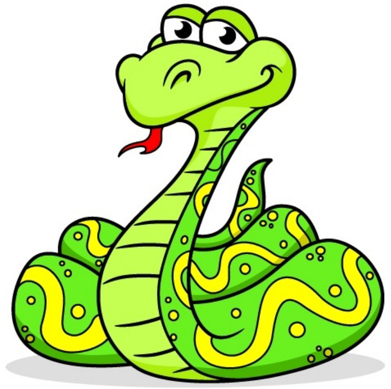
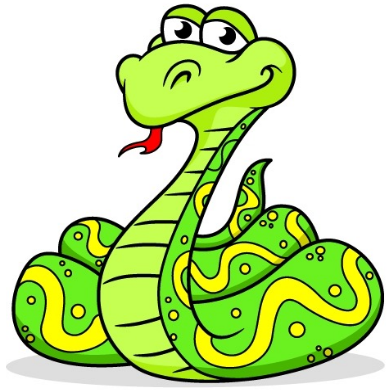
Simple Knowledge Series

Simple Knowledge Series



Programming in Python

Simple Knowledge Series



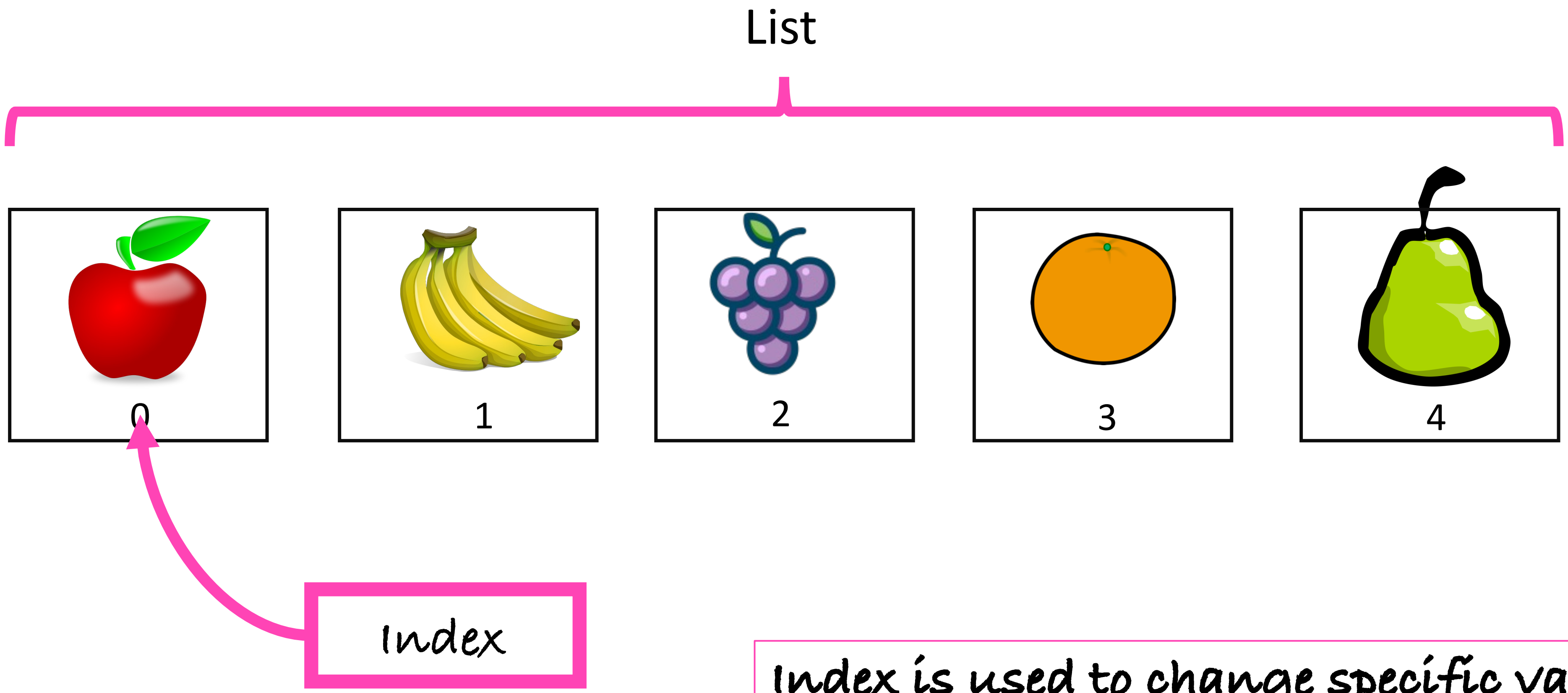
Data Types in Python

LISTS AND BOOLEAN EXPRESSIONS

LISTS

Each item is separated by a comma, but make sure the commas are on the outside of the quotation marks, so they are not included as part of the string

LISTS



Index is used to change specific values or items within the list

list[#]

Print(fruits[1])

