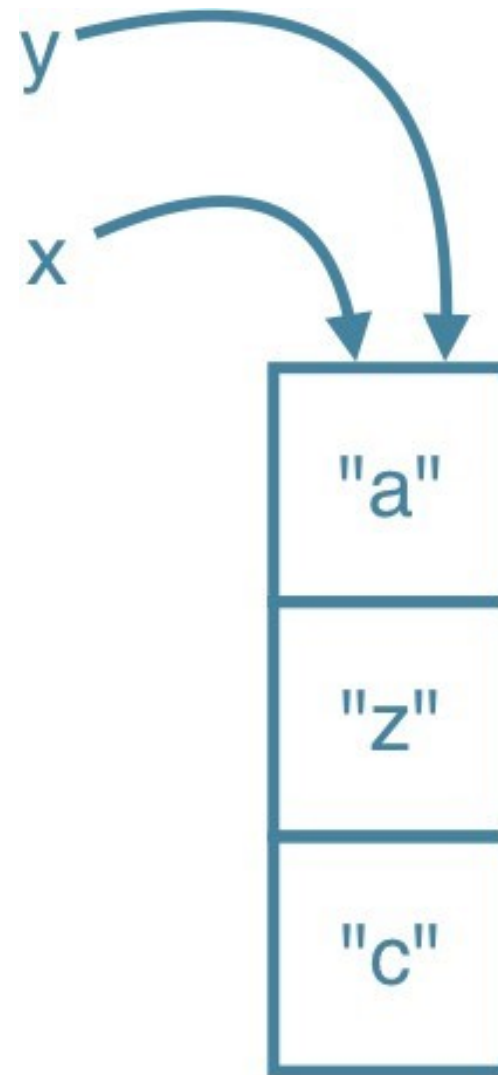
```
y[1] = "z"
```

```
y
```

```
['a', 'z', 'c']
```

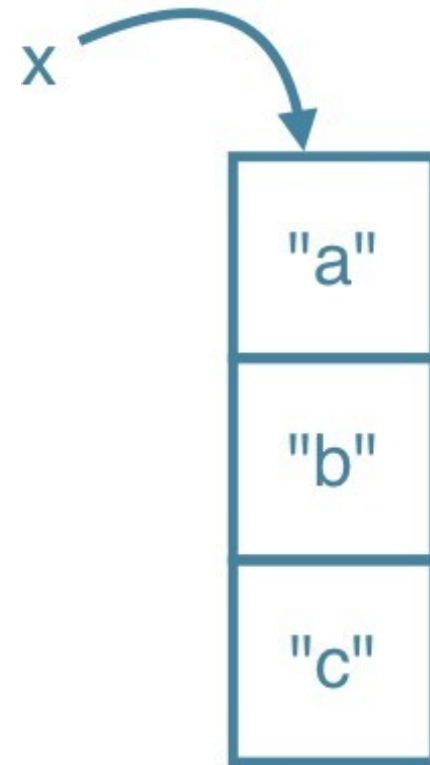
```
x
```

```
['a', 'z', 'c']
```



Behind the scenes (2)

```
x = ["a", "b", "c"]
```

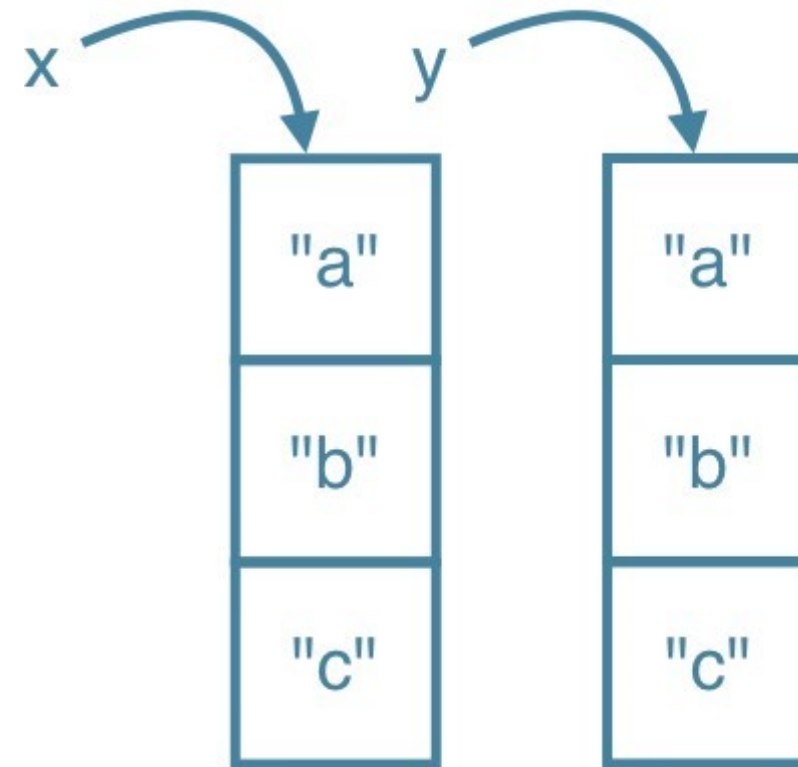


Behind the scenes (2)

```
x = ["a", "b", "c"]
```

```
y = list(x)
```

```
y = x[:]
```



Behind the scenes (2)

```
x = ["a", "b", "c"]
```

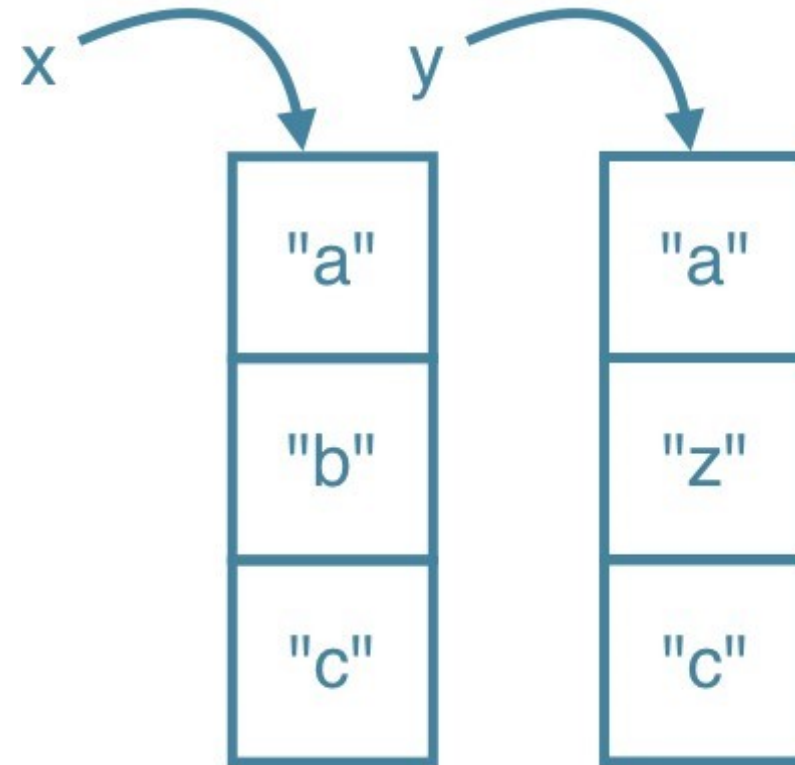
```
y = list(x)
```

```
y = x[:]
```

```
y[1] = "z"
```

```
x
```

```
['a', 'b', 'c']
```



Let's practice!

INTRODUCTION TO PYTHON

