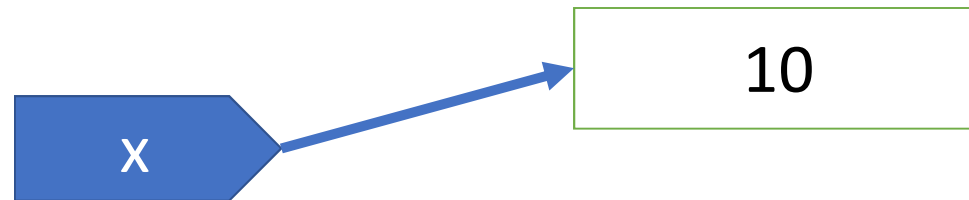


# Variable Assignment

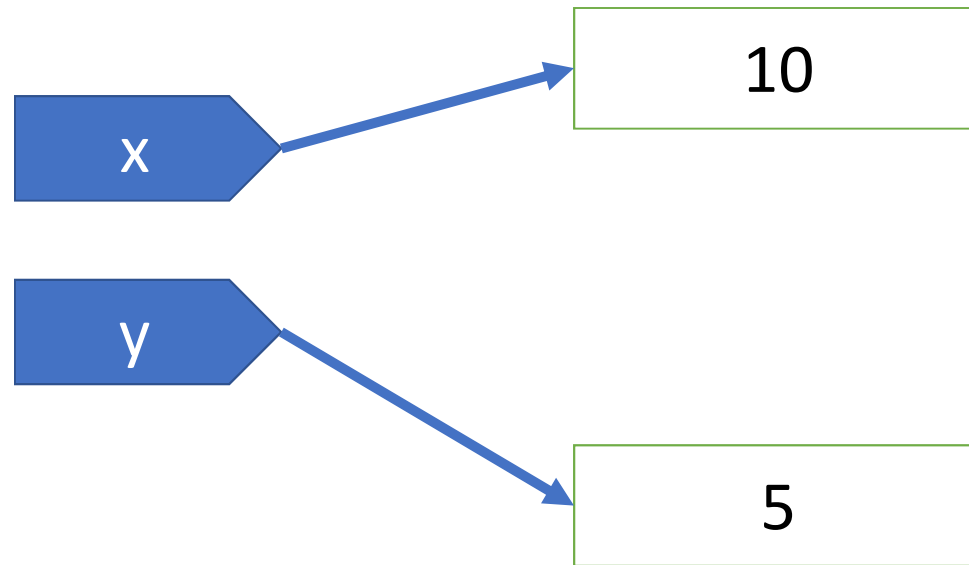
# Variables assignment

- $x = 10$



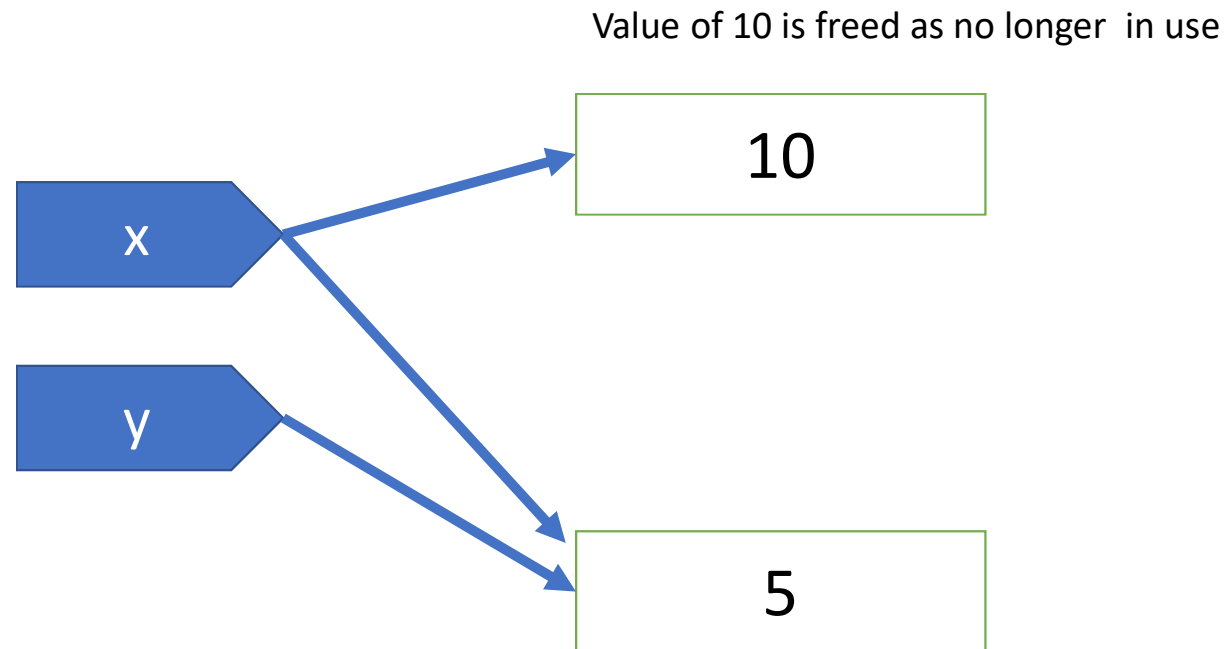
# Variables assignment

- $x = 10$
- $y = 5$



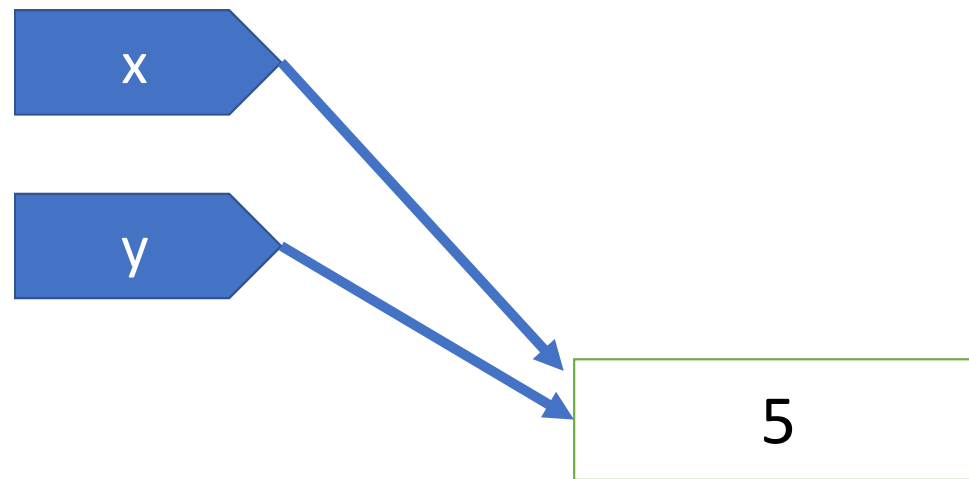
# Variables assignment

- $x = 10$
- $y = 5$
- $x = y$



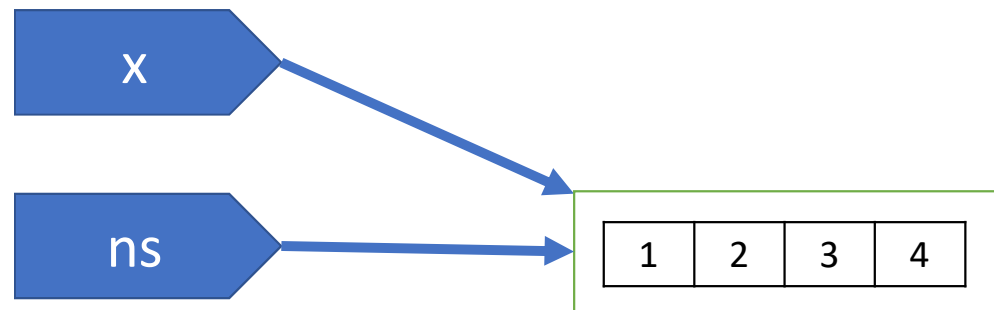
# Variables assignment

- $x = 10$
- $y = 5$
- **$x = y$**



# Variables assignment

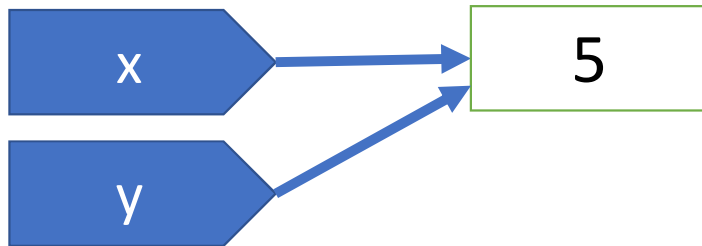
- `ns = [1, 2, 3, 4]`
- **`x = ns`**



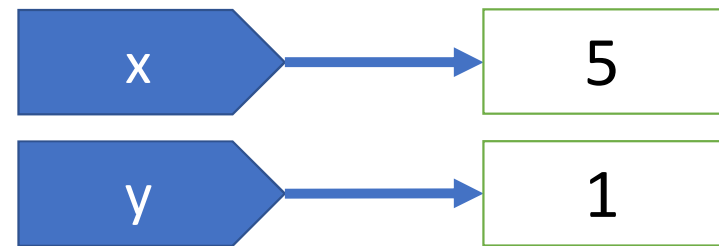
# Immutable types in Python

- **int, float, decimal, bool, string, tuple**
