Outline Oct 3:

• Recap lists
• Functions that modify (mutate) lists
• Swapping values
• Stack diagrams with lists (+ Handout 2)

Notes

• Quiz 2 on Friday
• Ninja session TONIGHT in this room! 7-10pm
• Go through Quiz 2 study guide on paper
• Lab 5 due Saturday night
• Office Hours 3-5pm Friday (or by appointment)
Quiz practice example of making a table for things that change throughout the loop

\[
\begin{array}{c|c}
  \text{i} & \text{lst[i]} \\
  \hline
  0 & 3 \\
  1 & 4 \\
  2 & 5 \\
\end{array}
\]

\[
\text{lst} = [3, 4, 5] \\
\text{for } i \text{ in range(len(lst))}: \\
\text{print}(i, \text{lst[i]})
\]
Recap Lists
Mini-quiz (discuss with a partner)

Which are valid ways of creating `test_lst` with 5 user-entered numbers?

A) ```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
test_lst = test_lst.append(num)
```  
B) ```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
test_lst[i] = num
```  
C) ```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    num.append(test_lst)
```  
D) ```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
test_lst = test_lst + num
```  
E) ```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
test_lst = test_lst + [num]
```  
F) ```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
test_lst.append(num)
```
Mini-quiz (discuss with a partner)

Which are valid ways of creating test_lst with 5 user-entered numbers?

A)
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst.append(num)
```
NO. Append does not return anything (only mutates), so test_lst will become None and not type list.

C)
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    num.append(test_lst)
```

D)
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst + num
```

E)
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst + [num]
```

F)
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst.append(num)
```
Mini-quiz (discuss with a partner)

Which are valid ways of creating test_lst with 5 user-entered numbers?

A) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       test_lst = test_lst.append(num)
   print(test_lst)

NO. Append does not return anything (only mutates), so test_lst will become None and not type list.

B) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       test_lst[i] = num

NO. test_lst does not start out with any elements, so we cannot index into it.

C) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       num.append(test_lst)

D) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       test_lst = test_lst + num

E) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       test_lst = test_lst + [num]

F) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       test_lst.append(num)
Mini-quiz (discuss with a partner)

Which are valid ways of creating `test_lst` with 5 user-entered numbers?

A) 
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst.append(num)
```

NO. Append does not return anything (only mutates), so `test_lst` will become None and not type list.

B) 
```python
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst[i] = num
```

NO. `test_lst` does not start out with any elements, so we cannot index into it.

C) 
```python
for i in range(5):
    num = int(input("Enter a number: "))
    num.append(test_lst)
```

NO. Cannot call append on a number.

D) 
```python
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst + num
```

E) 
```python
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst[i] + [num]
```

F) 
```python
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst.append(num)
```
Mini-quiz (discuss with a partner)

Which are valid ways of creating test_lst with 5 user-entered numbers?

A) 
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst.append(num)
```

NO. Append does not return anything (only mutates), so test_lst will become None and not type list.

B) 
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst[i] = num
```

NO. test_lst does not start out with any elements, so we cannot index into it.

C) 
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    num.append(test_lst)
```

NO. Cannot call append on a number.

D) 
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst + num
```

NO. Cannot concatenate a list and a number.

E) 
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst + [num]
```

F) 
```python
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst.append(num)
```
Mini-quiz (discuss with a partner)

Which are valid ways of creating test_lst with 5 user-entered numbers?

A) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       test_lst = test_lst.append(num)`

NO. Append does not return anything (only mutates), so test_lst will become None and not type list.

B) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       test_lst[i] = num`

NO. test_lst does not start out with any elements, so we cannot index into it.

C) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       num.append(test_lst)`

NO. Cannot call append on a number.

D) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
       test_lst = test_lst + num`

NO. Cannot concatenate a list and a number.

E) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
   test_lst = test_lst + [num]`

YES. This is a classic accumulator, but is not very efficient for lists.

F) `test_lst = []
   for i in range(5):
       num = int(input("Enter a number: "))
   test_lst.append(num)`


Mini-quiz (discuss with a partner)

Which are valid ways of creating test_lst with 5 user-entered numbers?

A) 
```
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst.append(num)
```
NO. Append does not return anything (only mutates), so test_lst will become None and not type list.

B) 
```
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst[i] = num
```
NO. test_lst does not start out with any elements, so we cannot index into it.

C) 
```
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    num.append(test_lst)
```
NO. Cannot call append on a number.

D) 
```
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    num.append(test_lst)
    test_lst = test_lst + num
```
NO. Cannot concatenate a list and a number.

E) 
```
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst = test_lst + [num]
```
YES. This is a classic accumulator, but is not very efficient for lists.

F) 
```
test_lst = []
for i in range(5):
    num = int(input("Enter a number: "))
    test_lst.append(num)
```
YES. This is the best way! Uses list mutability to add an element to test_lst.
Lists as a mutable data structure

- Lists are an essential data structure, can contain basically anything (even other lists!)
- Lists are mutable (we can change their data)

```python
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```
Lists as a mutable data structure

- Lists are an essential data structure, can contain basically anything (even other lists!)
- Lists are mutable (we can change their data)

```python
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```

- We can add elements to a list

```python
lst1.append(7)
lst1
[10, 3, 1, 7]
```
Lists as a mutable data structure

- Lists are an essential data structure, can contain basically anything (even other lists!)
- Lists are mutable (we can change their data)

```python
lst1 = [5, 3, 1]
lst1[0] = 10
lst1
[10, 3, 1]
```

- We can add elements to a list

```python
lst1.append(7)
lst1
[10, 3, 1, 7]
```

- Concatenating lists

```python
lst2 = [20, 25]
lst1 + lst2
[10, 3, 1, 7, 20, 25]
```
Stack Diagrams
```python
def add_twice(x, lst):
    lst.append(x)
    lst.append(x)

def main():
    data = [1]
    add_twice(2, data)
    print(data)
    add_twice(3, data)
    print(data)

main()
```

Output:
```
[1, 2, 2]
[1, 2, 2, 3, 3]
```

Diagram:
- `main` function calls `add_twice` with arguments `1, 2, 2, 3, 3`.
- `data` list contains 1, 2, 2, 3, 3.
- Stack and heap with values 1, 2, 2, 3, 3.
See slides posted on the website

• A Better/Longer Stack Diagram Example

Swapping! (find and work with a partner)

1) Write and test the function swap in inclass/w05/shuffle_list.py

2) Ask me or a ninja for Handout 2 and draw the stack for swap

3) Then write and test the shuffle function
Handout 2 solution

```
swap
i
j
1st

main

nijas

output

["AY", "RH", "KT", "MP"]
```