While Loops

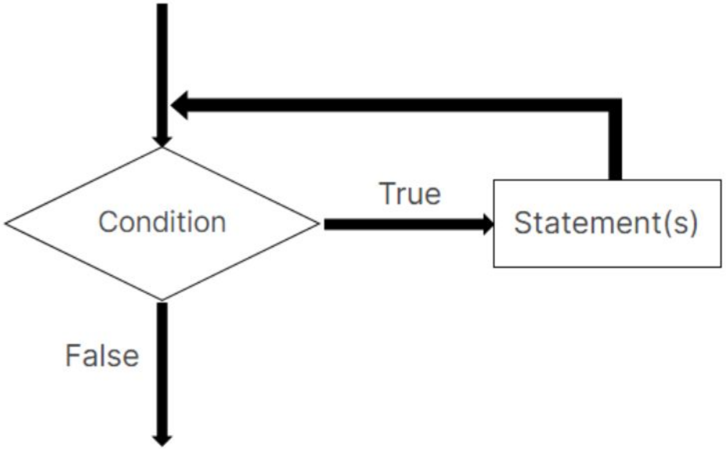
Formal definition of a while loop

Used to repeat code blocks when the number of repetitions is unknown ahead of time

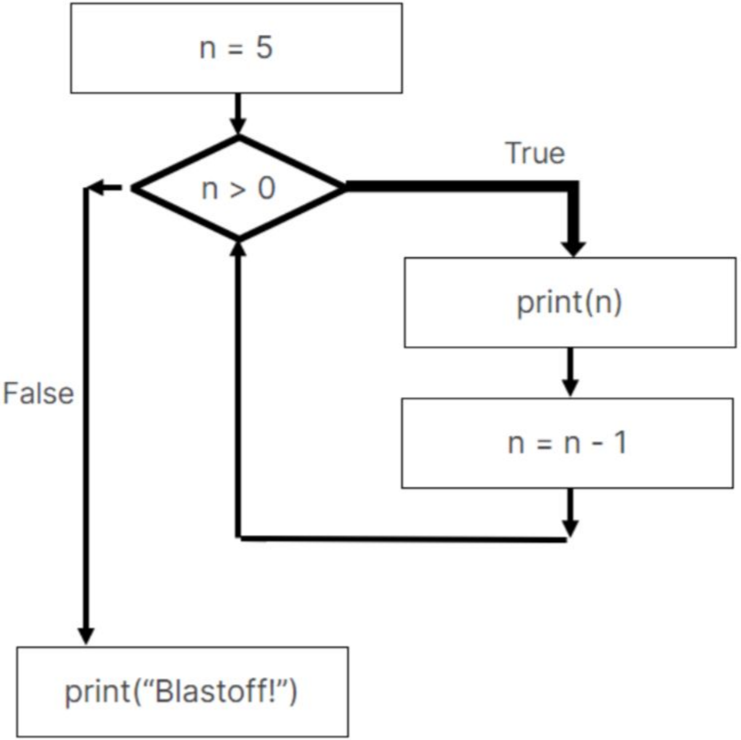
while a condition is true, do some task

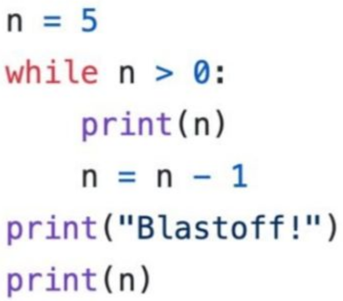
Like conditional statements, loops control program flow

Loop control keywords: **break** and **continue**



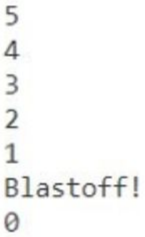
While loop (repeated steps)





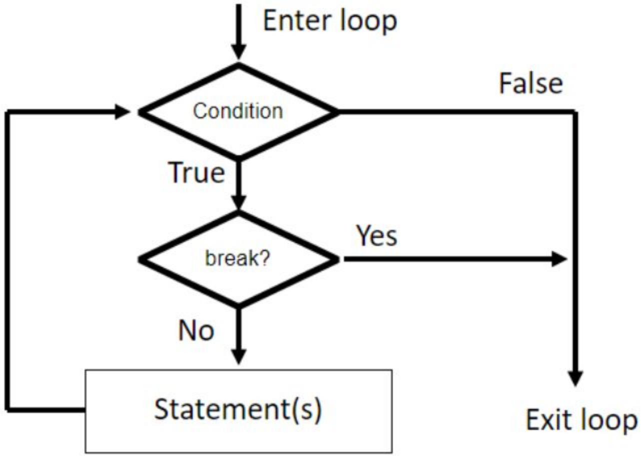
```python
n = 5
while n > 0:
    print(n)
    n = n - 1
print("Blastoff!")
print(n)
```

