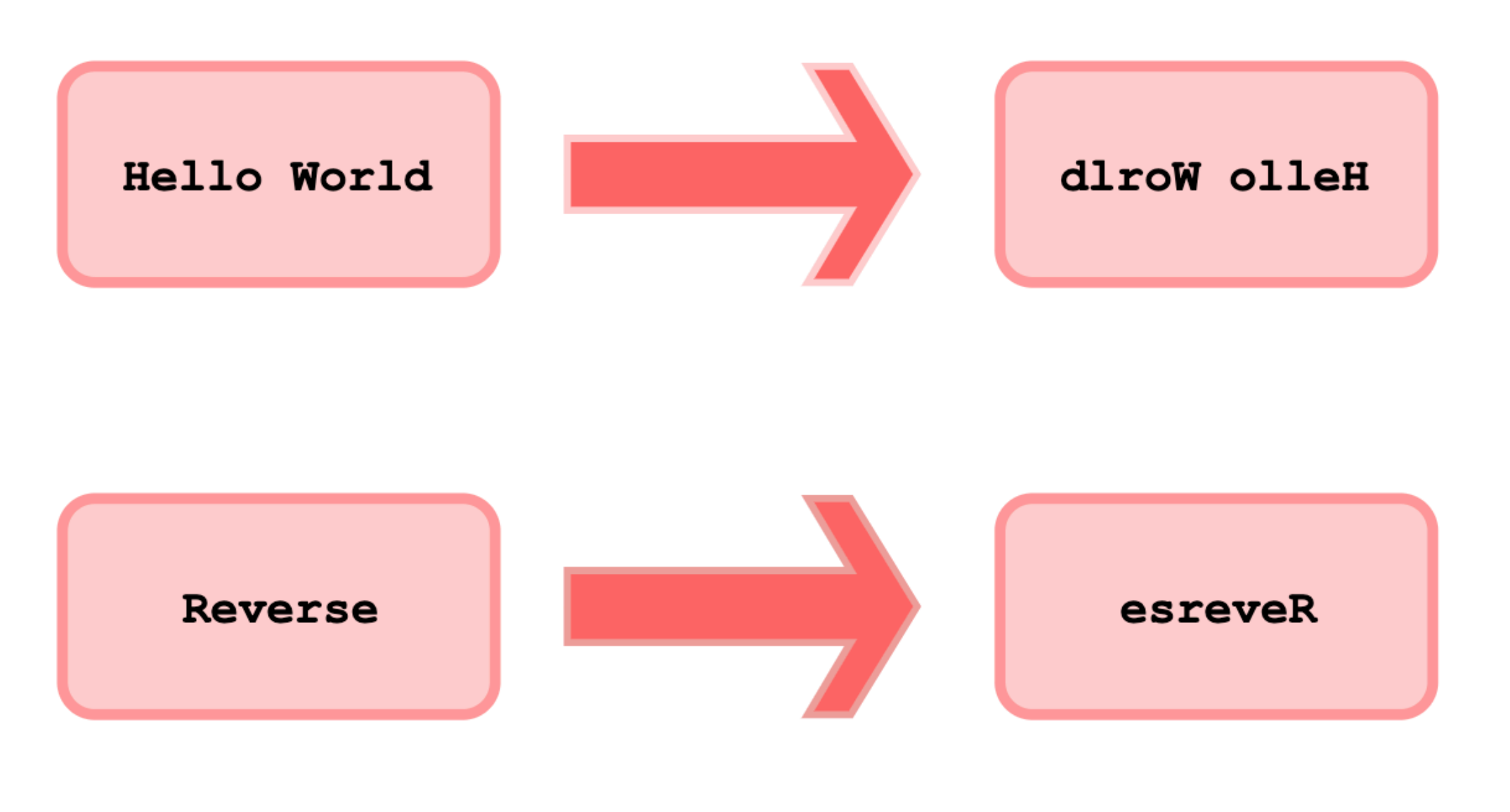
What does reversing a string mean?

When given a string, we can do multiple actions with it, one of this being-reversing a string. **Reversing** means to take the string and flip it:-all the letters in the front- go to the back and vice versa.

The illustration below shows you exactly how to do that!



Reverse a string

> 💡 **Note**
>
> *Keep in mind that a string can be one word or multiple words.*

Implementing the Code

The code snippet below shows how to reverse a string using recursion!

First, let's take a look at the code, and then we can move on to its explanation.

Try the code by changing the values of `string1` and `string2` to see how it works with other strings!

```
1  class ExampleClass {
2  
3      public static String reverseString(String myStr) {
4          // Base case
5          if (myStr.isEmpty()) {
6              return myStr;
7          }
8  
9          // Recursive case
10         else {
11             return reverseString(myStr.substring(1)) + myStr.charAt(0);
12         }
13     }
14 
15     public static void main( String args[] ) {
16         String string1 = "Hello World";
17         String string2 = "Reverse";
18 
19         System.out.println( "The Original String is: ");
20         System.out.println(string1);
21 
22         String resultStr = reverseString(string1);
23 
24         System.out.println("String after reversal: ");
25         System.out.println(resultStr);
26     }
27 
28 }
```

Run   Save   Reset

Understanding the Code

The recursive code can be broken down into **two parts**: the **recursive method** and the **main** where the method is called.

Driver Method

The recursive method is called within the driver function. Lets first look at what that function does-from **line 16 to 25**.

- The main method initializes two **String** variables, `string1` and `string2`.
- The main method *displays* the string, passes it to the `reverseString` *method*, and then *displays* it to show the changes.

Recursive Method

In a recursive method, there are two parts: the **base case** and the **recursive case**.

