

for Each Index

Indexes *are not* needed to *read* list elements.

```
names = ["emma", "liam", "aisha", "mateo", "sofia", "ravi"]
for name in names:
    print(f"Hello, {name}!")
```

Indexes *are* needed to *change* list elements.

```
for i in range(len(names)):
    name = names[i]
    # replace element at index i
    names[i] = name.capitalize()
```

Questions (20 min)

Start time:

1. Based on the above code:

- a) What is the length of the names list?
- b) What sequence of values does `range(len(names))` represent?
- c) What word does the variable `i` stand for?

2. Run the code, and describe what each loop does.

3. Compare the two loops in this model.

- a) Why does the first loop name the variable `name`?
- b) Why does the second loop name the variable `i`?

4. This code attempts to make each name in `names` ALL CAPS. Why doesn't it work?

```
for name in names:
    name = name.upper()
```

5. The built-in `enumerate()` function can be used to shorten the code in this model. Run the following two lines in a shell, and record the output.

a) `enumerate(names)` # Create an enumerate object

b) `list(enumerate(names))` # Convert it to a list

6. Describe in your own words what an `enumerate` object contains.

The second loop from before:

```
for i in range(len(names)):
    name = names[i]
    # replace element at index i
    names[i] = name.capitalize()
```

The same loop, using enumerate:

```
for i, name in enumerate(names):
    # replace element at index i
    names[i] = name.capitalize()
```

7. Write a `for` loop that adds an "!" to each string in the names list.

8. Write a `for` loop that increases all prices by 5%. (*Hint: Multiply by 1.05*)

`prices = [19.99, 6.34, 1.00, 12.79, 2.50]`