Because the index starts at 0
the second item has an index of 1

This will print : bananas

You can also print out a section of the list by naming the range of indexes to print like this

Ending index

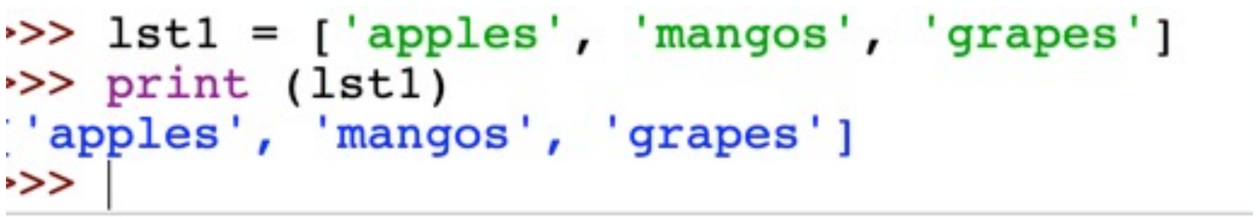
Print(fruits[1:4])

starting index

This will print : ['bananas', 'grapes', 'oranges',]

The ending index (4:pears)
Is not included

Be Aware



```
>>> lst1 = ['apples', 'mangos', 'grapes']
>>> print (lst1)
['apples', 'mangos', 'grapes']
>>> |
```

lists

- When you use **print( )** to display more than one item in a list the output will include the brackets and the quotation marks. This is because you are printing a list not just the item in a list

LISTS

To replace a value of an item within a list.

1. Reference (name) the index location
2. Reassign the value

```
fruits = ['apples', 'bananas', 'grapes', 'oranges', 'pears']

Fruits[2] = ['kiwi']
print(fruit)
['apples, 'bananas,, 'kiwi', 'oranges', 'pears']
```

To add an item to the end of a list use the **append( )** function

fruits.append('cherries")

['apples, 'bananas,, 'grapes', 'oranges', 'pears', ' cherries]

Insert ( )

fruits.insert(2, "peaches")

New item index location

Value to insert

['apples, 'bananas', 'peaches', 'grapes', 'oranges', 'pears', ' cherries]

To remove an item from the list use the **remove ( )** function

fruits.remove(‘banana”) – the list is now

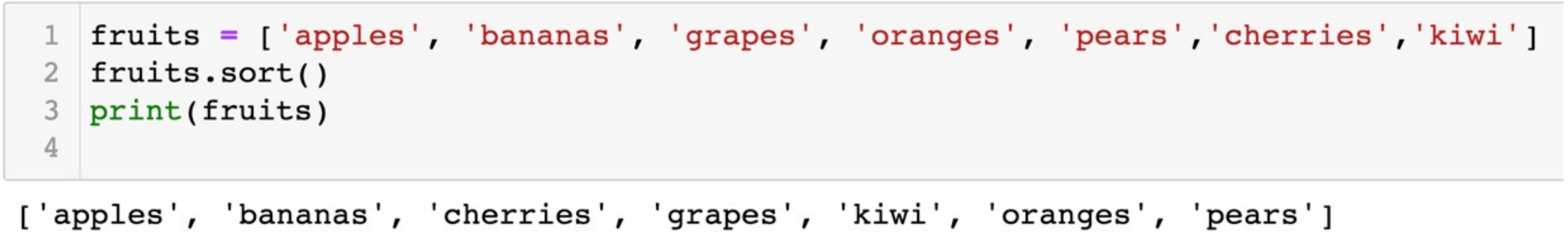
[‘apples’, ’peaches’, ‘grapes’, ‘oranges’, ‘pears’, ‘ cherries]

sort ( ) function – will put the list in numberical or alphabetical order

fruits.sort( )

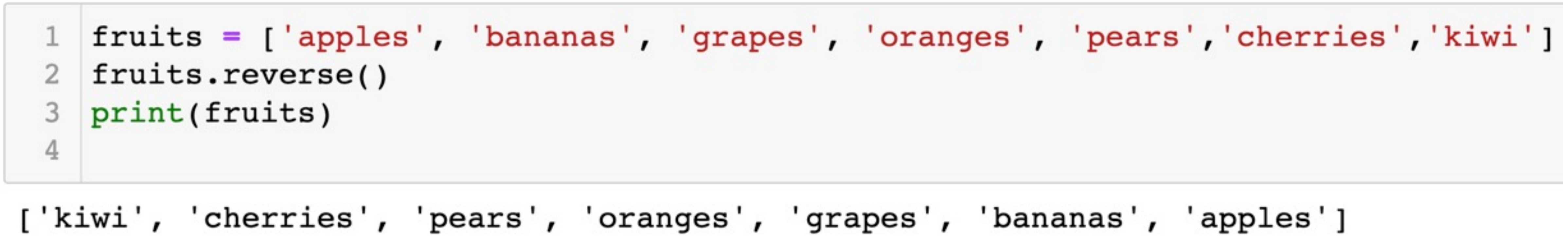
reverse ( ) function – will put the list in reverse order

fruits.reverse( )



```python
fruits = ['apples', 'bananas', 'grapes', 'oranges', 'pears','cherries','kiwi']
fruits.sort()
print(fruits)
```

```
['apples', 'bananas', 'cherries', 'grapes', 'kiwi', 'oranges', 'pears']
```



```python
fruits = ['apples', 'bananas', 'grapes', 'oranges', 'pears','cherries','kiwi']
fruits.reverse()
print(fruits)
```

```
['kiwi', 'cherries', 'pears', 'oranges', 'grapes', 'bananas', 'apples']
```