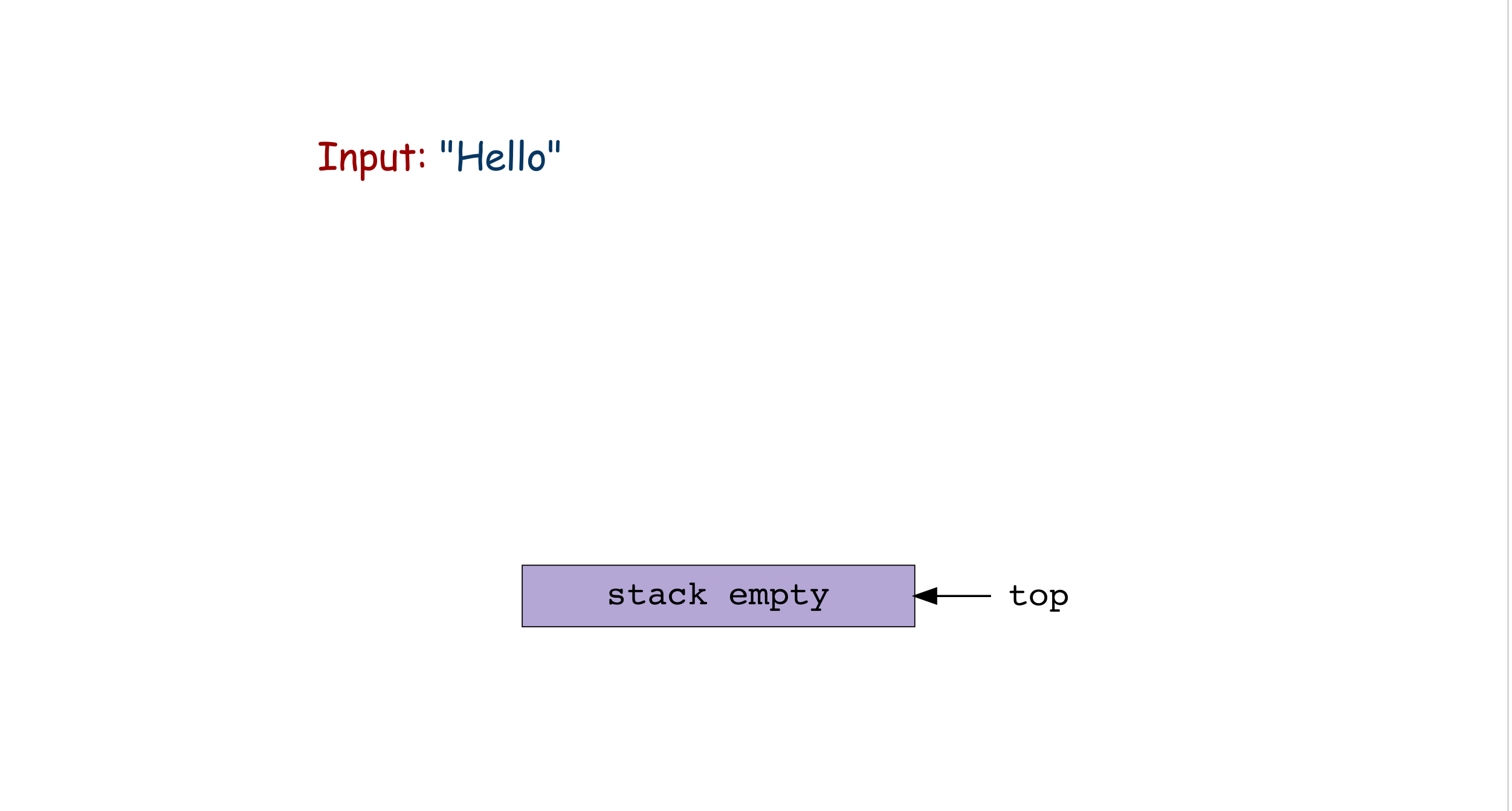
Base Case

- The *if condition* from **line 5 to line 7** checks to see if the string is empty. It then returns the string if this condition is true. This is where the method terminates. This enables us to know the point where the recursion will stop.

Recursive Case

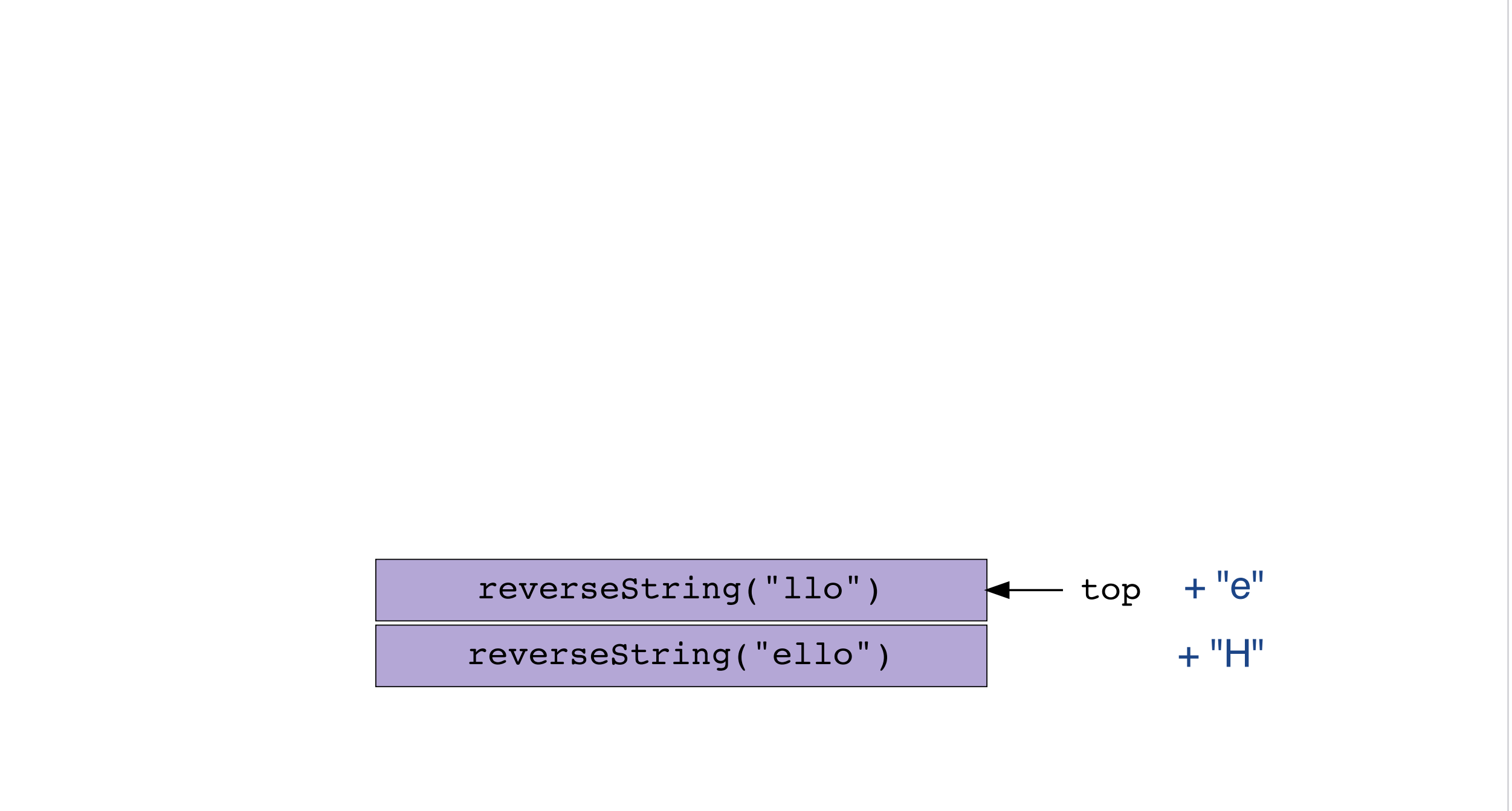
- The method takes in one argument: the string.
- When the recursive method is called, it takes the substring of the string from the 1st index using `substring(1)` until the end of that string. It then adds the 0th index using `charAt(0)` at the end of that string. This process is called repeatedly through the recursive method, eventually returning the string in its reversed form.
- For example, if the string is **Hello** when the function is first called, it will only take **ello** and then pass it to the recursive method. The **H** is added from what is returned using the recursive method. This process is repeated until the base case is reached and the string is fully reversed.

