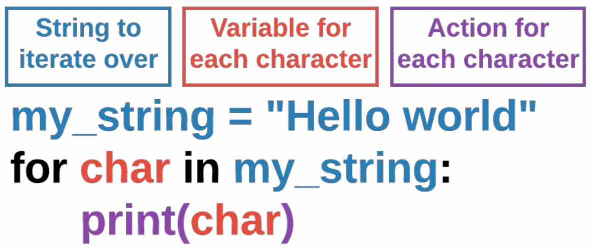
Learning Objectives - String Iteration

* **Define string iteration**
* **Identify two ways to iterate over a string**
* **Explain the inner workings of string iteration**

Iteration - For Loop

Iterating Over Strings

You have seen how you can make a copy of individual characters in a string with their index. Iterating over a string allows you to deal with each character of a string individually. You start with the character at index 0 and move through the end of the string.


String to iterate over
Variable for each character
Action for each character
my_string = "Hello world"
for char in my_string:
print(char)

String Iteration

```
my_string = "Hello world"
for char in my_string:
    print(char)
```

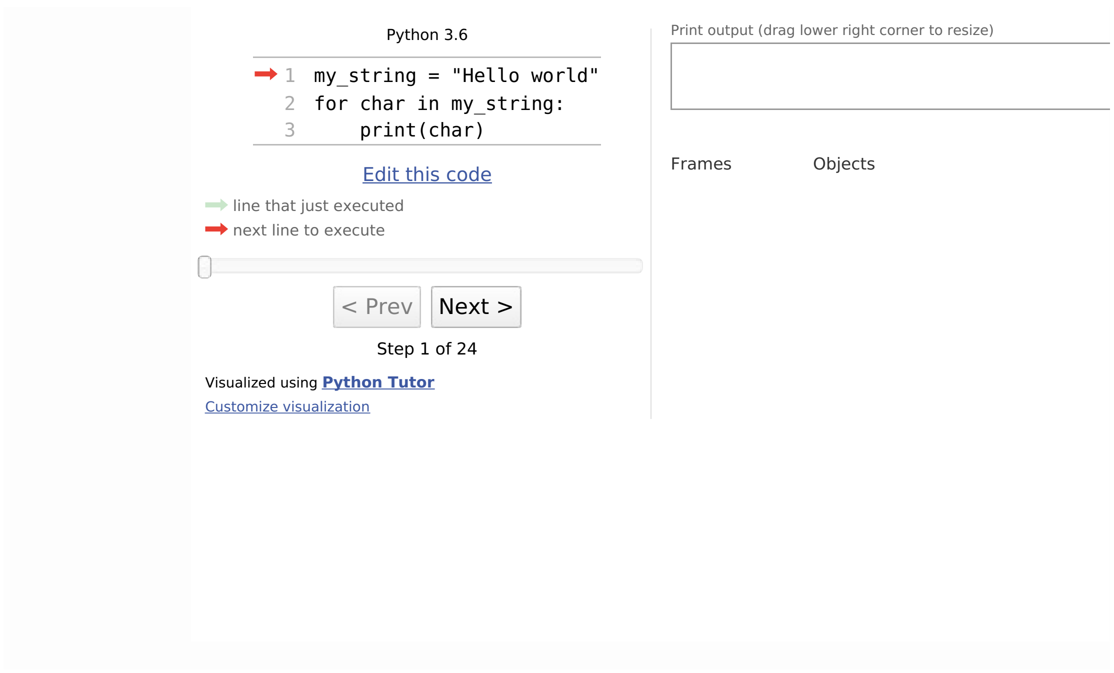
challenge

What happens if you:

- Change the value of `my_string` to `"10, 11, 12, 13, 14"`?
- Change the value of `my_string` to `"\u25A3\u25A8\u25D3\u25CC\u25A2"`?
- Change the `print` statement to `print(my_string)`?

Behind the Scenes

Use the code visualizer below and step through the code. Notice how the variable `char` is the value of the character. The index of the string is never referenced.



```
my_string = "Hello world"
for char in my_string:
    print(char)
```

Edit this code
➡ line that just executed
➡ next line to execute
< Prev
Next >
Step 1 of 24
Visualized using Python Tutor
Customize visualization
Print output (drag lower right corner to resize)
Frames
Objects

Iteration - While Loop

While Loop

String iteration is most often done with a for loop. However, a while can be used as well.

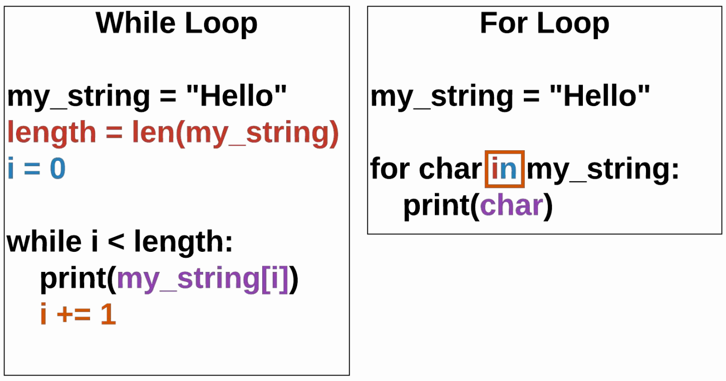
```
my_string = "Calvin and Hobbes"
length = len(my_string)
i = 0

while i < length:
    print(my_string[i])
    i += 1
```

challenge

What happens if you:

- Change the loop to `while i <= length:`?
- Change the `print` statement to `print(i)`?
- Remove `i += 1`?

Comparing While & For Loops



```
my_string = "Hello"
length = len(my_string)
i = 0

while i < length:
    print(my_string[i])
    i += 1
```

```
my_string = "Hello"

for char in my_string:
    print(char)
```

Compare While & For Loops

The for loop is more efficient than a while loop when iterating over a string. You do not need to declare variables for the length of the string (red text), declare a variable for the index of the string (blue text), or increment the index variable (orange text). All of this is handled by the `in` statement. In for loops, you can use the iteration variable to reference the string character. With a while loop, however, you need to use the string and index to reference the character (purple text).

Formative Assessment 1

Formative Assessment 2