To get the number of item in a list use the **len ( )** function

fruits.remove('banana") – the list is now

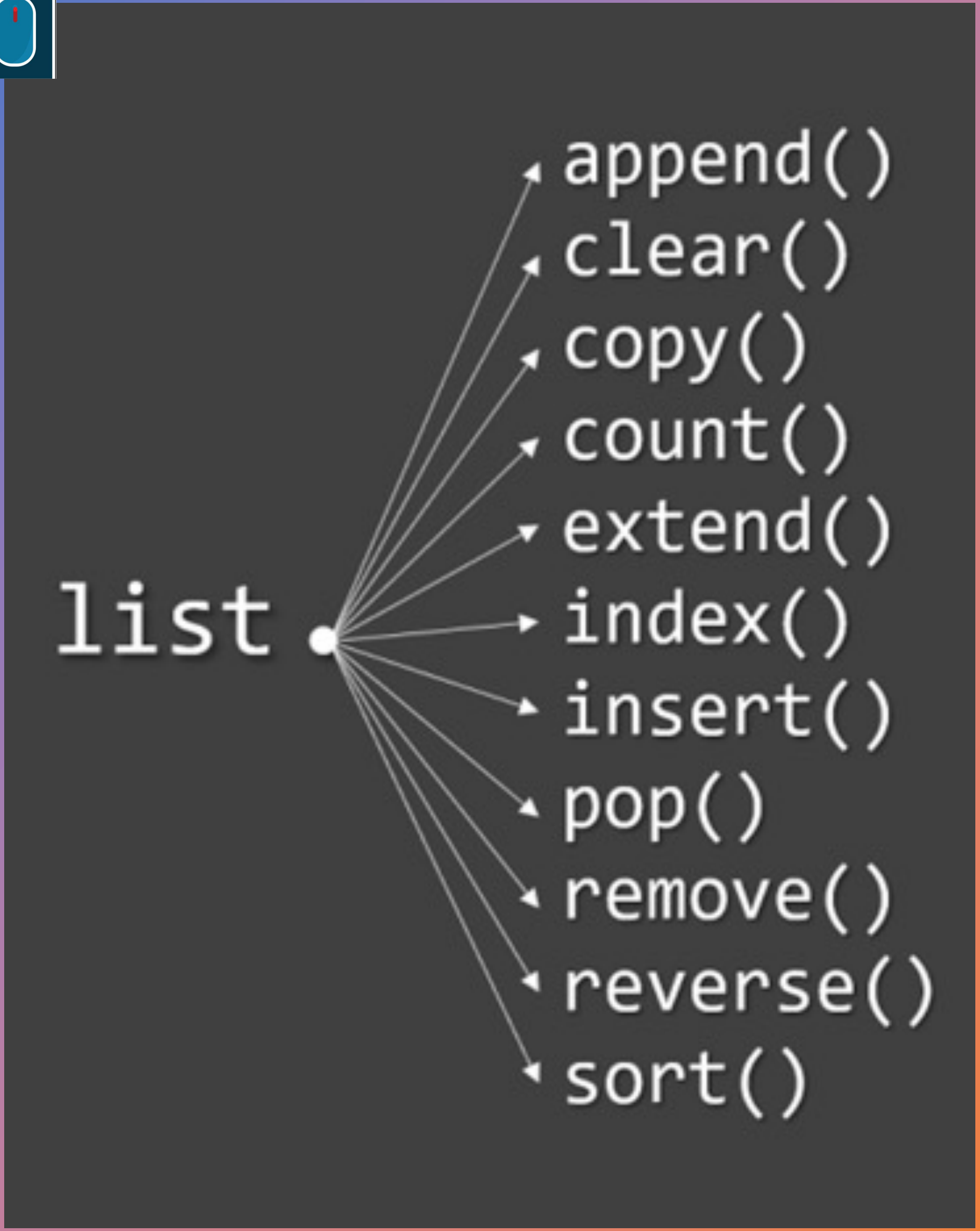
```
fruits = ['apples',  'peaches', 'grapes', 'oranges', 'pears', ' cherries]
fruit_length = len(fruits)
print(fruit_length)
```

