The *While* instruction

- We saw ‘for’ to iterate over a sequence

```
writers = [”Hemingway", ”Dickens", ”King"]
for x in writers :
       print(x)
```

- *With the while loop we can execute a set of statements only as long as a given condition is true*

While loop

- With the while loop we can execute a set of statements as long as a condition is true.

```
i = 1
while i < 6:
  print(i)
  i += 1
```

While loop

- With the break statement we can stop the loop even if the while condition is true:

```
i = 1
while i < 6:
  print(i)
  if i == 3:
    break
  i += 1
```

While loop

- We can have a while loop using an indefinite **True** condition

```
while True:       #  or while(True)
  num = int(input('enter an integer: '))
  if  num == 3:
    break
```

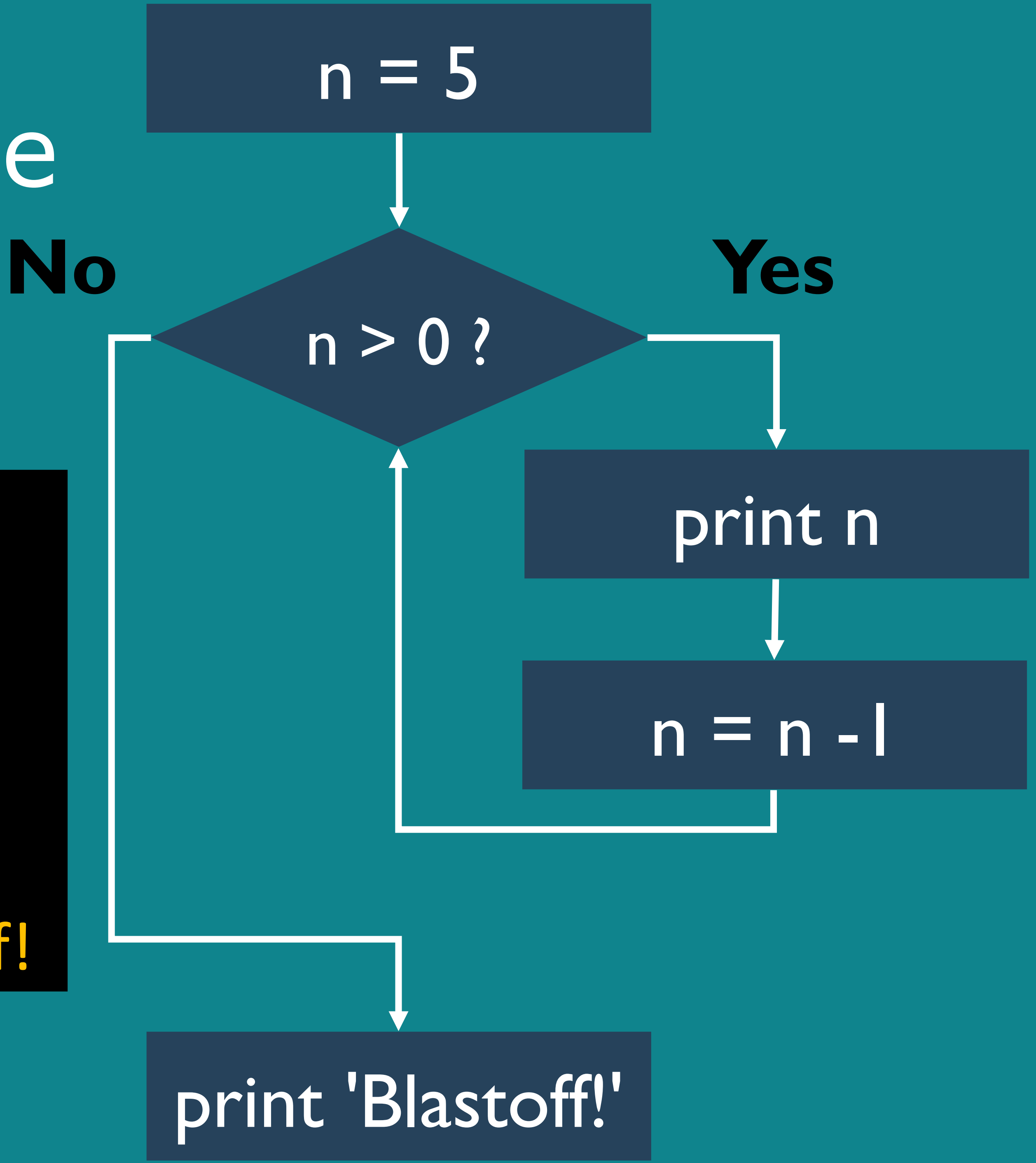
```
# True, not true...
```

While loop example

- What is this algo. doing?

```
n = 5
while n > 0 :
    print(n)
    n = n-1   #(n-=1)
print('Blastoff!')
```

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



Breaking from while

- The break statement ends the current loop and jumps out of to the while immediately
  - We use break to stop, when a condition is met

```
while(True):
    num = int(input("Enter a number, or 0 to exit"))
    print(num)
    if num == 0:
        print("...break!")
        break
```