Understanding through a Stack

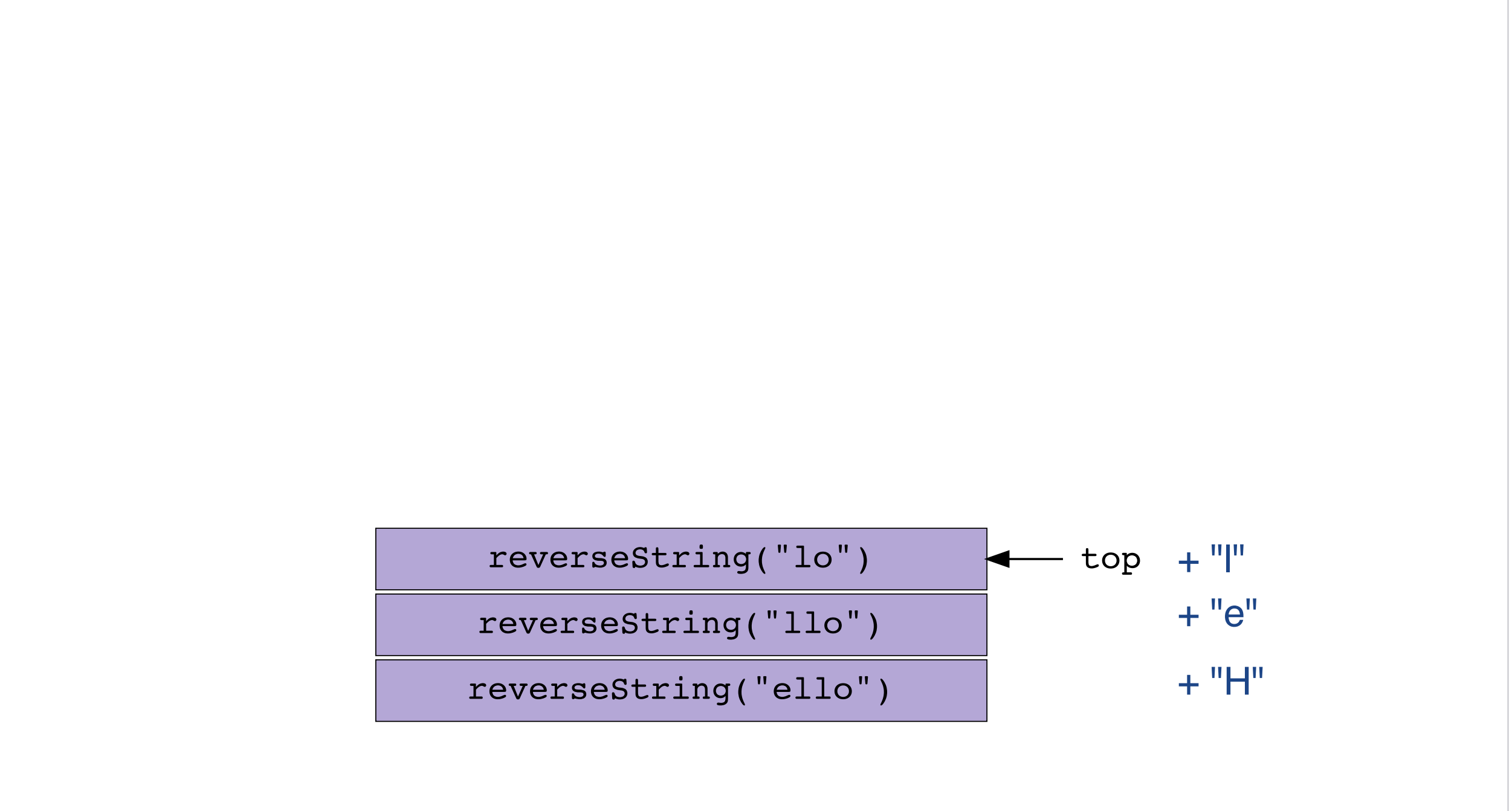


1 of 13

