

# Python Part 2

## Lists

In addition to numbers and strings, Python also lets us represent lists of things (just like in AppInventor).

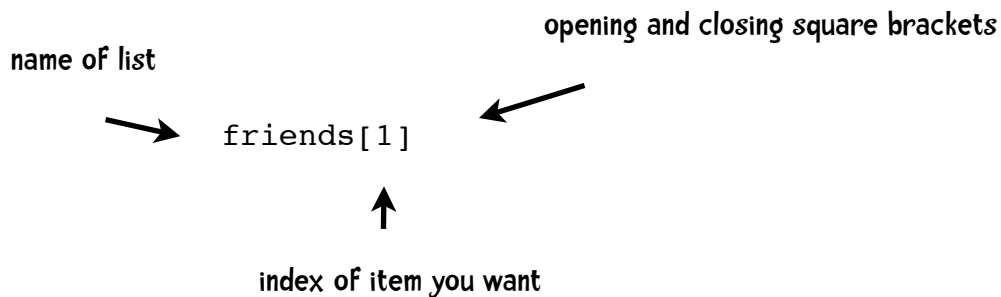
### Syntax of lists:

- the listed is delimited by square brackets: [ ]
- list items are separated by a comma
- list items can be any type of thing in Python (numbers, strings, lists, etc.)

### Here are some examples:

```
>>> friends = ['Lefty', 'Slim', 'Jake', 'Mustang Sally', 'Dusty']
>>> grades = [94, 87, 93, 90]
```

Just like in AppInventor, I can access individual members of a list by using an index. In this case the syntax is



In Python, as in most programming languages (except AppInventor) indices start at 0 (we start counting at zero).

```
>>> friends[0]
'Lefty'
>>> friends[1]
'Slim'
>>> grades[0]
94
```

You can get the last element in a list by using -1; the penultimate by using -2, etc.

```
>>> friends[-1]
'Dusty'
>>> friends[-2]
'Mustang Sally'
```

Finally you can get a sublist of a list by using a colon:

```
>>> friends[1:3]
['Slim', 'Jake']
>>> friends[1:4]
['Slim', 'Jake', 'Mustang Sally']
```

if you don't have a number to the left of the colon it starts at the beginning of the list; no number to the right of the colon means go to the end of the list:

```
>>> friends[1:]
['Slim', 'Jake', 'Mustang Sally', 'Dusty']
>>> friends[:3]
['Lefty', 'Slim', 'Jake']
>>>
```

### Checking if something is in a list

We can check to see if something is in a list by using 'in'

```
>>> 'Ann' in friends
False
>>> 'Slim' in friends
True
```

#### A short program

```
friends = ['Lefty', 'Slim', 'Jake', 'Mustang Sally', 'Dusty']
name = input('Enter name: ')
if name in friends:
    print("You are my friend")
else:
    print("Would you like to be my friend?")
```

## Adding and appending to lists

You can add lists together using the + operator:

```
>>> catWranglers = ['Pancho', 'Tex', 'Annie']
>>> bigList = friends + catWranglers
>>> bigList
['Lefty', 'Slim', 'Jake', 'Mustang Sally', 'Dusty', 'Pancho', 'Tex',
'Annie']
```

And you can append an item to the end of a list:

```
>>> catWranglers.append('Woody')
>>> catWranglers
['Pancho', 'Tex', 'Annie', 'Woody']
```

## Finding the length of a list

```
>>> len(catWranglers)
4
>>> len(friends)
5
```

## sorting a list

```
>>> friends
['Lefty', 'Slim', 'Jake', 'Mustang Sally', 'Dusty']
>>> friends.sort()
>>> friends
['Dusty', 'Jake', 'Lefty', 'Mustang Sally', 'Slim']
```

## Iterating through a list

### while loops

we can use our handy-dandy while loop to iterate through a list:

```
friends = ['Lefty', 'Slim', 'Jake', 'Mustang Sally', 'Dusty']
friends.sort()
numberOfFriends = len(friends)
i = 0
while i < numberOfFriends:
    print(friends[i])
    i += 1
```

when run this will print

```
Dusty
Jake
Lefty
Mustang Sally
Slim
```

## For loops

We can also use a new type of loop, a for loop. The syntax is:

a variable name you invent  
at each iteration, the current list item will  
be assigned to this variable

the name of an existing list



```
for item in list:
    your code that does something with item
```

For example

```
for friend in friends:
    print(friend)
```

prints

```
Dusty
Jake
Lefty
Mustang Sally
Slim
```

## range()

You can automatically create lists of integers by using the range function. The syntax is

```
range(start, end, step)
```