Prints 6

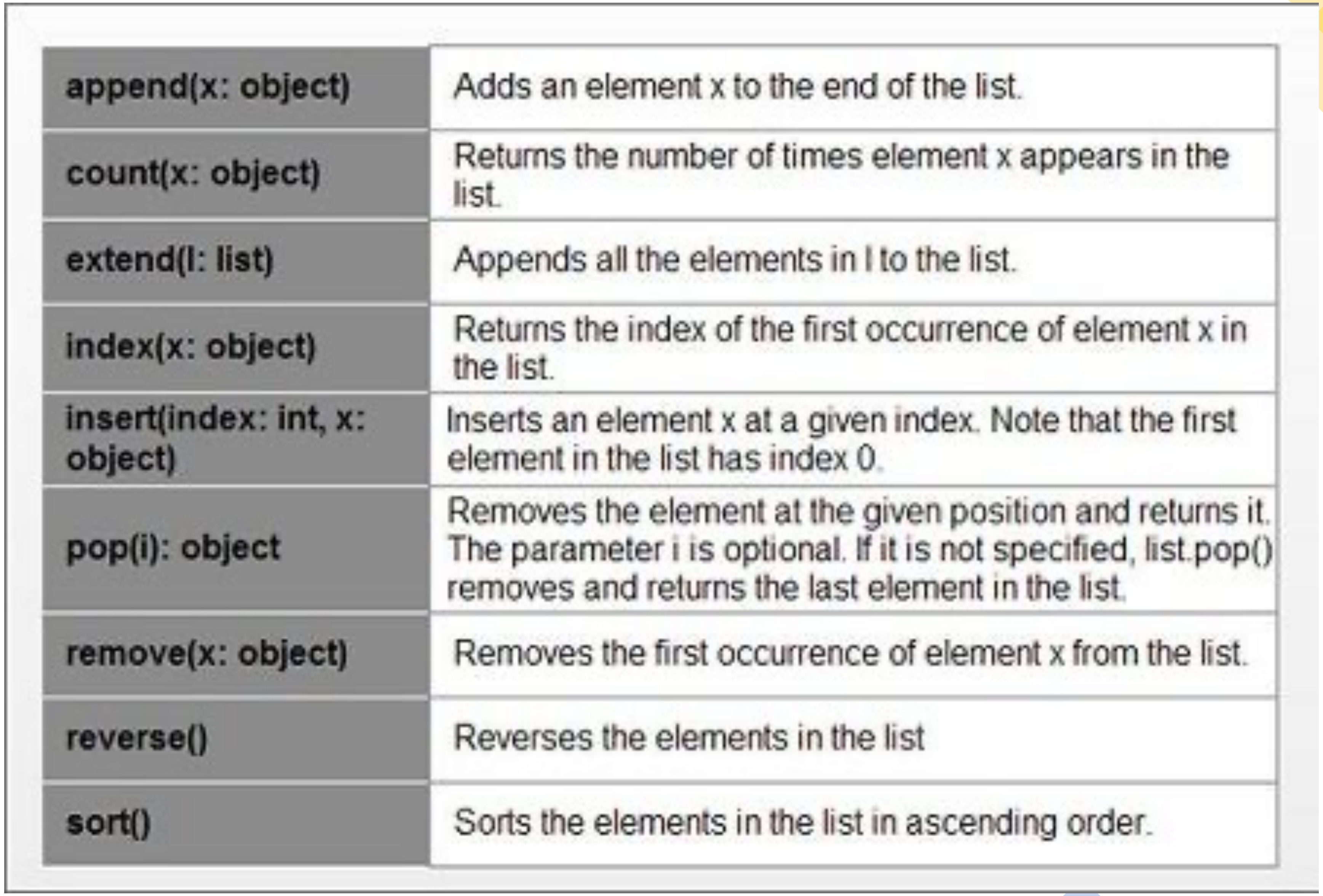
To add one list to the end of another list use the + operator

```
fruits = ['apples', 'pears', ' cherries]
Vegetables = ['spinach', 'carrot' ,peas']
produce =  fruit + vegetables
print (produce)
```

Prints ['apples', 'pears', ' cherries', 'spinach', 'carrot' ,peas'



List Functions



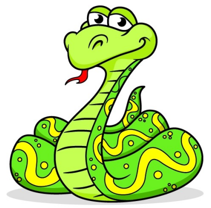
append(x: object)	Adds an element x to the end of the list.
count(x: object)	Returns the number of times element x appears in the list.
extend(l: list)	Appends all the elements in l to the list.
index(x: object)	Returns the index of the first occurrence of element x in the list.
insert(index: int, x: object)	Inserts an element x at a given index. Note that the first element in the list has index 0.
pop(i): object	Removes the element at the given position and returns it. The parameter i is optional. If it is not specified, list.pop() removes and returns the last element in the list.
remove(x: object)	Removes the first occurrence of element x from the list.
reverse()	Reverses the elements in the list
sort()	Sorts the elements in the list in ascending order.

