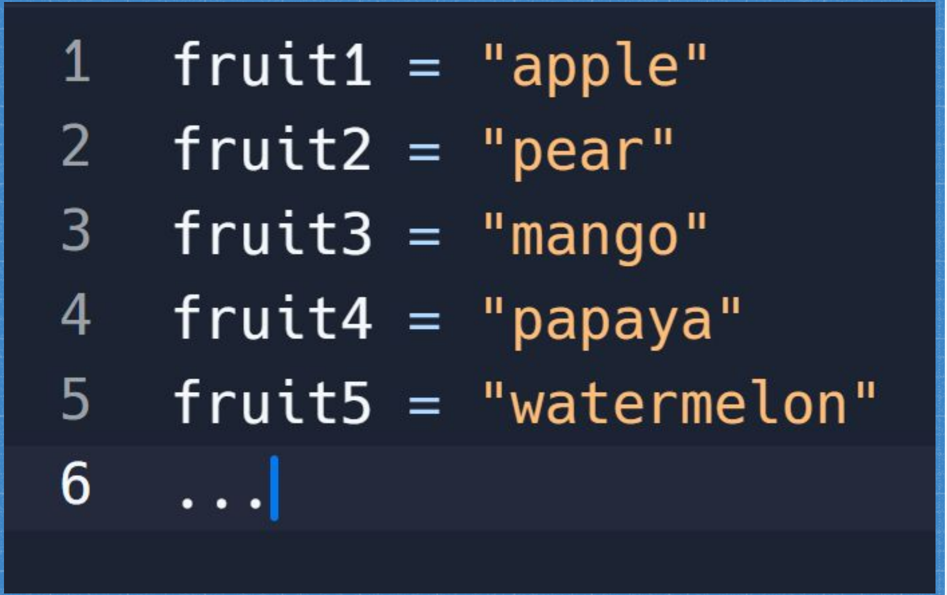
Day 3:
Python Intro

1

Lists

Why use lists?

- Variables can only store one value
- Lists can make the code on the right much shorter

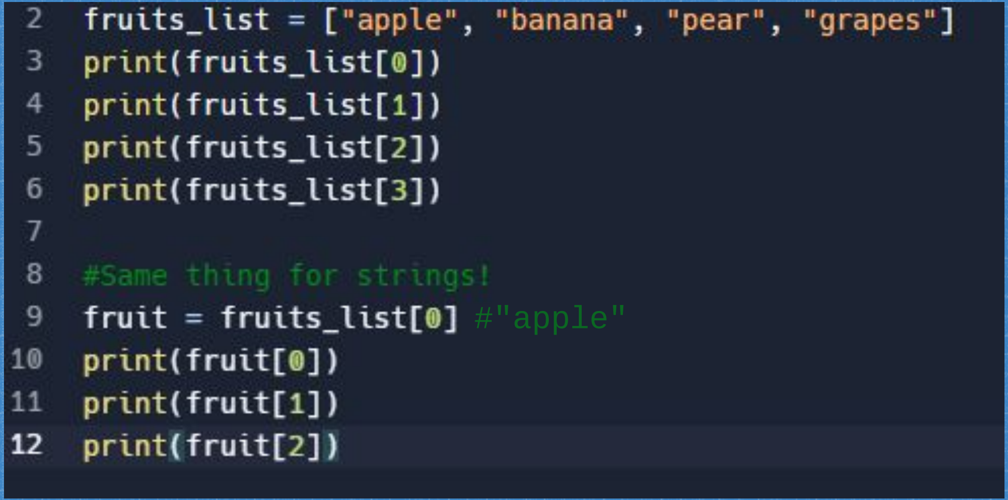


```
fruit1 = "apple"
fruit2 = "pear"
fruit3 = "mango"
fruit4 = "papaya"
fruit5 = "watermelon"
...
```

How to use lists

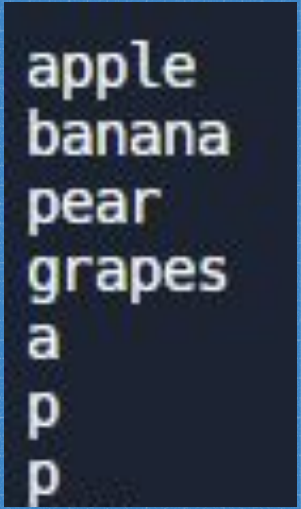
- Separate items with commas (items can be strings,numbers, and even lists)
- Access items by indexing

