

while loops.

Week 4 | Lecture 1 (4.1)

if nothing else, write `#cleancode`

This Week's Content

- **Lecture 4.1**
 - while loops
 - Reading: Chapter 9
- **Lecture 4.2**
 - More while loops, survey, midterm review
 - Reading: Chapter 9
- **Lecture 4.3**
 - Midterm review

Looping (Iterating)

- Looping means repeating something over and over until a particular condition is satisfied.
- Looping (aka iteration) is the second key control structure in programming (if-statements/branching was the first).

Looping (Iterating)

- Looping means repeating something over and over until a particular condition is satisfied.

Email ←

**Send
Promotional
Email**

Looping



**List of
Customers**

Looping (Iterating)

- Looping means repeating something over and over until a particular condition is satisfied.

Yes/No ←

Does the
Tweet
contain
#cleancode

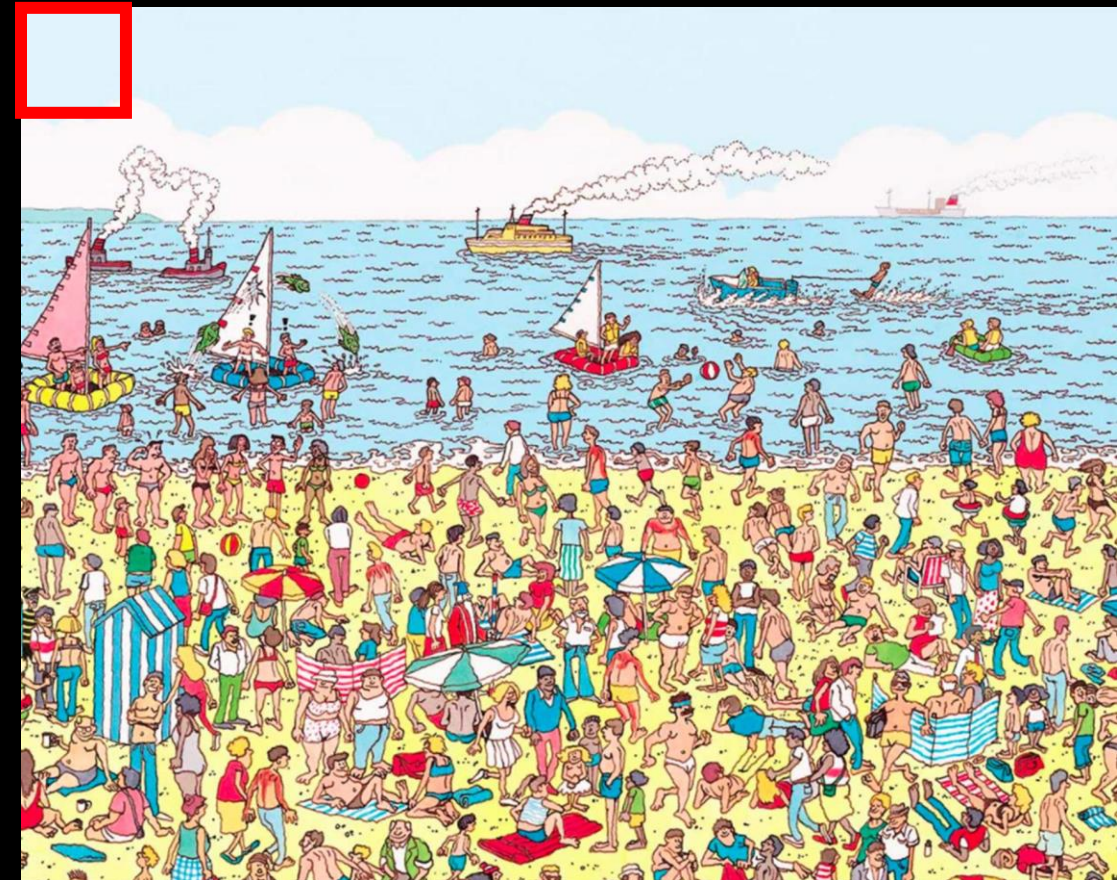
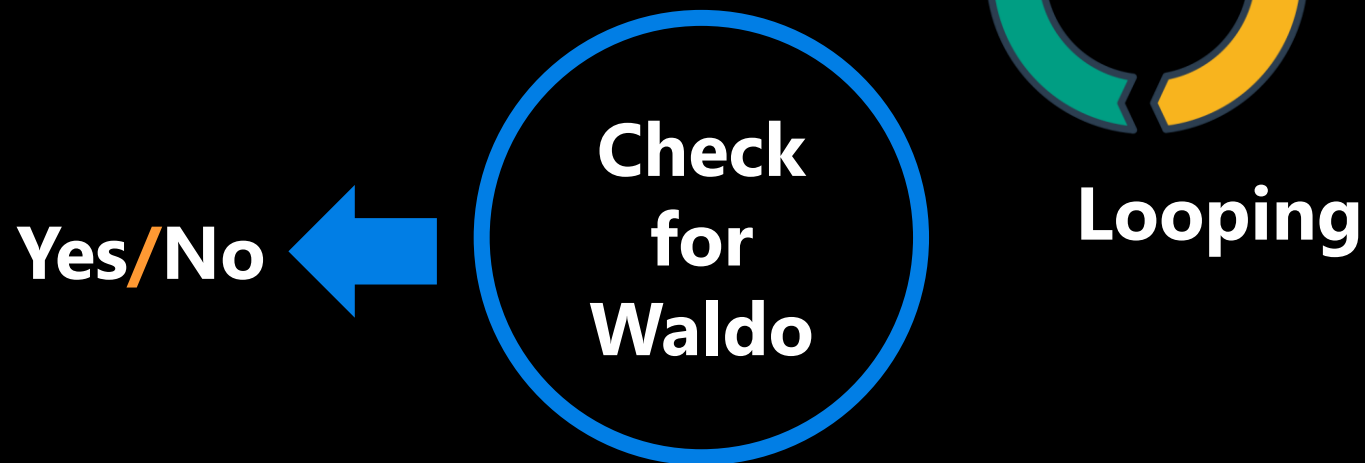
Looping