Lists within lists

A list can be set within another list. The second list is called an **INNER LIST**.

$$\text{List} = [\overbrace{\text{"A", “B”, “C”,}}^{\text{Outer list}}\ \underbrace{\text{[“D1”, “D2”, “D3”]}}_{\text{Inner list stored in forth item location of the outer list}},\ \overbrace{\text{“E”]}}^{\text{Outer list continued}}$$

Lists within lists



FOR EXAMPLE

1. Enter the opening bracket
2. add the type of apples in quotation marks and. Then end with a closing bracket
3. continue with the rest of the list

Outer list

Inner list

fruits = [‘bananas’, [‘Gala’, ‘Granny Smith’, ’Empire’,
‘Golden Delicious’], ‘grapes’, ‘oranges’, ‘pears’]

Inner list

Outer list continued

Print(fruits[1]) prints [‘Gala’, ‘Granny Smith’, ’Empire’,‘Golden Delicious’]

Lists within lists

To print just the favorite apples (the inner list from the list of favorite fruits

```
print (fruits[1])
```

2nd item

List item

prints ['Gala', 'Granny Smith', 'Empire', 'Golden Delicious']

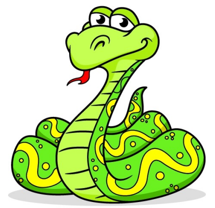
You can print a single apple name by using index location 1

```
print (fruits[1][2])
```

Specific apple

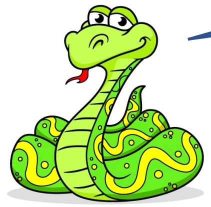
List of apples

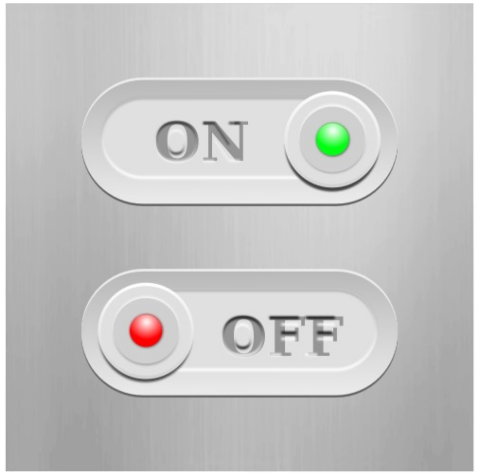
prints ['Empire']



Boolean expression

In programming it is very common to want to know if something is true or false.


BOOLEAN VARIABLES ARE LIKE LIGHT SWITCHES. THERE ARE ONLY TWO OPTIONS



when you are setting available to true or false,
make sure you capitalise it in T in truth and the F in false

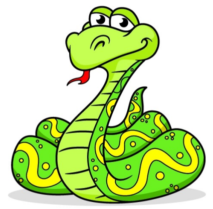
Boolean expression

```
height = 58
meet_ limit = height > 50
print (meet_limit)
```

Assigns "height" to the value of 58

Assigns "meet_limit" the values of "height > 50 ". This means that a height of greater 50 will be considered true

The expression is TRUE because the given height is 58 and 58 > 50
Which means that meet_limit is true

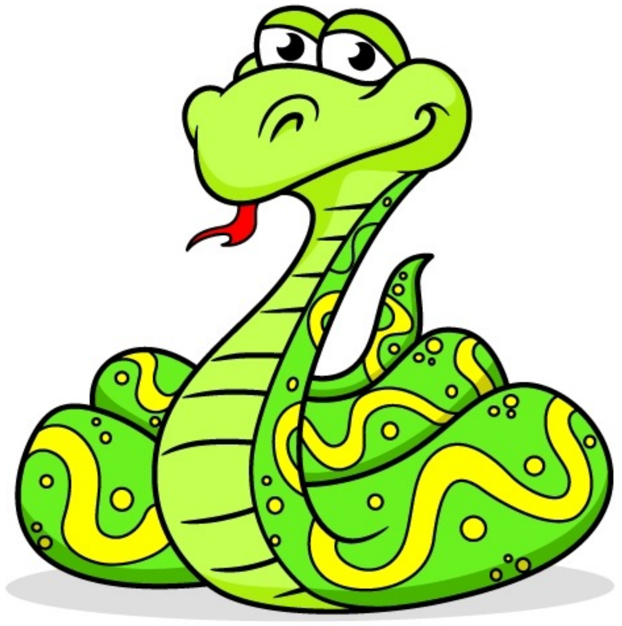


Boolean expression

This example show assigning the variable of test1 to the Boolean expression 2 is equal to4 which is false

```
test1 = 2 == 4
print (test1)
```

Prints: `False`



Comparison operators

Symbol	Meaning
==	Is equal to
!=	Is not equal to
<	Is less than
>	Is greater than
<=	Is less or equal to
>=	Greater than or equal to