Index 0 1 2 3

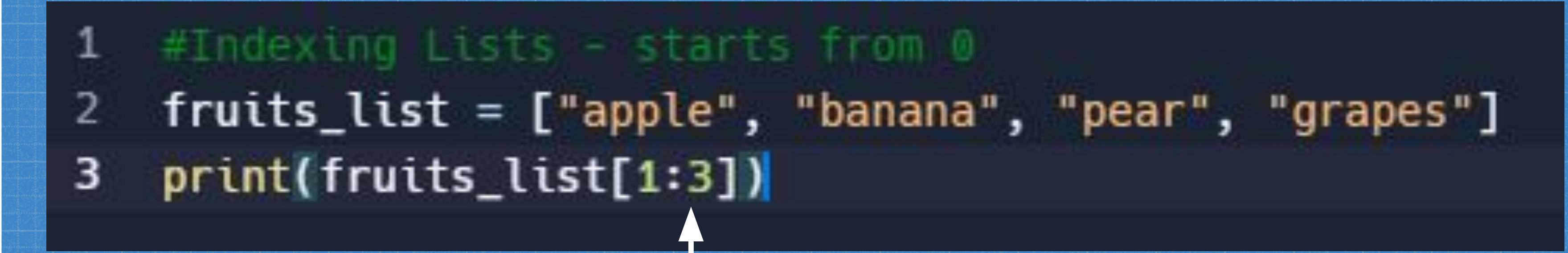


```python
fruits_list = ["apple", "banana", "pear", "grapes"]
print(fruits_list[0])
print(fruits_list[1])
print(fruits_list[2])
print(fruits_list[3])

#Same thing for strings!
fruit = fruits_list[0] #"apple"
print(fruit[0])
print(fruit[1])
print(fruit[2])
```

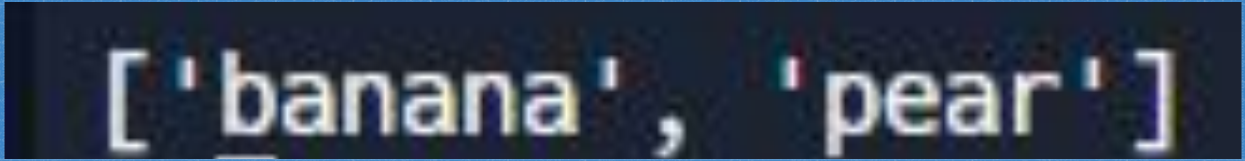


```
apple
banana
pear
grapes
a
p
p
```



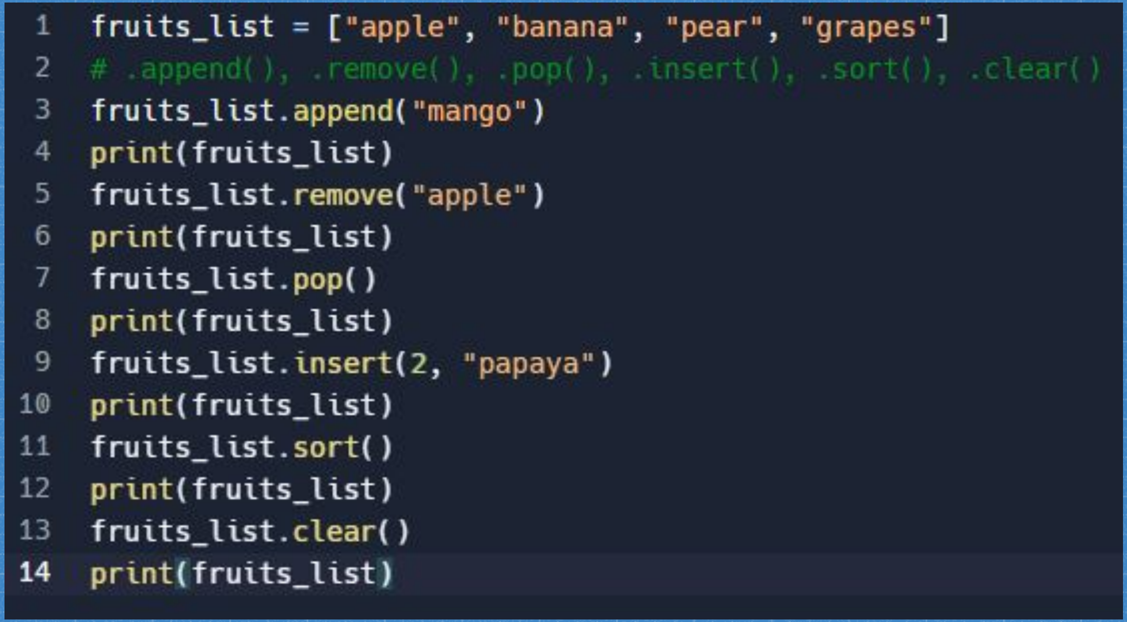
```python
#Indexing Lists - starts from 0
fruits_list = ["apple", "banana", "pear", "grapes"]
print(fruits_list[1:3])
```