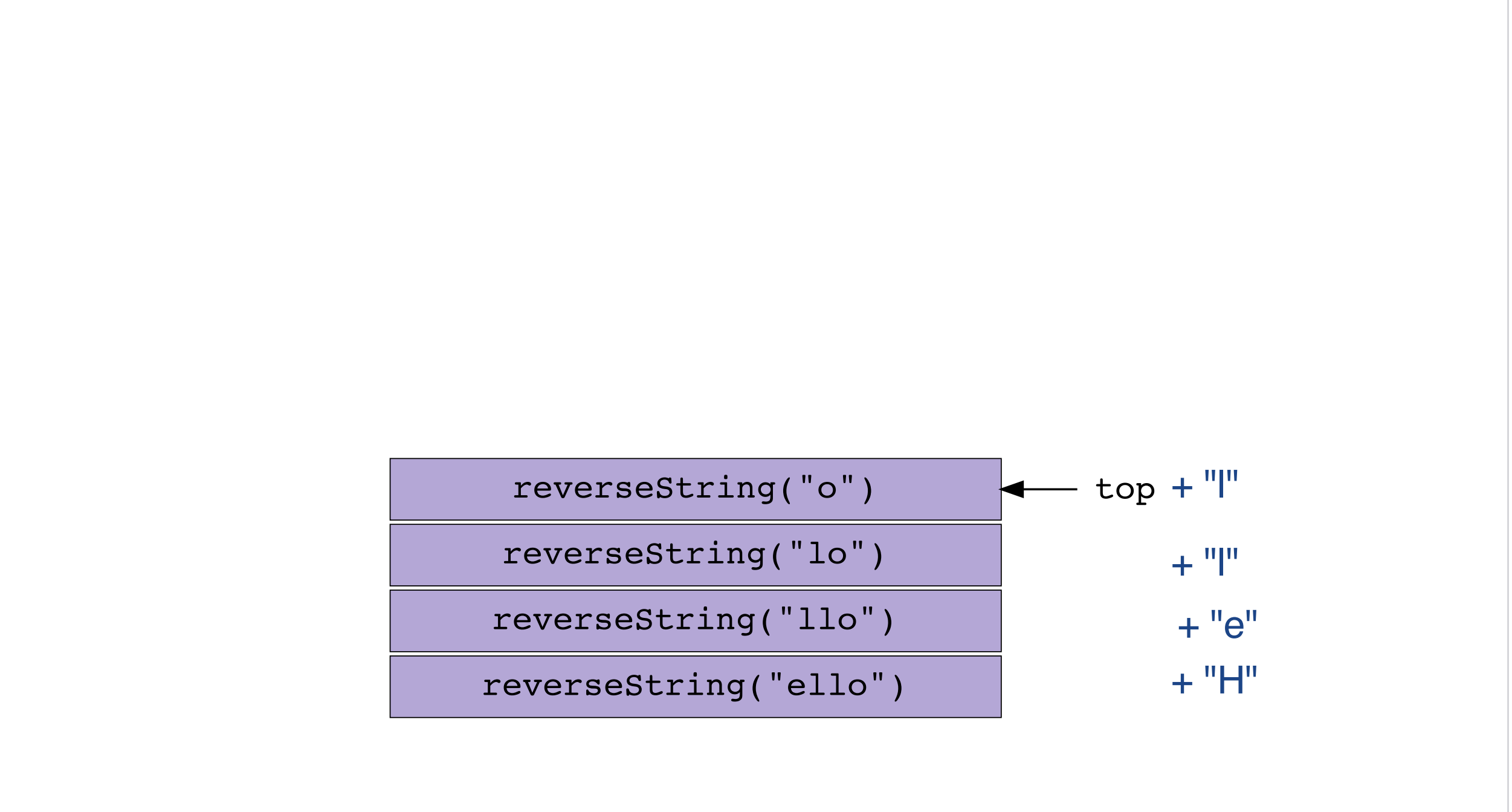
2 of 13



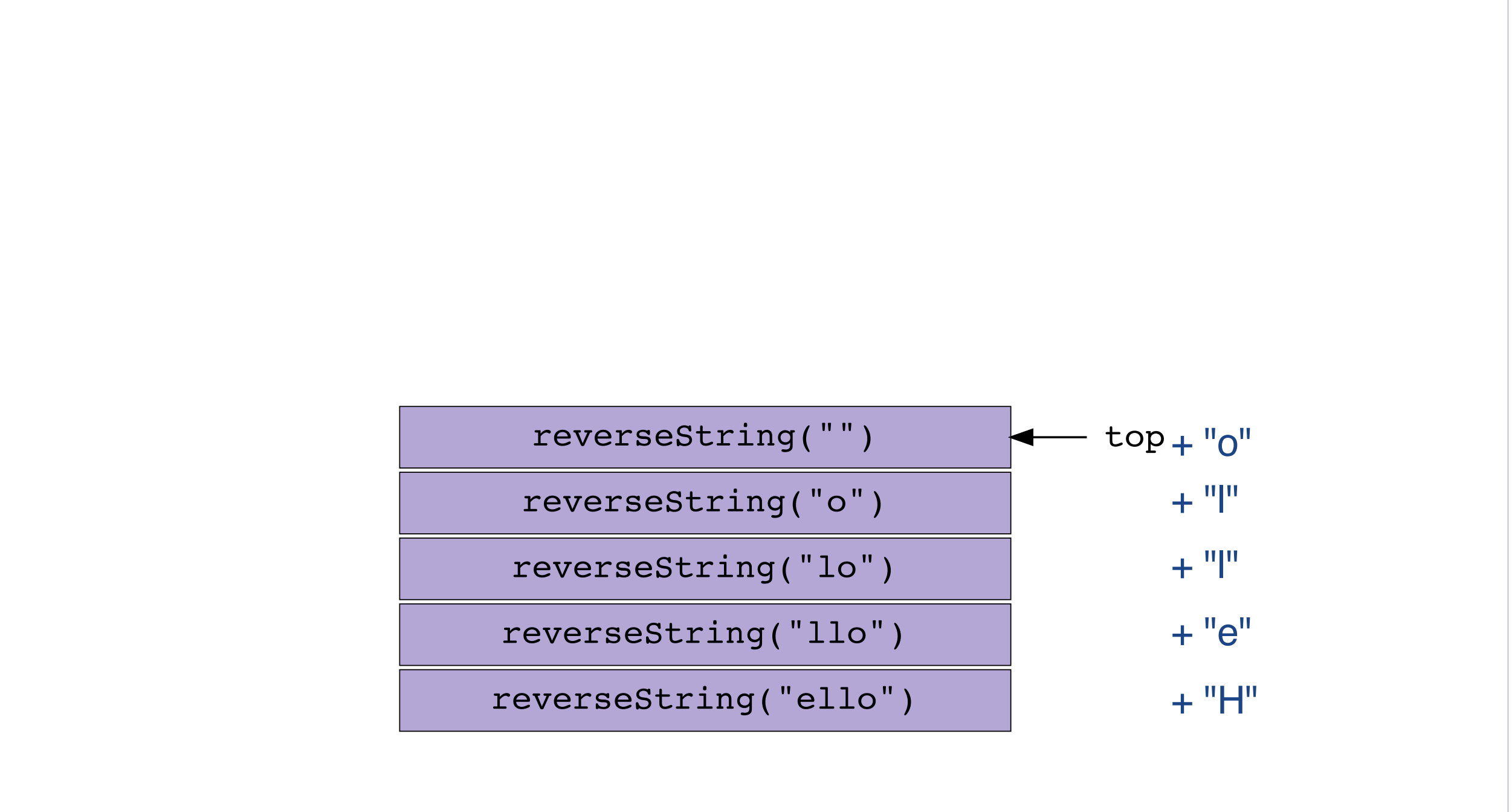
3 of 13