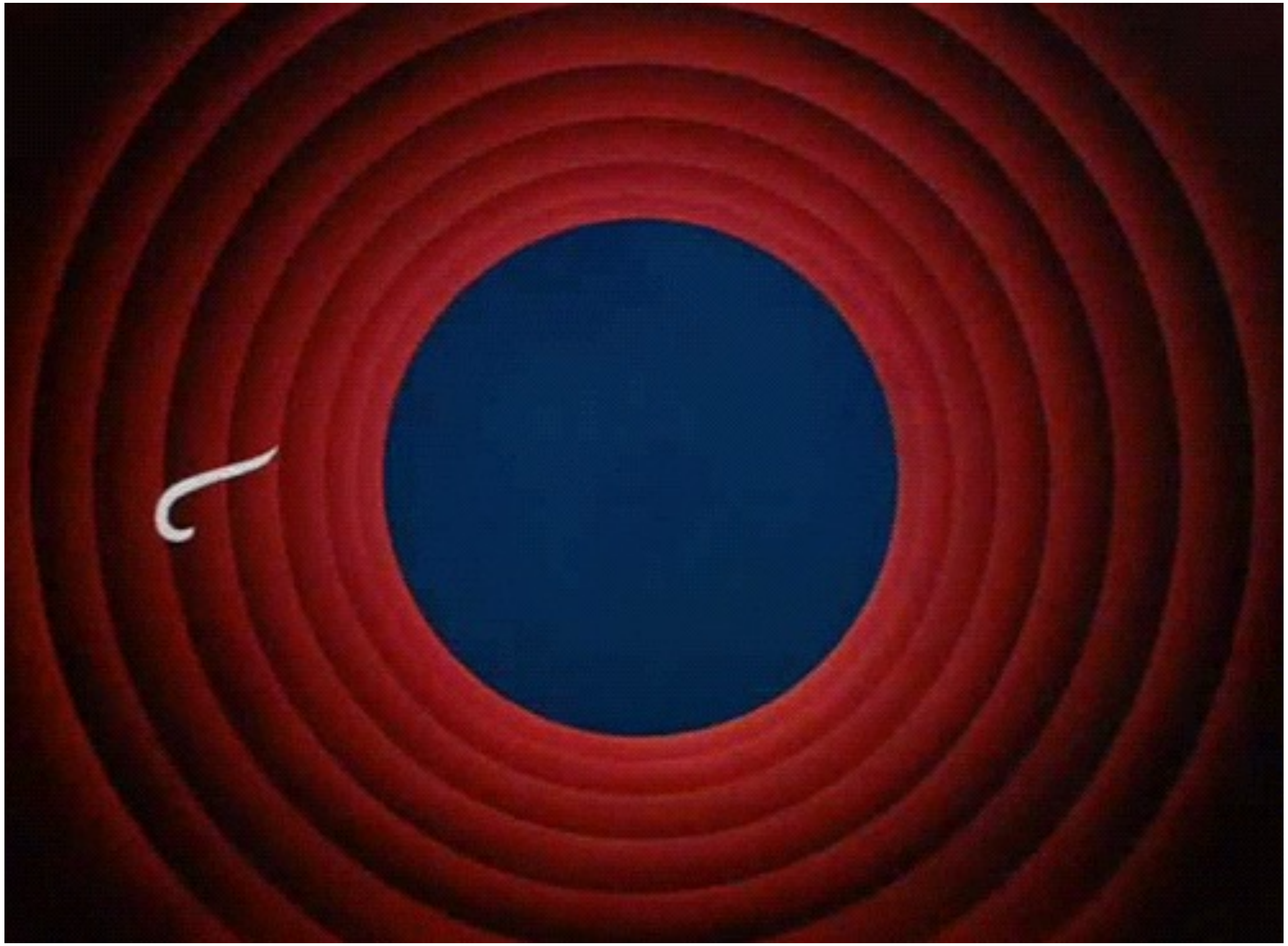
Comparison operators

Symbol	Meaning	Value
2 == 2	2 Is equal to 2	True
2 != 3	2 Is not equal to 3	True
2 < 3	2 Is less than 3	True
4 > 3	4 Is greater than 3	True
2 <= 3	2 Is less or equal to 3	True
5 >= 3	5 Greater than or equal to 3	True

Example of expressions that evaluate to True:

Symbol	Meaning	Value
2 == 5	2 Is equal to 5	False
2 != 2	2 Is not equal to 2	False
3 < 3	2 Is less than 3	False
2 > 3	4 Is greater than 3	False
5 <= 3	2 Is less or equal to 3	False
2 >= 3	5 Greater than or equal to 3	False

Example of expressions that evaluate to False:



CHECK YOUR KNOWLEDGE

1. ________________________ are used to store multiple values

2 What function should you use to add an item to the end of a list

1. ____Lists_______ are used to store multiple values

ANSWER

2 What function should you use to add an item to the end of a list

ANSWER Append ( )

3 How can you replace the second item in the list "cars with "Porsche"

4. If you want to ask a user their favourite colour, what would be a good function to use ? How would you write it

3. How can you replace the second item in the list "cars with "Porsche"

ANSWER Cars[1] = "Porsche"

