

Lists I

What is a list?

- Lists are strings on steroids.
- A ***string*** stores an ordered sequence of single characters.
- A ***list*** stores an ordered sequence of *any* data type.



String "banana" →

0	1	2	3	4	5
"b"	"a"	"n"	"a"	"n"	"a"

List of ints →

0	1	2	3	4	5
98	85	90	100	75	88

List of floats →

0	1	2
34.5	-66.12	2347.0

List of strings →

0	1	2
"cat"	"dog"	"fish"

Why use lists?

- Lists exist so programmers can store multiple related variables together.
- Useful when we don't know ahead of time how many items we are going to store.
 - Lists solve this problem because a single list can hold from zero to practically any number of items in it.

Basic list operations

- Lists are **created** using square brackets around items separated by commas.

```
mylist = [1, 2, 3]
```

```
numbers = [-9.1, 4.77, 3.14]
```

```
fred = ["happy", "fun", "joy"]
```

- Lists are accessed using indices/positions just like strings.
- Most (but not all) string functions also exist for lists.

Strings

```
string_var = "abc123"
```

```
string_var = ""
```

```
len("abc123")
```

```
len(string_var)
```

```
string_var[p]
```

```
string_var[p:q]
```

```
str3 = str1 + str2
```

```
str3 = "abc" + "def"
```

```
"i" in "team" -> False
```

Lists

```
list_var = [item1, item2, ...]
```

```
list_var = [ ]
```

```
len([3, 5, 7, 9])
```

```
len(list_var)
```

```
list_var[p]
```

```
list_var[p:q]
```

```
list3 = list1 + list2
```

```
list3 = [1, 2, 3] + [4, 5, 6]
```

```
7 in [2, 4, 6, 8] -> False
```

One important difference

- Strings are *immutable*
 - You can't change a string without making a copy of it.

```
s = "abc"
```

```
s[0] = "A"           # illegal!
```

```
s = "A" + s[1:]     # legal
```

One important difference

- Lists are *mutable*
 - Can be changed "in-place" (without explicit copying)

```
L = [2, 4, 6, 8, 10]
```

```
L[0] = 15           # legal
```

```
L.append(26)       # legal
```


Compare mutable and immutable

- How can we switch the first and last letter in a string?
 - (Change a string to swap the first & last letters)
- How can we switch the first and last items in a list?
 - (Change a list to swap the first & last items)

Three common ways to create a list

- Create an empty list:

```
lst = []
```

Common when you're going to add things in the list later (maybe from the keyboard or a file).

- Create a list and initialize it with some data:

```
lst = [4, 7, 3, 8]
```

- Create a list of a certain length that has the same element in all positions:

```
lst = [0] * 4 # creates the list [0, 0, 0, 0]
```

Common when you need a list of a certain length ahead of time.

Simple list problems

What does this code do?

```
lst = [1, 3, 5]
```

```
lst.append(1)
```

```
lst.insert(2, 6)
```

```
lst2 = [4] * 2
```

```
lst3 = lst[0:3] + lst2
```

```
lst3[-1] = 9
```

```
x = 1
```

```
lst3[x] *= 4
```

Simple list problems

What does this code do?

```
lst = [5, 6, 3, 4, 1, 2]  
for pos in range(0, len(lst), 2):  
    lst[pos] = -lst[pos]
```

Simple list problems

- How would we write a function to convert a number from 1-12 into the corresponding month of the year as a string?

```
def getmonth(month):  
    # what goes here?
```

Example: **getmonth(2)** should return
"February"

```
num = int(input("Enter number: "))
while num != -1:
    print("Your number is", num)
    num = int(input("Enter number: "))
```

```
while True:
    num = int(input("Enter number: "))
    if num == -1:
        break
    print("Your number is", num)
```

```
lst = []
num = int(input("Enter number: "))
while num != -1:
    lst.append(num)
    num = int(input("Enter number: "))
```

```
lst = []
while True:
    num = int(input("Enter number: "))
    if num == -1:
        break
    lst.append(num)
```

- (1) Write a program to allow the user to input integers from the keyboard until they type -1. As each integer is entered, store them in a list.
- (2) After the input loop**, write a loop to calculate and print the sum of all the numbers in the list.
- (3) After the input loop**, write code to find & print the largest and smallest values in the list.
- (4) Create a file with at least 10 integers in it, one per line. Call this numbers.txt. Change the input loop in your code to read the integers from the file instead, rather than from the keyboard.
- (5) After the input loop**, write code to print out the sum of every adjacent pair of numbers in the list (don't use sliding window; use indices)
- (6) Change the input loop so the numbers are not read from a file, but are generated at random (say, between 1 and 100).