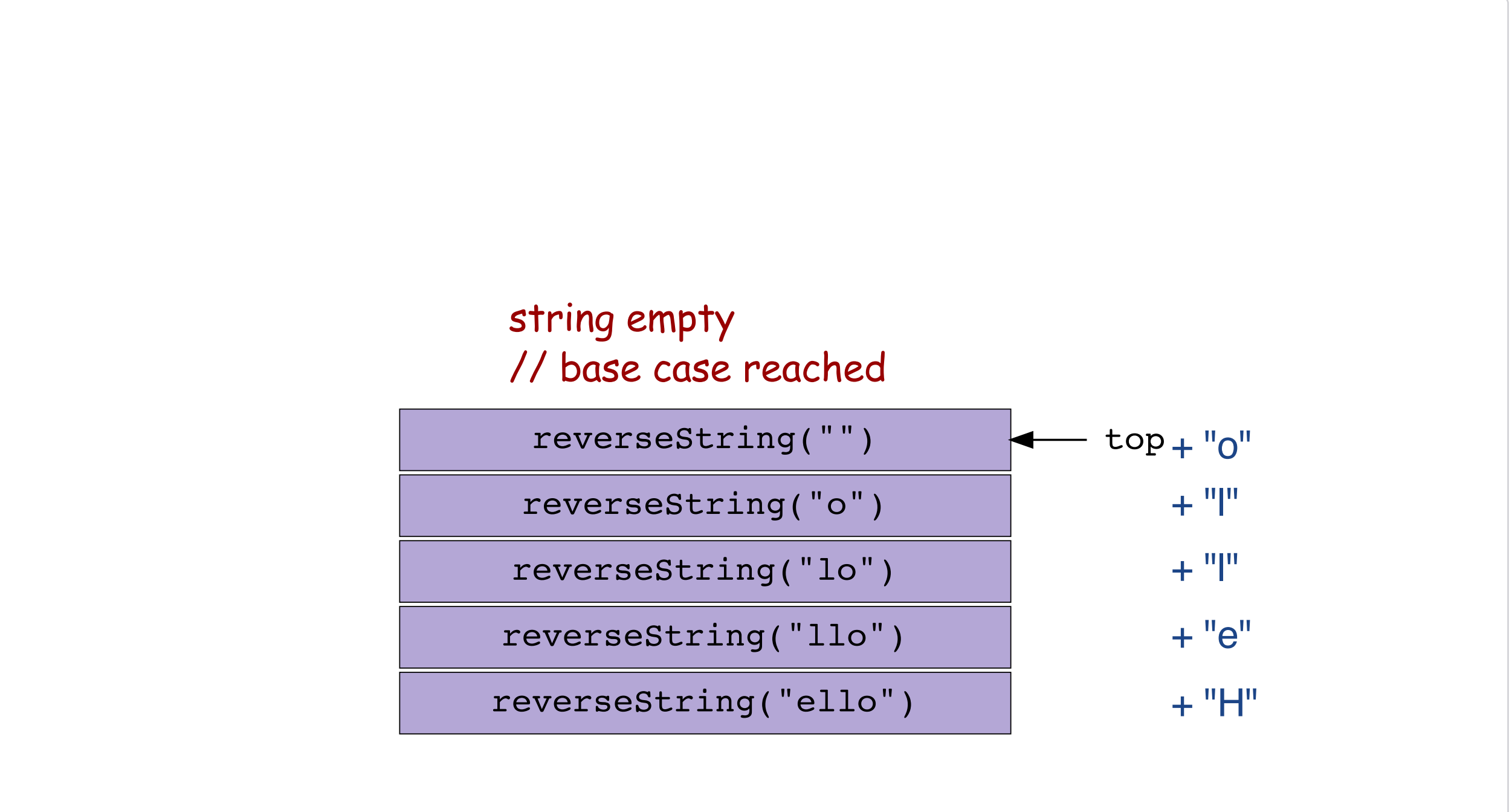
4 of 13



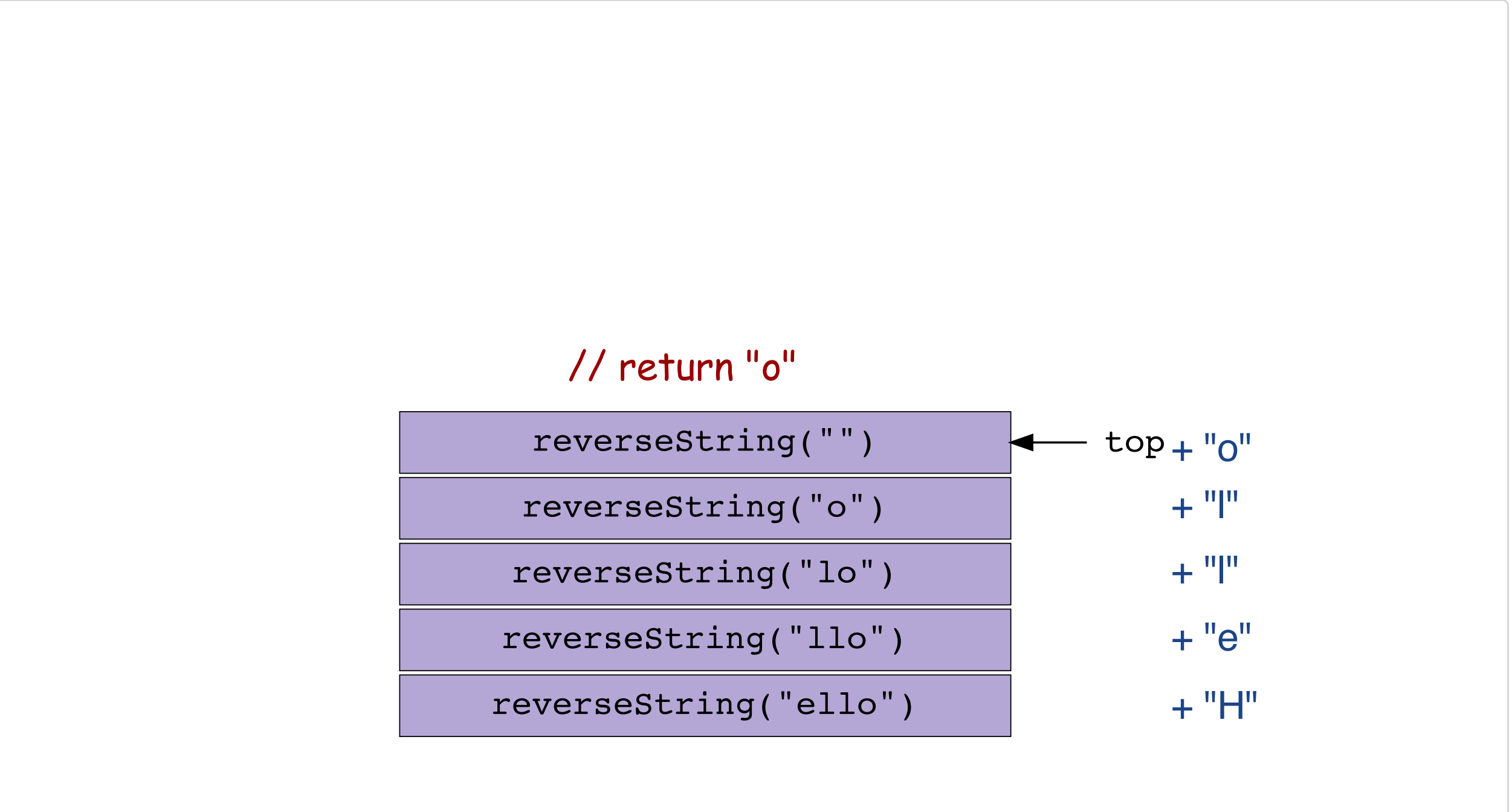
5 of 13