4. What function should you use to add an Item between 2 Existing Items in a list

ANSWER Insert ( )

5. How do you store a list within a list?

6. Explain what the sort ( ) function does

5. How do you store a list within a list?

ANSWER Add a second set of brackets around the list inside another list. The inner list will take one item spot in the out list"

6. Explain what the sort ( ) function does

ANSWER The sort ( ) function rearranges the list into numerical or alphabetical order

7.. Write a code that would print "bananas" given the list

```
fruits = ['apples, 'bananas,, 'grapes', 'oranges', 'pears']
```

8. Which of the following does NOT evaluate to a Boolean value ?

A. True
B. 3**2
C. False
D. 3 > 2

$\boxed{7}$.. Write a code that would print "bananas" given the list

```
fruits = [‘apples, ‘bananas,, ‘grapes’, ‘oranges’, ‘pears’]
```

ANSWER

```
Print(fruits[1])
```

$\boxed{8}$. Which of the following does NOT evaluate to a Boolean value ?

ANSWER

A. True
B. 3**2
C. False
D. 3 > 2

9 What will the following program print?

```
score = 3
game_over = score > 5
print (game_over)
```

10 What will the following code print?

```
print (100 ==25)
```

9 What will the following program print?

```
score = 3
game_over = score > 5
print (game_over)
```

ANSWER False

10 What will the following code print?

```
print (100 ==25)
```