  - These objects has values that cannot be change once they are created.

```
x = 10  
y = 5  
x = y
```



```
x = 10  
y = 5  
x = y  
y = 1
```



# Immutable types in Python

- **int, float, decimal, boolean, string, tuple**
  - These objects has values that cannot be change once they are created.

```
from decimal import Decimal
a = 10 #integer
b = 0.1 #float
c = Decimal(b) #decimal
d = False #boolean
e = "Hello World" #string
f = (1, 2, 3, 4) #tuple. A tuple is a collection which is ordered and unchangeable.

print("a is an integer value of", a)
print("b is a float value of", b)
print("c is a decimal value of", c)
print("d is a bool value of", d)
print("e is a string value of '%s'." % e)
print("f is a tuple value of", f)
```

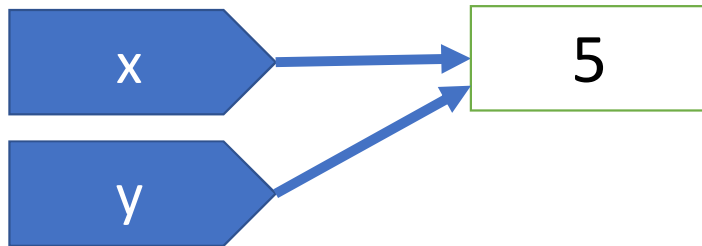
a is an integer value of 10  
b is a float value of 0.1  
c is a decimal value of 0.1000000000000000055511151231257827021181583404541015625  
d is a bool value of False  
e is a string value of 'Hello World'.  
f is a tuple value of (1, 2, 3, 4)



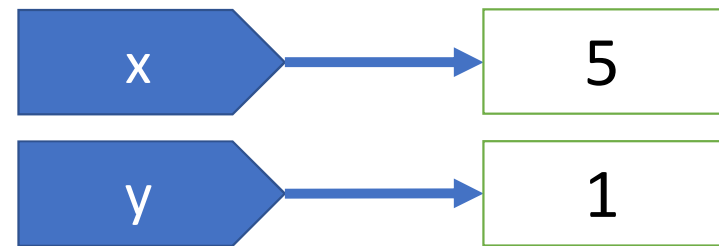
# Immutable types in Python

- **int, float, decimal, bool, string, tuple**
  - These objects has values that cannot be change once they are created.

```
x = 10  
y = 5  
x = y
```