List of
Tweets

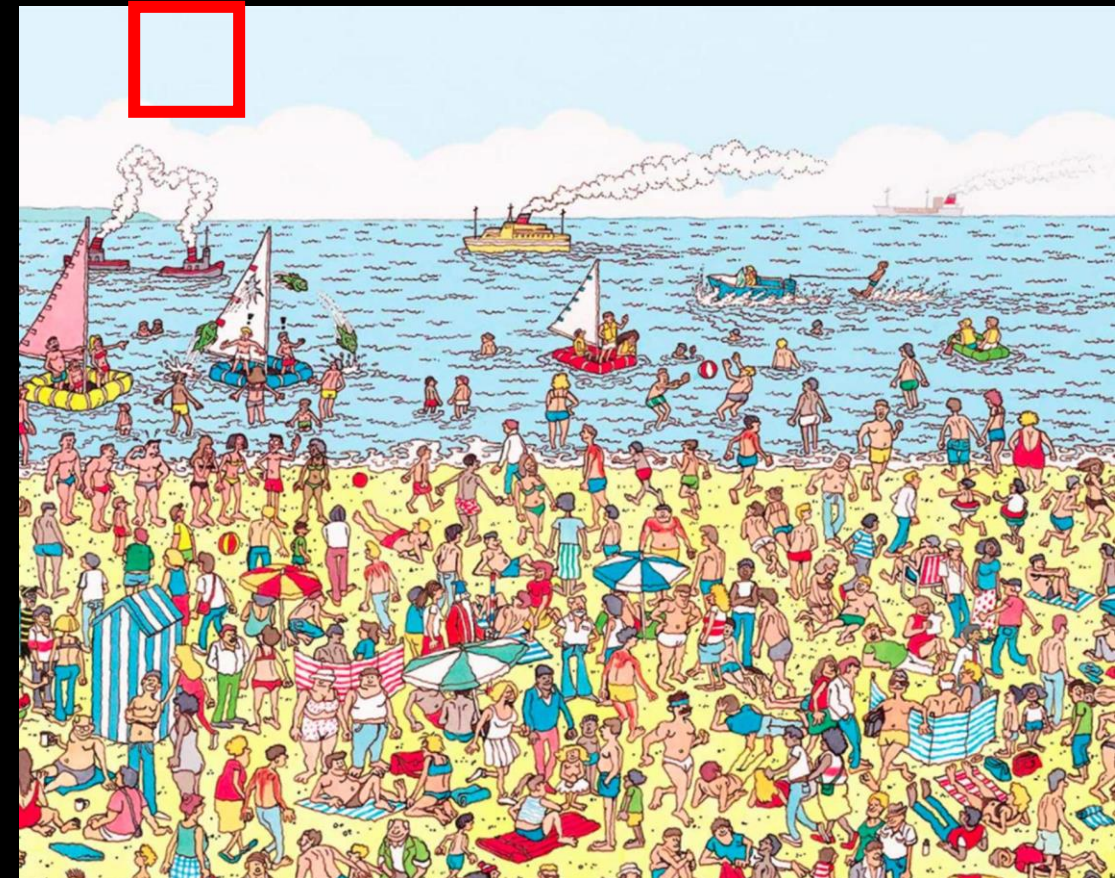
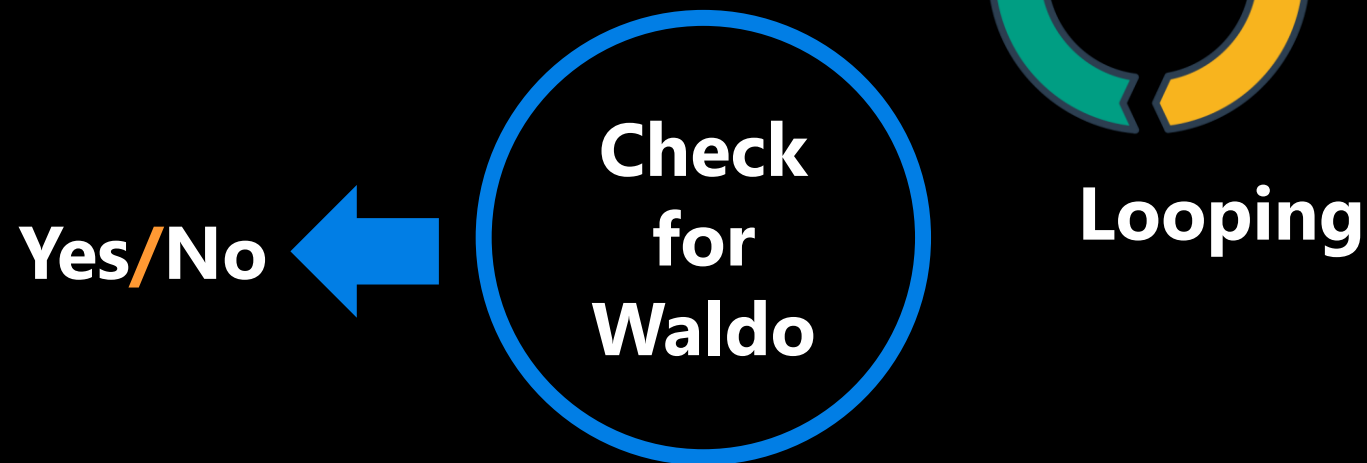
Looping (Iterating)

- Looping means repeating something over and over until a particular condition is satisfied.