Starting from index 1, up to but not including index 3

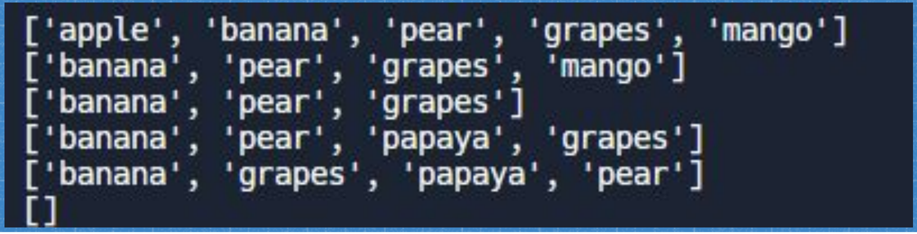


```
['banana', 'pear']
```

List Methods



```python
fruits_list = ["apple", "banana", "pear", "grapes"]
# .append( ), .remove( ), .pop( ), .insert( ), .sort( ), .clear( )
fruits_list.append("mango")
print(fruits_list)
fruits_list.remove("apple")
print(fruits_list)
fruits_list.pop()
print(fruits_list)
fruits_list.insert(2, "papaya")
print(fruits_list)
fruits_list.sort()
print(fruits_list)
fruits_list.clear()
print(fruits_list)
```



```
['apple', 'banana', 'pear', 'grapes', 'mango']
['banana', 'pear', 'grapes', 'mango']
['banana', 'pear', 'grapes']
['banana', 'pear', 'papaya', 'grapes']
['banana', 'grapes', 'papaya', 'pear']
[]
```

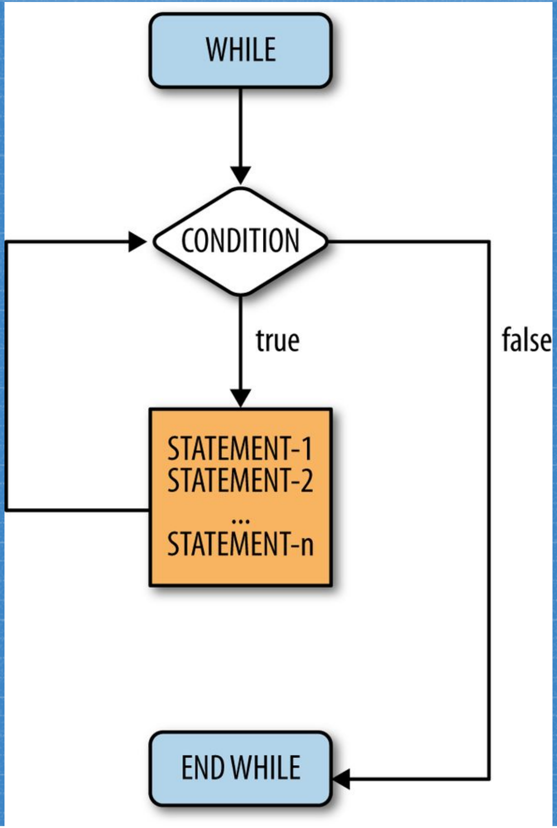
2

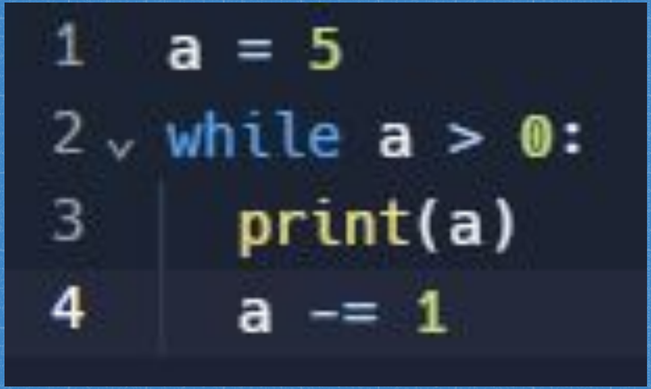
While Loops

Booleans Review:

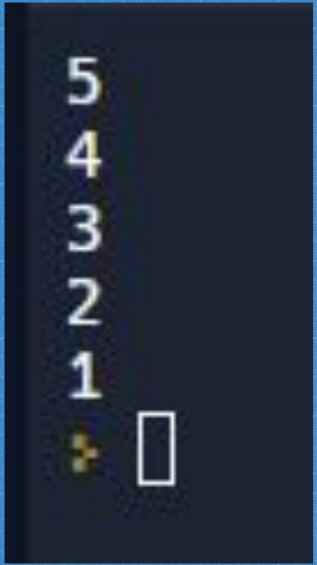
- True or False:
  - Today is Wednesday
  - It is raining

What are "while" loops?





```python
a = 5
while a > 0:
    print(a)
    a -= 1
```



```
5
4
3
2
1
>
```

Practice:

Let’s edit our calculator program so that instead of running just once, we can run it until the user types stop.

