Lesson 10: For Loops

Using For Loops with Lists

Remember the last exercise from Lesson 9.

Example 1

Write a function to find value 20 in the list, and if it is present, replace it with 200.

Using while loops, we solved the problem like so:

```
def change_20_into_200(nums: list):
    i = 0
    while i < len(nums):
        if nums[i] == 20:
            nums[i] = 200
        i += 1
    return nums
```

For loops allow us to iterate through lists without a counter:

```
def change_20_into_200(nums: list):
    for num in nums:
        if num == 20:
            num = 200
```

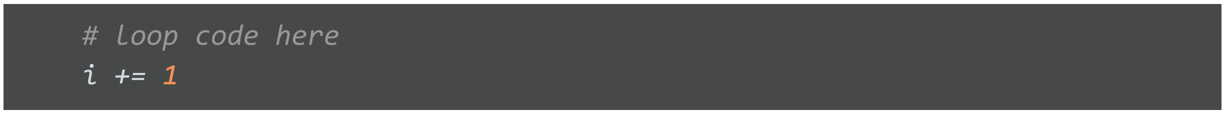
1. Notice that we don't have to do any **i += 1** stuff here.

For Loops vs For Each Loops

If you've programmed in other languages, you may notice that for loops in python look different than for loops in Java, Javascript, C++, etc.. That's because the conventional "for loop" doesn't actually exist in Python. A for loop in Python is the same as a "for each loop" in other languages, meaning that rather than counting, they iterate through each element in some set. If you'd like to do counting in Python, you can either use a **while loop** or a for loop together with the **range()** function:

```
for i in range(10):
    # loop code here

i = 0
while i < 10:
```



```
    # loop code here
    i += 1
```

More List Exercises with For Loops

Exercise 1

1) Create a function that counts the number of odd values in a list of integers.
2) Create a function that prints all the odd values in a list as a list

Exercise 2

Create a function that returns the sum of integers in a list.

Exercise 3

Write a function that returns the highest value in a list of integers.