

Validating input:

Will keep prompting till the input gives the valid input

Other Uses:

Accumulator Pattern

```
def sumTo():
```

```
    the sum = 0
    aNbr = 0
```

```
    while the sum < 2:
```

```
        the sum = the sum + 1/2 aNbr
        aNbr = aNbr + 1
```

```
    return the sum
```

Iterations:

↳ used for definite iteration

```
ex: for i in range(5):
    print(i)
```

↳ While statement: (You can write any for loops as a while loop)

→ used for indefinite iteration (repeats until condition is met)

→ risk of never ending (infinite loop).

Examples:

```
while n > 0:
    print(n)
```

example:

```
def sumTo(upperLimit):
```

```
    sum = 0
    nextNumber = 1
```

} setting the initializations

```
    while nextNumber <= upperLimit:
```

```
        sum = sum + nextNumber
```

```
        nextNumber = nextNumber + 1
```

```
    return sum
```

Sentinel Values

special values that signal termination of an input or a loop.

ex: entering 0 to stop input

ex: if price != 0 → sentinel value

else:
 moreItems = False

→ print sequence of 3n+1 until it reaches 1

3n + 1:

```
def seq3np1(n):
```

```
    while n != 1:
```

```
        print(n) → if it's 1 just print n.
```

```
        if n % 2 == 0: -
```

```
            n = n // 2
```

```
        else:
```

```
            n = 3 * n + 1
```

```
        print(n) → when n != 1
```

eg seq3np1(6)

5, 16, 8, 4, 2, 1