Breaking from while

- The break statement ends the current loop and jumps to the statement immediately following the loop

```
while(True):
    num = int(input("Enter a number, or 0 to exit"))
    print(num)
    if num == 0:
        print("...break!")
        break
```

Quiz

- What does it print?

```
i = 1
while i<6:
  i=i+1
  if i == 4:
    break
  print(i)
```

Quiz

- What does it print?

```python
i = 1
while i<6:
  i=i+1
  if i == 4:
    break
  print(i)
```

```
2
3
```

Quiz

- What does it print?

```
i = 1
while i<6:
  print(i)
  if i == 4:
    break
  i=i+1
```

Quiz

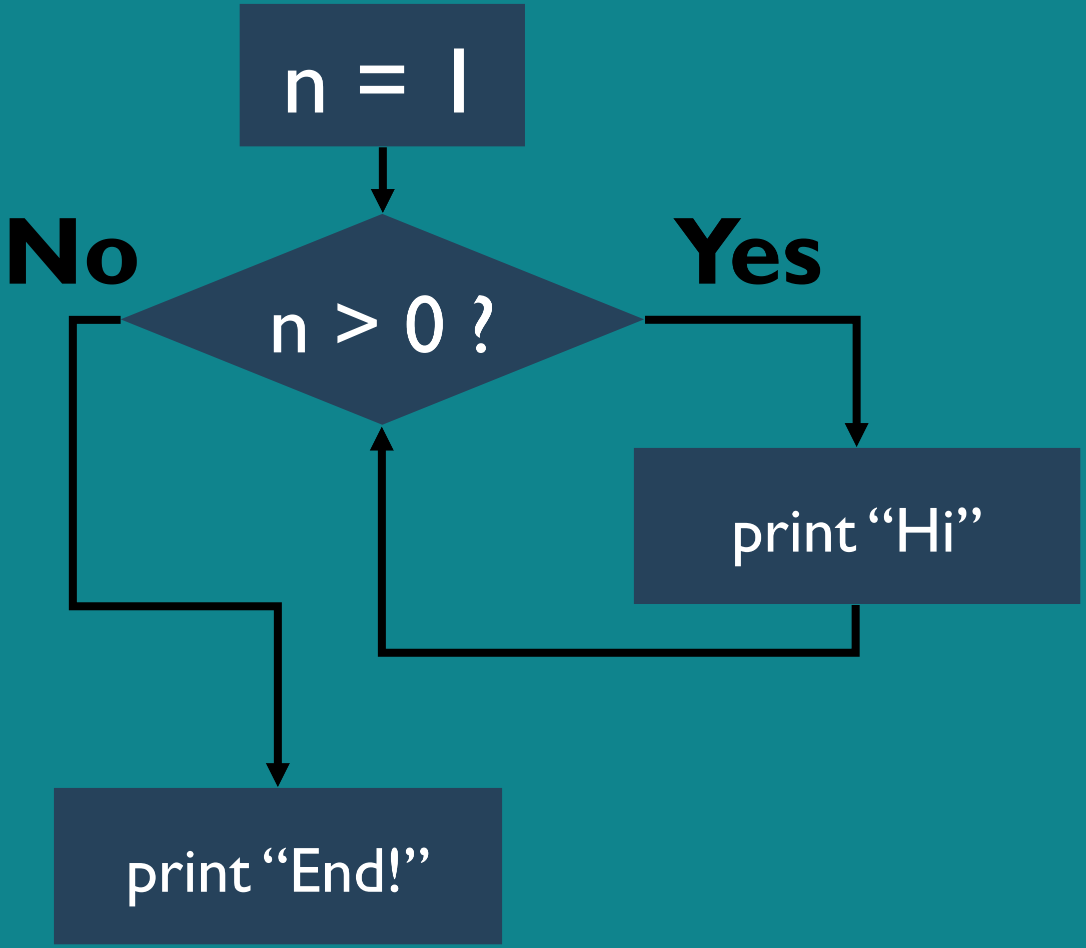
- What does it print?

```python
i = 1
while i<6:
  print(i)
  if i == 4:
    break
  i=i+1 # We always increase at the end
```

```python
i=i+1
# OR
i+=1
```