The start and step are optional. By default a list starts at 0 and increments (steps by 1). Here is an example where I create a list of 10 numbers and print them:

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

This prints out the numbers 0 through 9.

```
for i in range(10, 1, -1):  
    print(i)
```

(Start at 10; stop at 1; and decrement by one every time)  
prints the numbers 10 to 1 (our familiar blastoff example)

## Strings revisited

Much of what we have just done with lists we can do with strings. (Both are sequences of things after all)

consider:

```
>>> sentence = 'Pancho Villa invaded Columbus, New Mexico'
```

**using an index:**

```
>>> sentence[0]  
'P'  
>>> sentence[1]  
'a'  
>>> sentence[-1]  
'o'  
>>> sentence[7:12]  
'Villa'
```

**is something in the string?**

```
>>> 'Villa' in sentence  
True  
>>> 'Lefty' in sentence  
False  
>>> 'cho' in sentence  
True
```

**finding the length of a string (the number of characters in the string)**

```
>>> len(sentence)  
41  
>>> vowels = 'aeiou'  
>>> len(vowels)  
5
```

**iterating through a string character by character**

```
sentence = 'Pancho Villa invaded Columbus, New Mexico'  
for character in sentence:  
    print(character)
```

Here's a short program that will count how many vowels are in a string:

```
vowels = 'AEIOUaeiou'
# totalVowels keeps track of how many vowels I've seen
totalVowels = 0

sentence = input('Input sentence: ')
for ch in sentence:
    if ch in vowels:
        totalVowels += 1
print('There were', totalVowels, 'vowels in that sentence')

>>> name = 'Anna'
>>> grade = 92
>>> grade1 = 92
>>> grade2 = 100
>>> average = (grade1 + grade2) / 2
>>> average
96.0
```

### **split - dividing a string into words**

This is cool. We can divide a string into words. The result will be a list of words:

```
>>> sentence = 'Pancho Villa invaded Columbus, New Mexico'
>>> sentence
'Pancho Villa invaded Columbus, New Mexico'
>>> words = sentence.split()
>>> words
['Pancho', 'Villa', 'invaded', 'Columbus,', 'New', 'Mexico']
>>>
```

Here's a program that asks a user for a sentence and prints each word of the sentence one per line:

```
# get a sentence
sentence = input('Input sentence: ')
# divide it into words (words is a list of words)
words = sentence.split()
# now for each word in the list, print it
for word in words:
    print(word)
```

# Python Team Worksheet

 **Team:\_\_\_\_\_**

Name of team members present \_\_\_\_\_

---

---

1. Type in the following code in the Python shell and explain what happens and why it happens:

```
>>> s = ['a', 'cat', 'in', 'the', 'hat']
>>> length = len(s)
>>> length
5
>>> s[length]
```

Members who completed this task: \_\_\_\_\_

2. Type in, run the following code, and observe the results

```
sentence = input('Input sentence: ')
words = sentence.split()
numWords = len(words)
for i in range(numWords - 1, -1, -1):
    print(words[i], end = ' ')
```

In a sentence describe what this code does:

change the for line to

```
for i in range(numWords - 1, 0, -1):
```

Explain what happens and why this happens:

change the for line to

```
for i in range(numWords, -1, -1):
```

Explain what happens and why this happens:

Members who completed this task: \_\_\_\_\_

**3. Write a program that asks a user for a sentence and prints how many words are in that sentence.**

```
Input sentence: Build a man a fire, and he'll be warm for a day. Set  
a man on fire, and he'll be warm for the rest of his life
```

```
There are 27 words in that sentence
```

Members who completed this task: \_\_\_\_\_

**4. Write a program that starts with this line:**

```
grades = [88, 92, 87, 90, 96, 89]
```

It computes and prints the average grade.

Members who completed this task: \_\_\_\_\_

**5. Alter this program:**

```
vocabWord = input('Enter vocabulary word: ')  
# when the user just presses enter stop  
while vocabWord != '':  
    print(vocabWord)  
    vocabWord = input('Enter next word: ')
```



so that it prints the entered words in alphabetical order. Here's an example of input and output:

```
Enter vocabulary word: sapo
Enter next word: sepo
Enter next word: ano
Enter next word: dias
Enter next word: mes
Enter next word:
```

VOCABULARY LIST

```
-----
ano
dias
mes
sapo
sepo
```

Members who completed this task: \_\_\_\_\_

6. Write a program that asks a user for a sentence and prints out the words in reverse:

```
Input sentence: Colorless green ideas sleep furiously
furiously sleep ideas green Colorless
```

Members who completed this task: \_\_\_\_\_