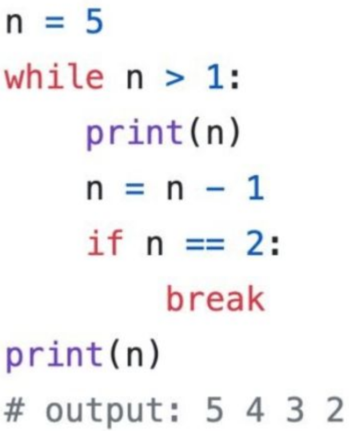
Output:



```
5
4
3
2
1
Blastoff!
0
```

The break statement



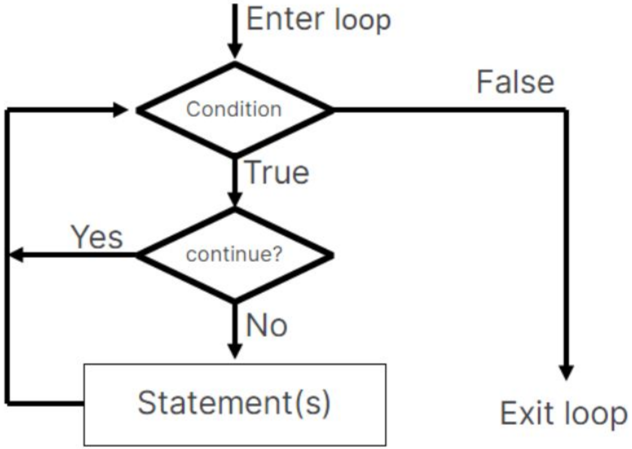


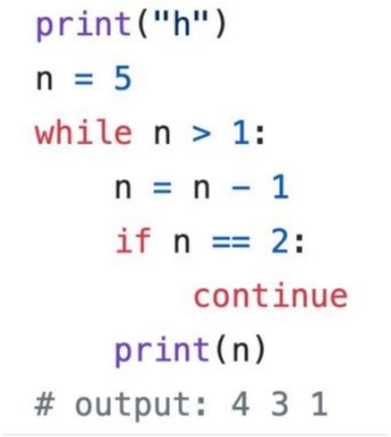
```
n = 5
while n > 1:
    print(n)
    n = n - 1
    if n == 2:
        break
print(n)
# output: 5 4 3 2
```

Terminates current loop, resumes execution at next statement

Most commonly used when a condition is met
that requires immediate exit

The continue statement





```
print("h")
n = 5
while n > 1:
    n = n - 1
    if n == 2:
        continue
    print(n)
# output: 4 3 1
```

Skips all remaining statements in current loop iteration, continues normal execution at next iteration

Very useful for skipping specific iteration(s) but keep looping to the end

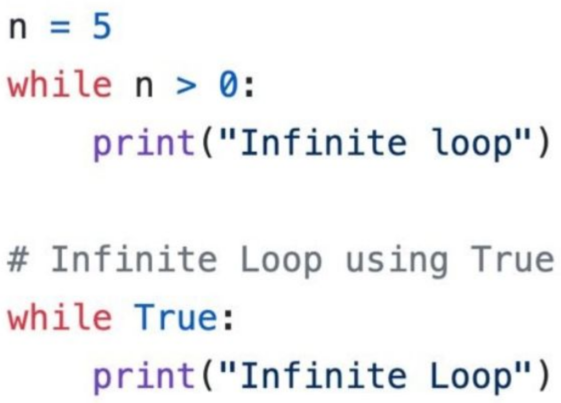
Infinite loops

DON'T run the code to the right on your computer. It will create an infinite loop that causes your computer to stall

Infinite loops run forever

If you write one by mistake, your program could hang until forcefully terminated or the computer rebooted

Infinite loops can be used to grab input from a user until the input is an exit command that conditionally executes a break statement



```
n = 5
while n > 0:
    print("Infinite loop")

# Infinite Loop using True
while True:
    print("Infinite Loop")
```