Use another input() inside of the loop to ask the user if they would like to stop the program before every calculation. If the user types “stop,” set the condition for the while loop to False which will stop the loop.

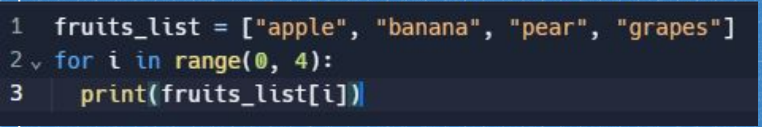
3

For Loops

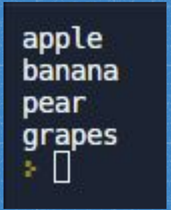
What are "for" loops?

- Iterates (runs through) a fixed number of times
- 4 Parts: "**for**" keyword, **variable**, "**in**", **iterable**

Ex. 1 (range):

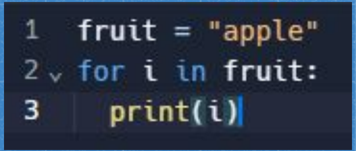


```python
fruits_list = ["apple", "banana", "pear", "grapes"]
for i in range(0, 4):
    print(fruits_list[i])
```

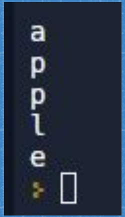


```
apple
banana
pear
grapes
```

Ex. 2 (string):

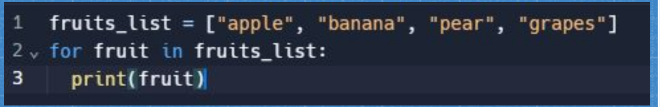


```python
fruit = "apple"
for i in fruit:
    print(i)
```

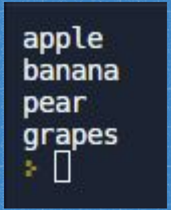


```
a
p
p
l
e
```

Ex. 3 (list):



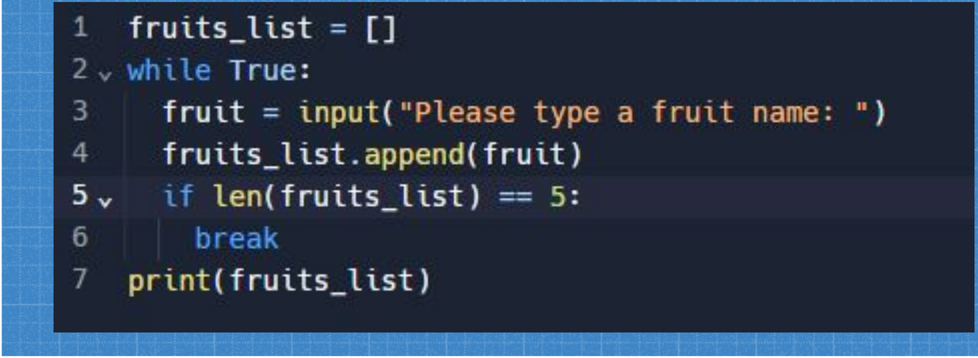
```python
fruits_list = ["apple", "banana", "pear", "grapes"]
for fruit in fruits_list:
    print(fruit)
```



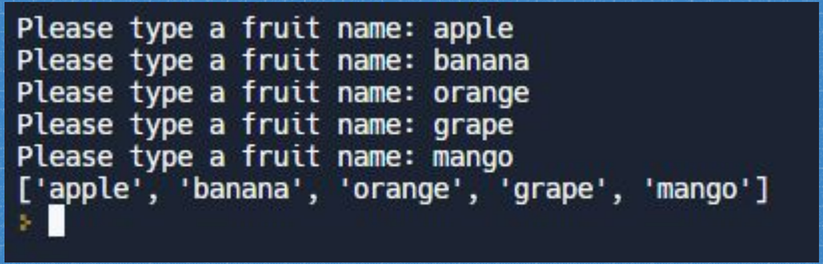
```
apple
banana
pear
grapes
```

"break"

- To stop a while/for loop, you can also use a break statement
- Try including the break statement in your code
- "while True" is an infinite loop



```python
fruits_list = []
while True:
  fruit = input("Please type a fruit name: ")
  fruits_list.append(fruit)
  if len(fruits_list) == 5:
    break
print(fruits_list)
```



```
Please type a fruit name: apple
Please type a fruit name: banana
Please type a fruit name: orange
Please type a fruit name: grape
Please type a fruit name: mango
['apple', 'banana', 'orange', 'grape', 'mango']
> 
```

Let’s Practice!

1. Make a shopping list program
   a. Use a while loop to ask the user for an item until they type "stop"
   b. Print the list of items using a for loop
2. https://codingbat.com/python/List-1