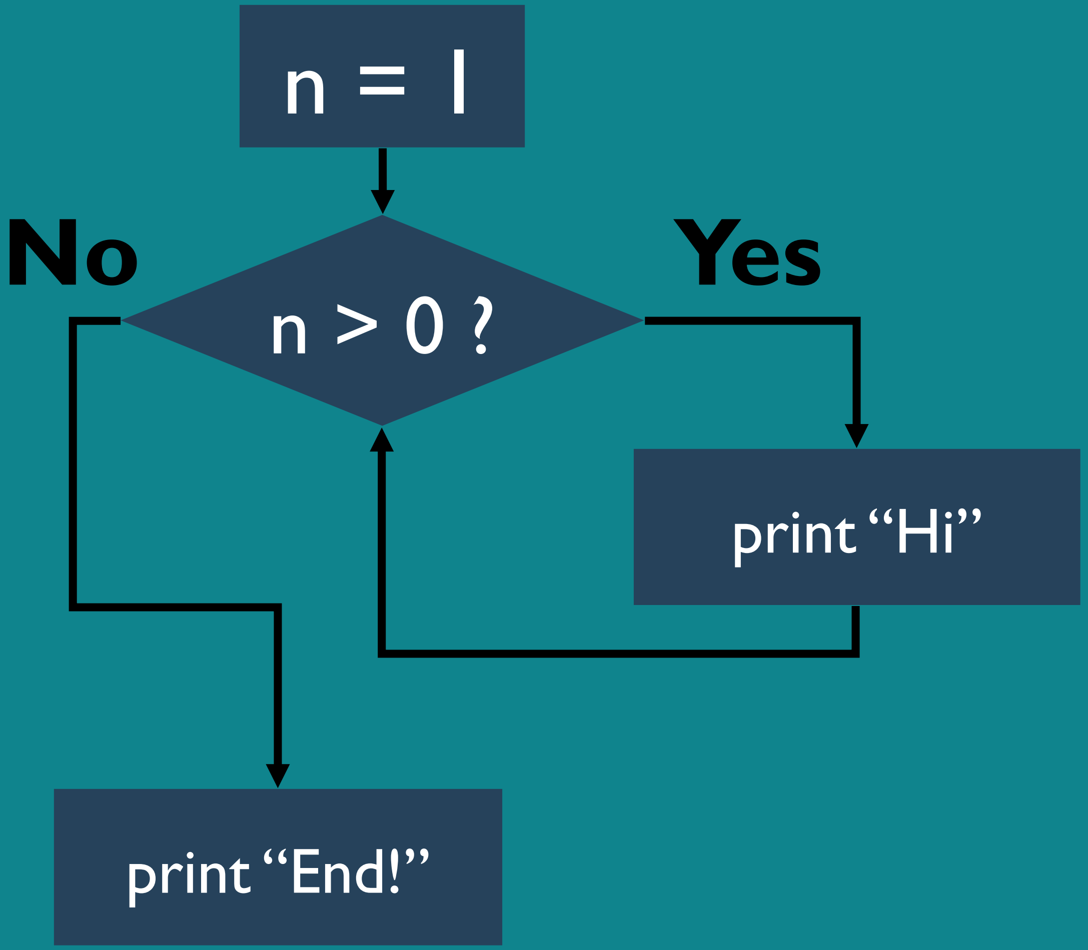
```
1
2
3
4
```

Quiz: What’s wrong with this loop?



```
n = 1
while n > 0 :
    print(“Hi”)
print(“End”)
```

Quiz: What's wrong with this loop?



```
n = 1
while n > 0 :
    print("Hi")
print("End")
```

This loop will never terminate!

Indefinite loops

- “While” loops are called "indefinite loops" because they keep going until a logical condition becomes False
- The loops we have seen so far are pretty easy to examine to see whether they will terminate or become "infinite"
- Sometimes it is a little harder to make sure that a loop will terminate
  - We need to create conditions to check all possible scenarios
  - We need to do it ourselves, and ahead of actually running the code. Why?

3 minutes with hardcore Computer Science

Alan Turing’s “Halt” theorem: deciding whether a code containing loops will ever end is not possible in general

Click here for a simple proof (YT video)

Rice’s theorem: all non-trivial properties of a code can be mapped to Turing’s “Halt” problem (see above)

Consequence: checking that a code terminates in reasonable time, as well as other properties cannot really be automated

No automated Software Engineering is possible: you have to do it yourself.

Click here to see what a “Turing machine” is

Indefinite loops

- Quite often we have a list of items, effectively a finite set of things
- We can use a **for** loop to iterate over a list of data.
- Those are called definite loops because will execute an exact number of times
- We say that:
  - definite loops iterate through the members of a set

For vs. While loop

- The for statement is used to iterate over the elements of a sequence.

- The while loop tells the computer to do something as long as the condition is met. Its construct consists of a block of code and a condition.

While in lists

- For loops:
  - Iterate for each element in a list
- While there are elements:
  - Iterate

```python
alist = [1, 3, 7, 9]
for anum in alist:
    print(anum)
```

```
1
3
7
9
```

```python
alist = [1, 3, 7, 9]
count = 0 # Represents the index
while(True):
    print(alist[count])
    if count == len(alist)-1:
        print("end")
        break
    count = count + 1
```

```
1
3
7
9
```

Quiz!

- A while loop in Python is used for what type of iteration?
  - Definite
  - Indeterminate
  - Indefinite
  - Discriminant

Quiz!

- A while loop in Python is used for what type of iteration?
  - Definite
  - Indeterminate
  - **Indefinite**
  - Discriminant