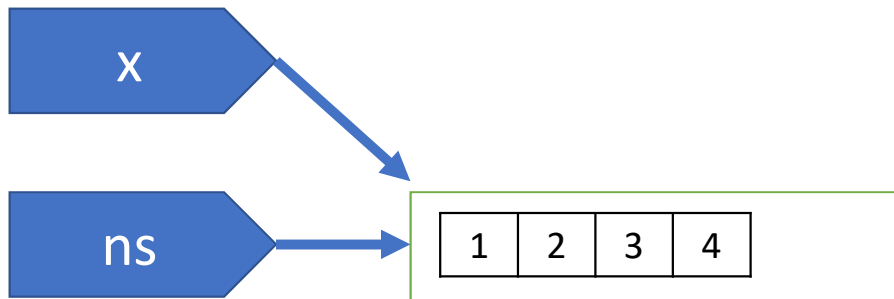
```
x = 10  
y = 5  
x = y  
y = 1
```



# Try appending a list

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

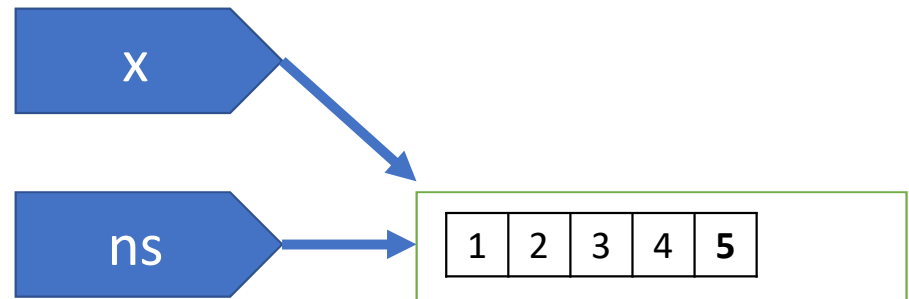
```
x = ns
```



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

```
x = ns
```

```
x.append(5)
```



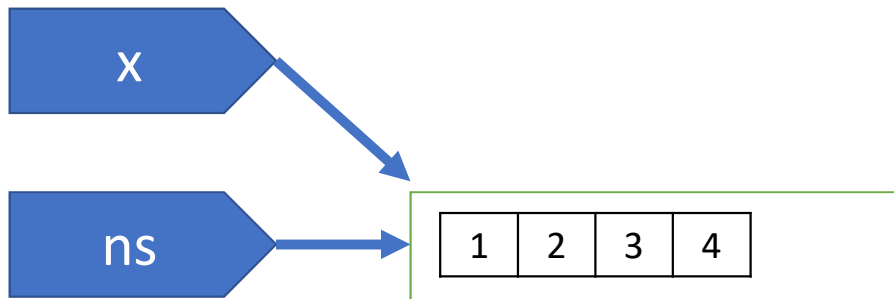
```
print (ns)
```

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

# Try appending a list

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

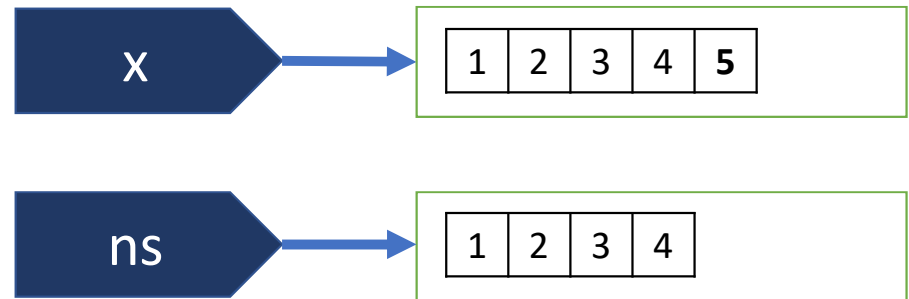
```
x = ns
```



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

```
x = ns
```

```
x = x + [5]
```



