COMP1021
Introduction to Computer Science

Stacks

David Rossiter and Gibson Lam

Outcomes

- After completing this presentation, you are expected to be able to:
  1. Explain the stack structure and its push and pop operations
  2. Implement a stack using Python code

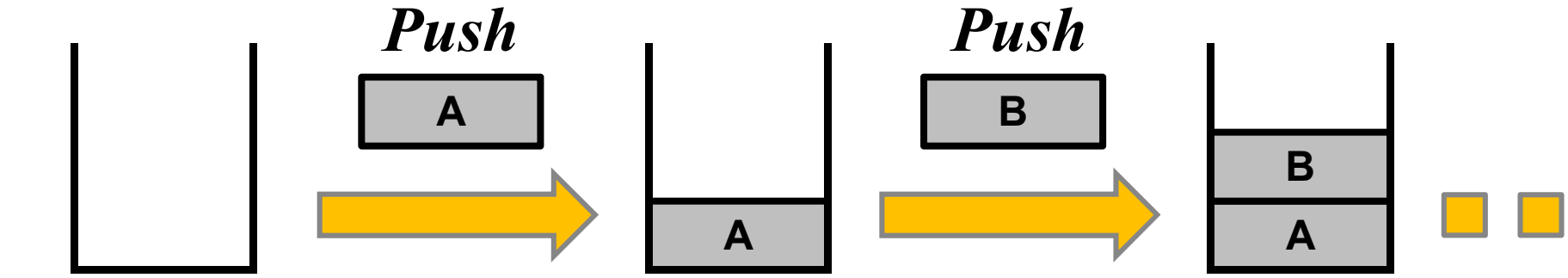
What is a Stack?

- A stack is a special kind of data structure
- It is used to store a collection of things
- Sometimes it is a called a *Last In First Out* (LIFO) structure, because of the way it works

Stack Operations

- There are two operations for a stack:
  - Push
    - It adds a new item to the top of the stack
  - Pop
    - It takes the top item off the stack

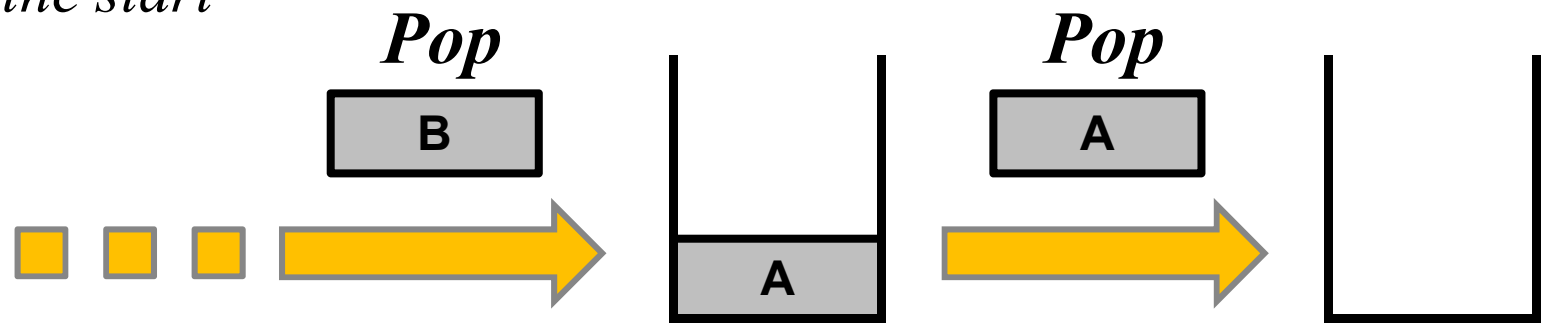
An Illustration of a Stack



Let's have an empty stack at the start

A is put at the top of the stack

B is put at the top of the stack



B is removed from the top of the stack

A is removed from the top of the stack

Using the Stack Idea in Python

- Any Python list can use the stack idea
- There are two built-in functions for a Python list:
  - `list.append()` (this is the Python word for 'push')
  - `list.pop()`
- The *append* operation is used to add a new item to the end of the list (= the top of the stack)
- The *pop* operation is used to remove and return the last element in the list (= the top of the stack)

Some Simple Code Which Uses a Stack

```
# Create a list with two numbers in it
all_numbers = [11, 22];
print("The numbers in the list are", all_numbers)

all_numbers.append(33)
all_numbers.append(44)
all_numbers.append(55)
print("Now, the numbers in the list are", all_numbers)

all_numbers.pop()
all_numbers.pop()
all_numbers.pop()
all_numbers.pop()
print("Now, the numbers in the list are", all_numbers)
```

Running the Program

- Here is the display after running the program:

Three new items have been pushed onto the stack

```
The numbers in the list are [11, 22]
Now, the numbers in the list are [11, 22, 33, 44, 55]
Now, the numbers in the list are [11]
>>>
```

Four items have been popped from the stack

Using pop()

- Note that there are several ways to use `pop()`:

`list.pop()`	– takes the top item off the list and throws it away
`x = list.pop()`	– takes the top item off the list and puts it in x
`list.pop(2)`	– takes the third item out of the list and throws it away
`x = list.pop(2)`	– takes the third item out of the list and puts it in x