Successive Values

```python
def syr_list(n):
    seq = [n]
    while n != 1:
        n = syracuse(n)
        seq.append(n)
    return seq
```

- Not "wrong"

- Limited to always creating a sequence

Generator Functions

- Use the **yield** statement

- Act as an iterable

- Can be used in a **for** statement

  - Almost anywhere that a collection is used

- Won't create big list objects

- Lazy — they don't compute anything unless forced to by another function consuming results

  - `list(some_function())`

  - Builds a list object by consuming values from the generator