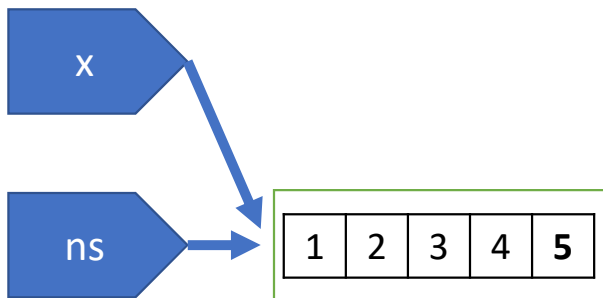
```
print (ns)
```

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

# Mutate vs Rebind

## Mutate object

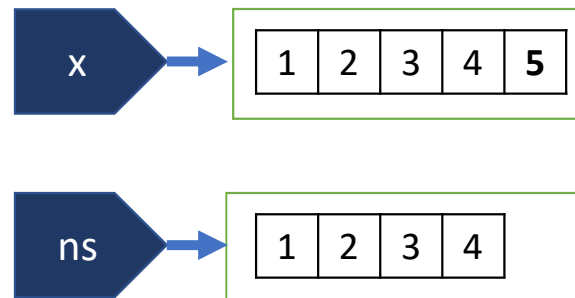
```
ns = [1, 2, 3, 4]
x = ns
x.append(5)
```



```
print (ns)
[1, 2, 3, 4, 5]
```

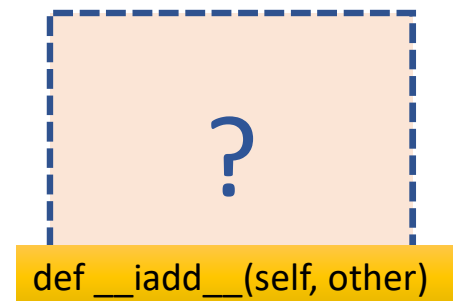
## Rebind – link to new object

```
ns = [1, 2, 3, 4]
x = ns
x = x + [5]
```



```
print (ns)
[1, 2, 3, 4]
```

```
ns = [1, 2, 3, 4]
x = ns
x += [5]
```



```
print (ns)
[1, 2, 3, 4, 5]
```

Working with copies of list

# Creating a copy of a list

- `new_list = old_list.copy()`
- `new_list = list(old_list)`
- `new_list = old_list[:]`
- Import copy  
`new_list = copy.copy(old_list)`
- Note that these copy are shallow. They do not copy objects within the list
  - Use `deepcopy` to also copy the objects, such as list, within list

```
Import copy
new_list = copy.deepcopy(old_list)
```

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

new_list1 = old_list.copy()
new_list2 = list(old_list)
new_list3 = old_list[:]

import copy
new_list4 = copy.copy(old_list)

old_list.append(5)  #change/mutate the original old_list

print(old_list)
print(new_list1)
print(new_list2)
print(new_list3)
print(new_list4)
```

# Recap on Notebook Creation and Markdown

# Markdown for documentation and comments

## Headers

- Display header text of 6 sizes
- Noted the space between the hash and header name
- Run the cell before the text is formatted for display

```
# Header 1  
## Header 2  
### Header 3  
#### Header 4  
##### Header 5  
##### Header 6
```

```
<h1>Header 1</h1>  
<h2>Header 2</h2>
```

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

Header 1

Header 2



# Markdown for documentation and comments

## Lists

- First number determine the start of an ordered lists
- Noted the space between the dot and list item
- Run the cell before the text is formatted for display

```
1. Feelings
  1. Happiness
  2. Sadness
  1. Anger
  10. Calm
```

```
2. Countries
```

```
3. Singapore
A. Indonesia
B. Malaysia
```



1. Feelings

A. Happiness

B. Sadness

C. Anger

D. Calm

2. Countries

C. Singapore A. Indonesia B. Malaysia

# Markdown for documentation and comments

## Lists

- Dash or asterisk can be used
- Noted the space between the dash and list item
- Run the cell before the text is formatted for display

```
-Fruits  
-apple  
-orange
```

```
- Tools  
* Hammer  
- Screwdriver
```



```
-Fruits -apple -orange
```

```
• Tools  
▪ Hammer  
▪ Screwdriver
```

# Markdown for documentation and comments

- Formatting **bold** and *italics* text

This is in `<i>italics</i>`.

This is in *italics*.

This is in `*italics*`.

This is in *italics*.

This is in `<b>bold</b>`.

This is in **bold**.

This is in `**bold**`.

This is in **bold**.

This is in `***bold italics***`.

This is in ***bold italics***.

# Markdown for documentation and comments

- Creating a table
  - Use pipe (|) and dash (-)
  - Alignment of columns will be made in the display output

```
|index|name|age|  
|---|---|---|  
|1|John Smith| 21|  
|2|Mary Jane|20|
```



index	name	age
1	John Smith	21
2	Mary Jane	20