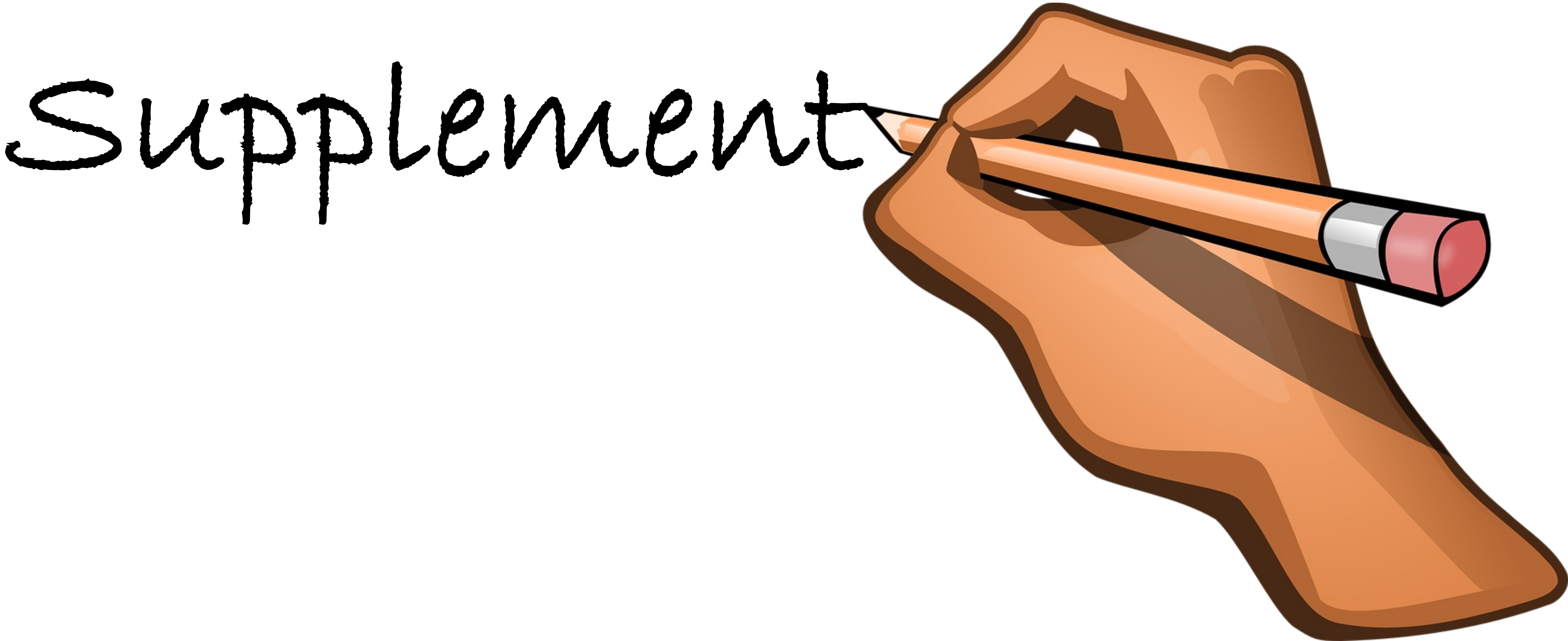
ANSWER False

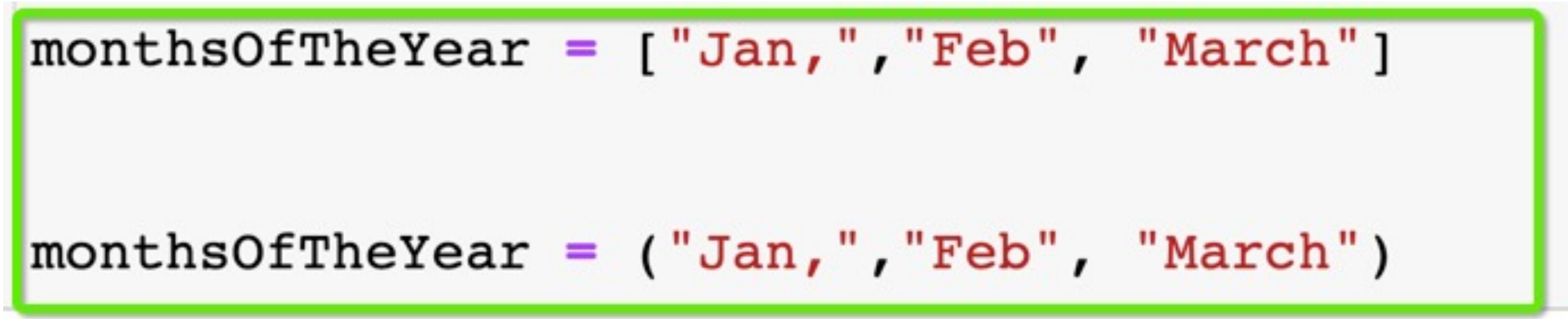
11. How is “==“ different from “=“ in python

11. How is “==“ different from “=“ in python

ANSWER

“==” is a comparison operator – checks if two values are exactly the samеariable
“=“ is the assignment operator and assigns a value to a variable

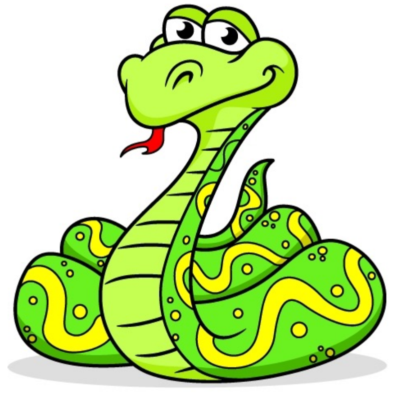

Supplement


```
monthsOfTheYear = ["Jan,","Feb", "March"]


monthsOfTheYear = ("Jan,","Feb", "March")
```

Simple Knowledge Series

Tuples



TUPLE

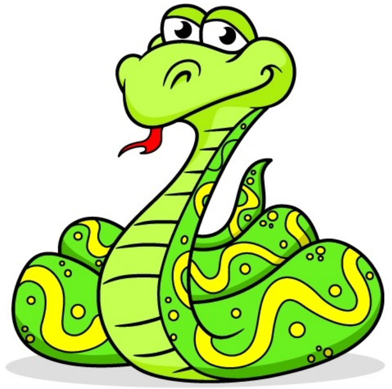
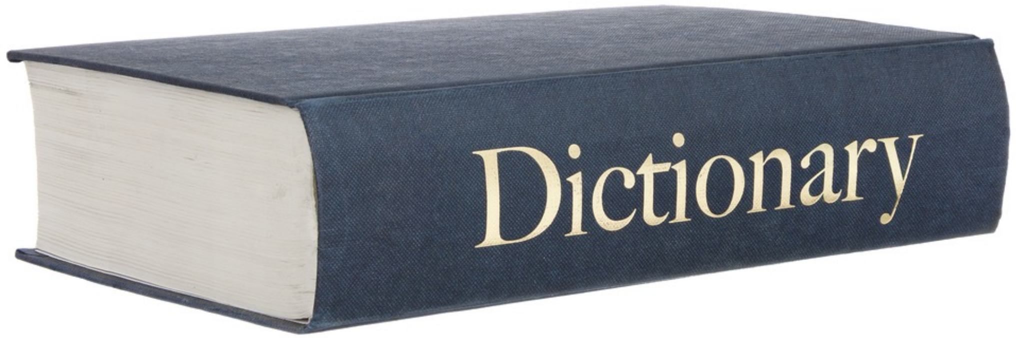
- used to store multiple items in a single variable
- values are ordered and CANNOT be modified

```
thistuple = ("apple", "banana", "cherry")

thistuple [0] = ”apple”
thistuple [-1] = ”cherry”
thistuple [-3] = ”cherry”
```

Simple Knowledge Series

Dictionary

Dictionary

- Dictionaries are used to store data values in key:value pairs.
- A dictionary is a collection which is ordered,
- changeable and do not allow duplicates.
- Dictionaries are written with curly brackets
- To declare a dictionary, you write -

Dictionary

```
d = { <key>: <value>,
 <key>: <value>,
   .
   .
  .
<key>: <value>
 }
```

Dictionary

Example

```
mydictionary = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

userNameAndAge = { "Peter" : 30, "James" : 32,
"John" : 17}
```