

```
colors = ["red", "green", "blue", "purple"]
i = 0
while i < len(colors):
    print(colors[i])
    i += 1
```

## range of length

I often see new Python programmers attempt to recreate traditional `for` loops in a slightly more creative fashion in Python:

```
colors = ["red", "green", "blue", "purple"]
for i in range(len(colors)):
    print(colors[i])
```

This first creates a range corresponding to the indexes in our list (0 to `len(colors) - 1`). We can loop over this range using Python's for-in loop (really a [foreach](#)).

Because we don't actually care about the indexes in our loop, there is a **much simpler method of looping** we can use:

```
colors = ["red", "green", "blue", "purple"]
for color in colors:
    print(color)
```

So instead of retrieving the item indexes and looking up each element, we can just loop over our list using a plain for-in loop.

## range of length

We could use `range(len(our_list))` and then lookup the index like before:

```
presidents = ["Washington", "Adams", "Jefferson", "Madison", "Monroe", "Adams", "Jackson"]
for i in range(len(presidents)):
    print("President {}: {}".format(i + 1, presidents[i]))
```

But there's a more idiomatic way to accomplish this task: use the `enumerate` function.