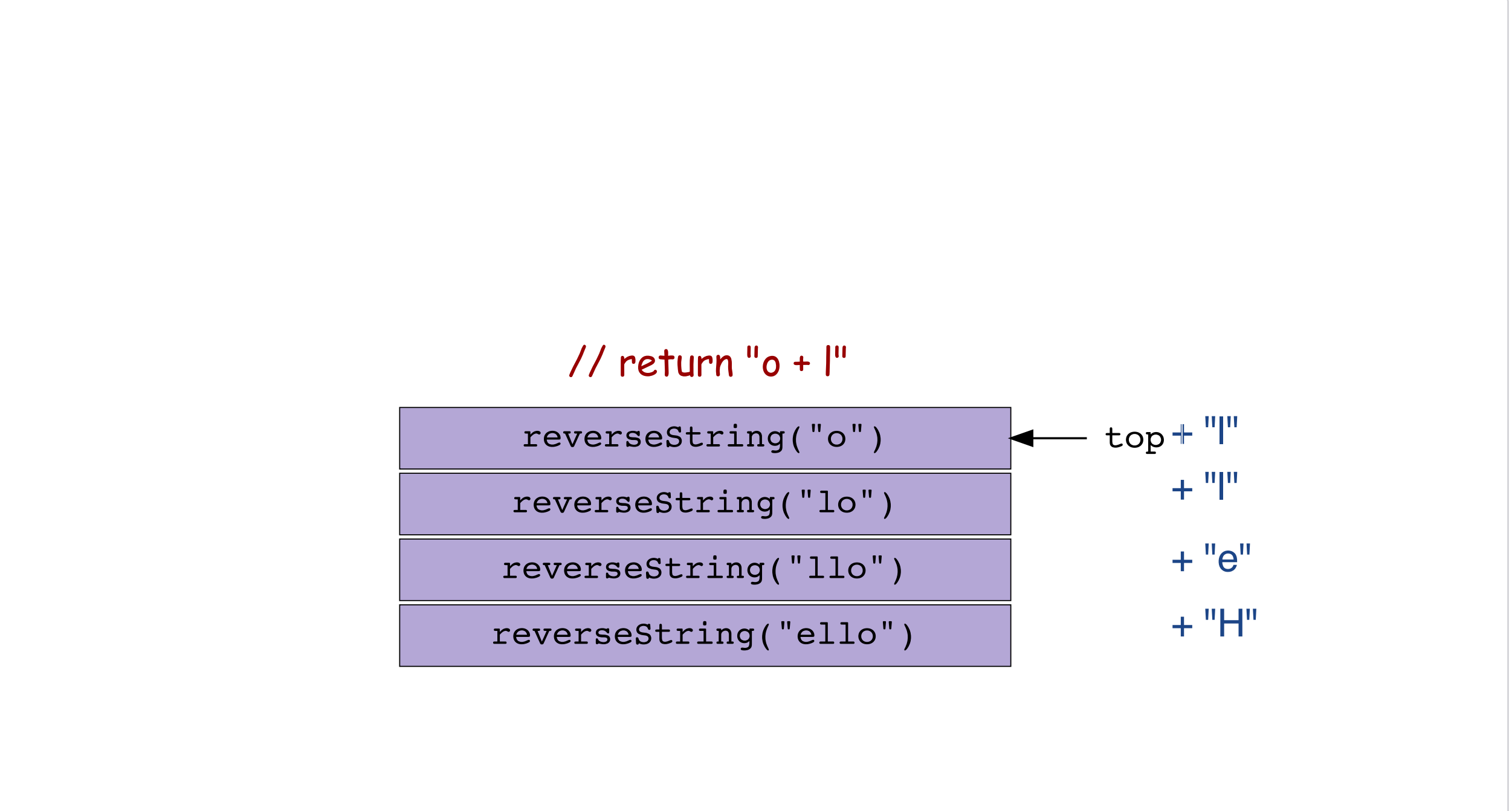
6 of 13



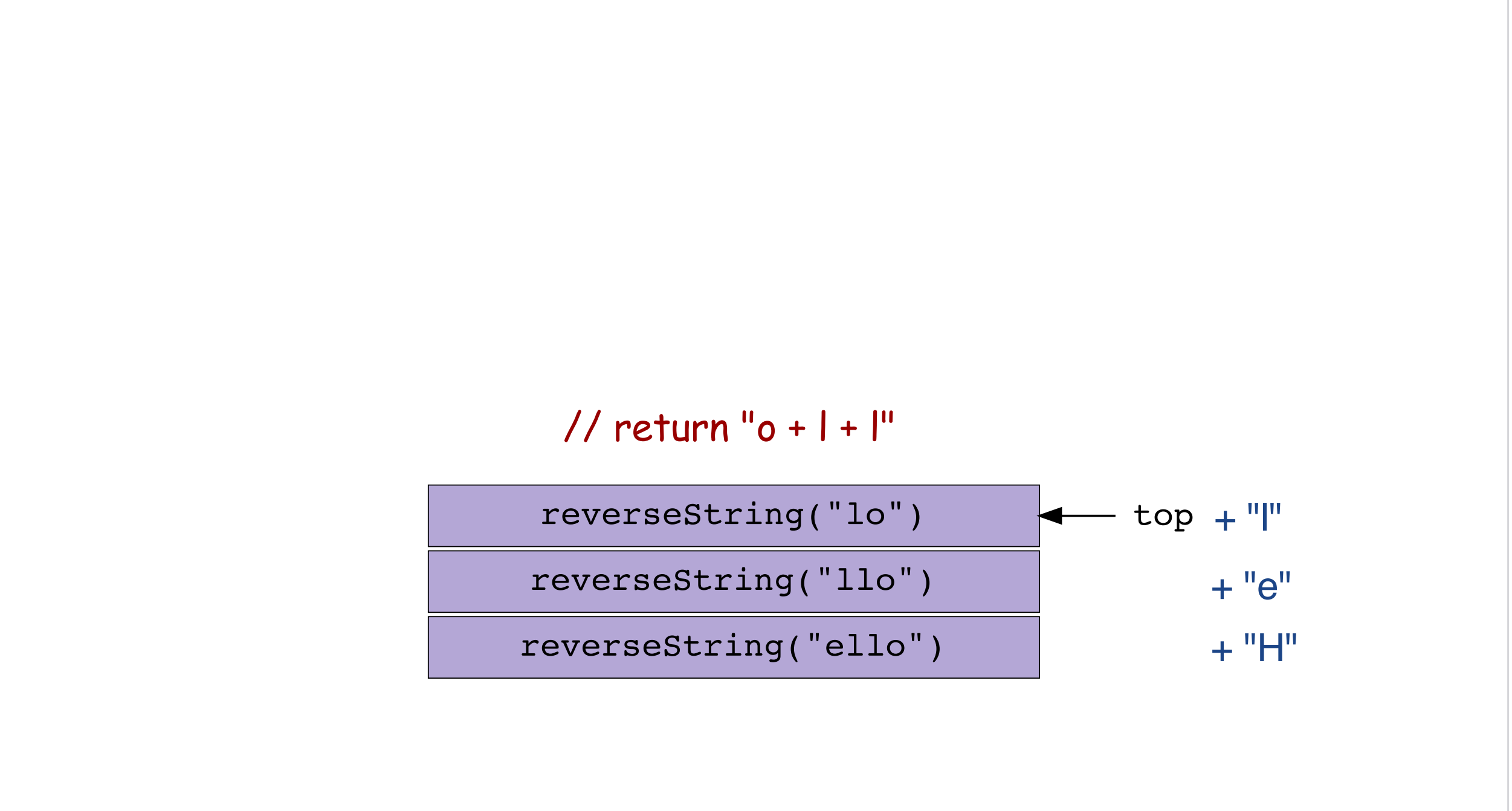
