

JONATHAN PARLETT

JUNE 11, 2022

**Dynamic Array Programming Example**

# WHAT WERE GOING TO DO

- Using what we've covered so far we are going to create a dynamic data structure.

# WHAT WERE GOING TO DO

- Using what we've covered so far we are going to create a dynamic data structure.
- We will create a dynamic array that grows in size automatically as we store elements in it.

# WHAT WERE GOING TO DO

- Using what we've covered so far we are going to create a dynamic data structure.
- We will create a dynamic array that grows in size automatically as we store elements in it.
- We will use a pointer to store our data.

# WHAT WERE GOING TO DO

- Using what we've covered so far we are going to create a dynamic data structure.
- We will create a dynamic array that grows in size automatically as we store elements in it.
- We will use a pointer to store our data.
- We will adjust the amount of memory allocated to this pointer in our program as the array grows in size.

- Python lists are an amazing data structure.

- Python lists are an amazing data structure.
- We can store an arbitrary amount of elements and even store elements of different types in the same list.

- Python lists are an amazing data structure.
- We can store an arbitrary amount of elements and even store elements of different types in the same list.
- Knowing what you know about memory how do you think they store so many elements?



- Python lists are an amazing data structure.
- We can store an arbitrary amount of elements and even store elements of different types in the same list.
- Knowing what you know about memory how do you think they store so many elements?
- It would be very inefficient to simply allocate space for a billion items and hope no one goes over the limit.

# PYTHON LISTS

- Python lists are an amazing data structure.
- We can store an arbitrary amount of elements and even store elements of different types in the same list.
- Knowing what you know about memory how do you think they store so many elements?
- It would be very inefficient to simply allocate space for a billion items and hope no one goes over the limit.
- Python lists grow in size as items are added to them.

# PYTHON LISTS

- Python lists are an amazing data structure.
- We can store an arbitrary amount of elements and even store elements of different types in the same list.
- Knowing what you know about memory how do you think they store so many elements?
- It would be very inefficient to simply allocate space for a billion items and hope no one goes over the limit.
- Python lists grow in size as items are added to them.
- The underlying data structure is in fact an array.

# PYTHON LISTS

- Python lists are an amazing data structure.
- We can store an arbitrary amount of elements and even store elements of different types in the same list.
- Knowing what you know about memory how do you think they store so many elements?
- It would be very inefficient to simply allocate space for a billion items and hope no one goes over the limit.
- Python lists grow in size as items are added to them.
- The underlying data structure is in fact an array.
- The size of this array is what is adjusted as items are added.

## THINGS TO KEEP IN MIND

- Try to relate what we are doing to the concepts we've previously discussed.

## THINGS TO KEEP IN MIND

- Try to relate what we are doing to the concepts we've previously discussed.
- Dynamic vs Static memory. As we write try to notice where in the program we are using dynamic, and static memory.

## THINGS TO KEEP IN MIND

- Try to relate what we are doing to the concepts we've previously discussed.
- Dynamic vs Static memory. As we write try to notice where in the program we are using dynamic, and static memory.
- Stack and Heap. As we write the program try to think about what information we are storing on the Heap and what we are storing on the stack.

## THINGS TO KEEP IN MIND

- Try to relate what we are doing to the concepts we've previously discussed.
- Dynamic vs Static memory. As we write try to notice where in the program we are using dynamic, and static memory.
- Stack and Heap. As we write the program try to think about what information we are storing on the Heap and what we are storing on the stack.
- Memory model. If a structure looks confusing try to use the model we showed previously to visualize how it might be stored.



## THINGS TO KEEP IN MIND

- Try to relate what we are doing to the concepts we've previously discussed.
- Dynamic vs Static memory. As we write try to notice where in the program we are using dynamic, and static memory.
- Stack and Heap. As we write the program try to think about what information we are storing on the Heap and what we are storing on the stack.
- Memory model. If a structure looks confusing try to use the model we showed previously to visualize how it might be stored.
- Everyone has their own style of writing and thinking. This is just mine. You don't have to do things exactly as I do them.

## THINGS TO KEEP IN MIND

- Try to relate what we are doing to the concepts we've previously discussed.
- Dynamic vs Static memory. As we write try to notice where in the program we are using dynamic, and static memory.
- Stack and Heap. As we write the program try to think about what information we are storing on the Heap and what we are storing on the stack.
- Memory model. If a structure looks confusing try to use the model we showed previously to visualize how it might be stored.
- Everyone has their own style of writing and thinking. This is just mine. You don't have to do things exactly as I do them.
- Lets get started.