7 of 13



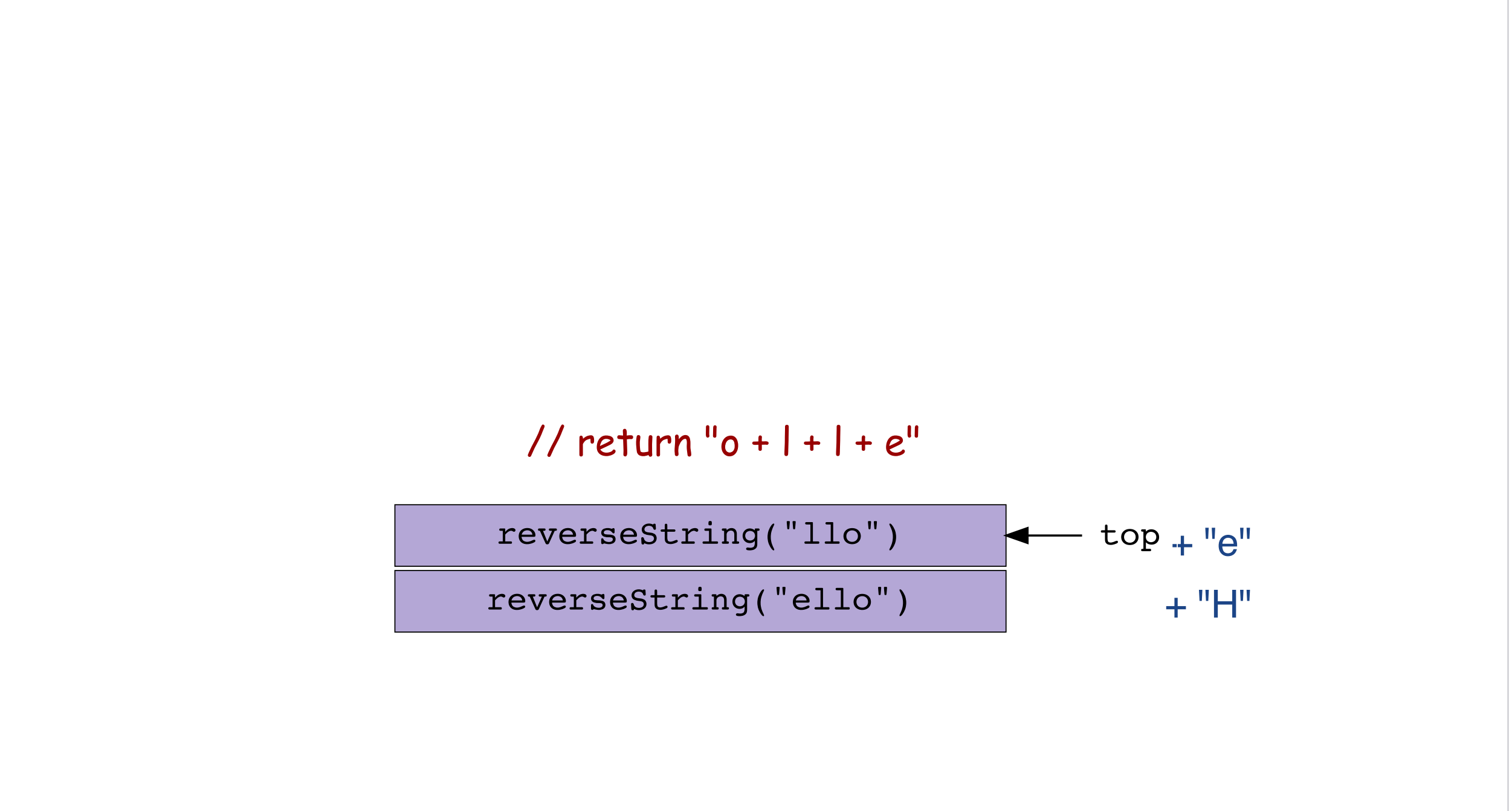
8 of 13



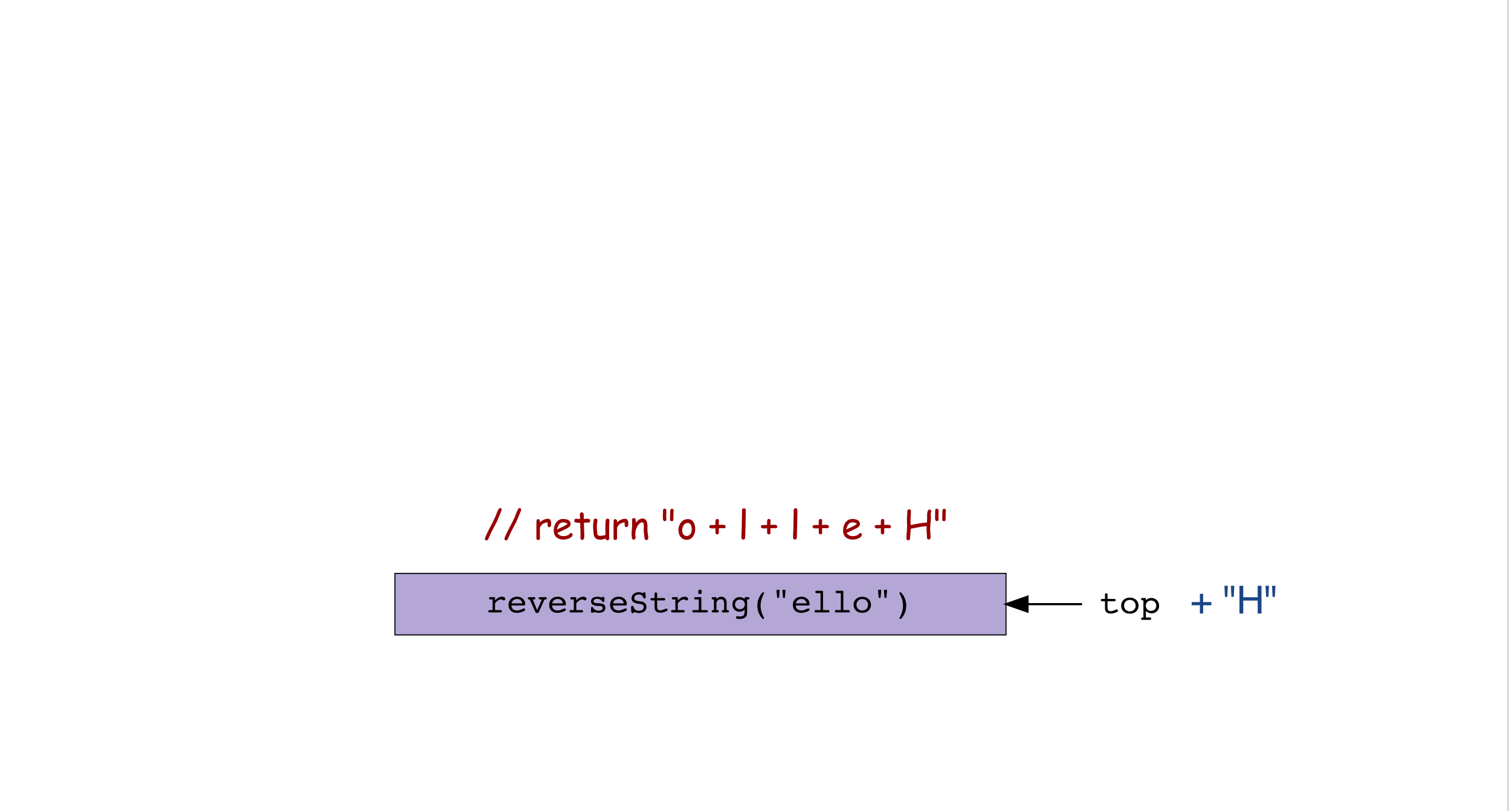
9 of 13



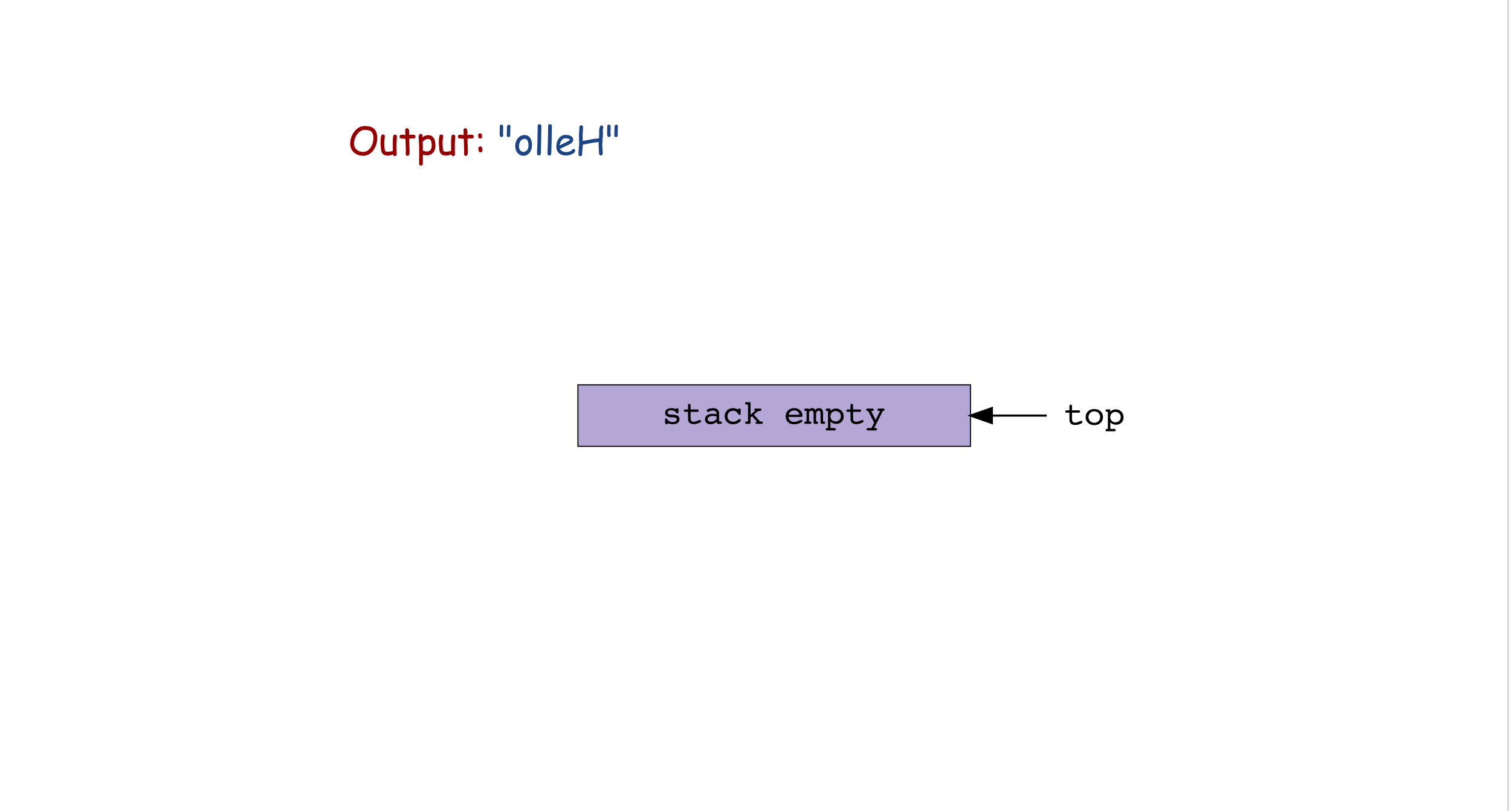
10 of 13



11 of 13



12 of 13



13 of 13