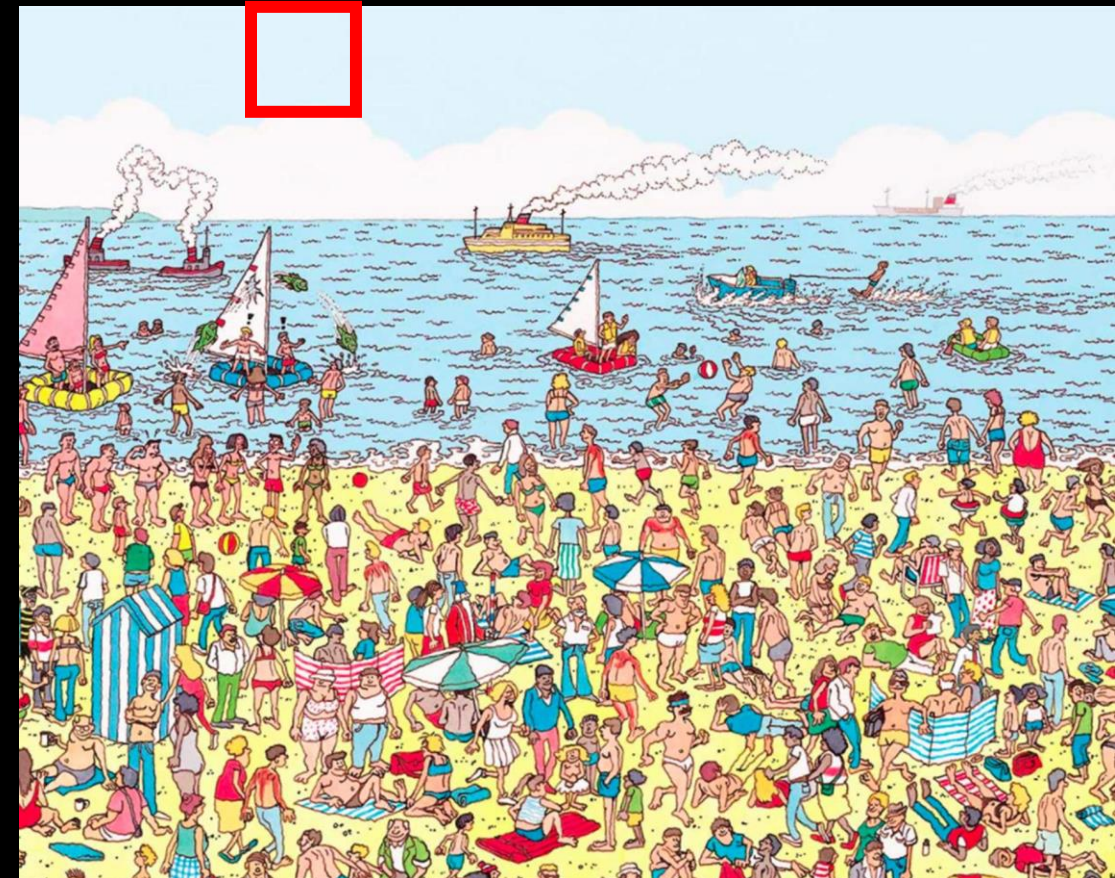
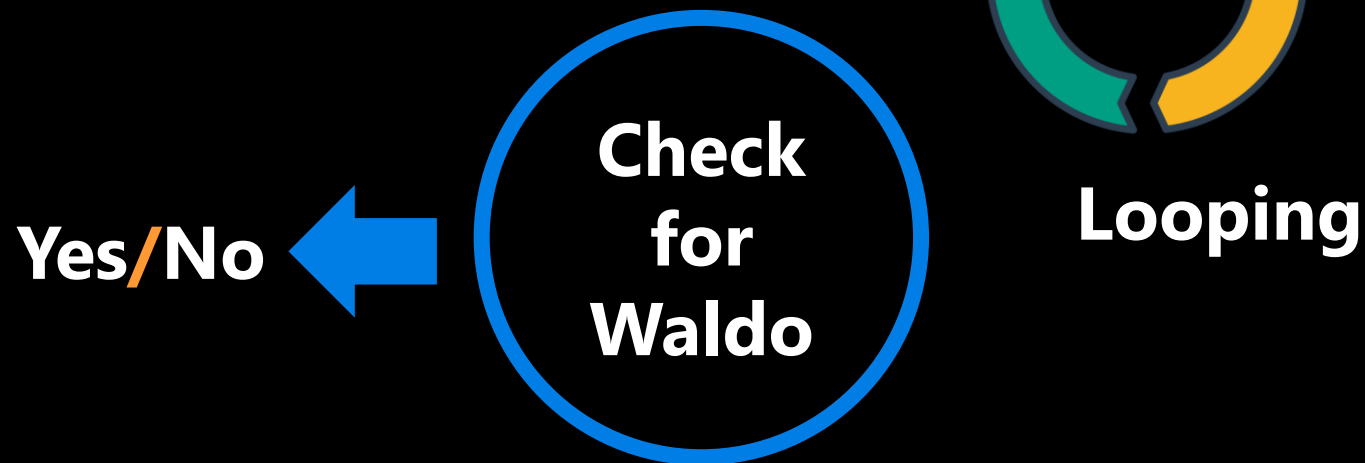
Looping (Iterating)

- Looping means repeating something over and over until a particular condition is satisfied.



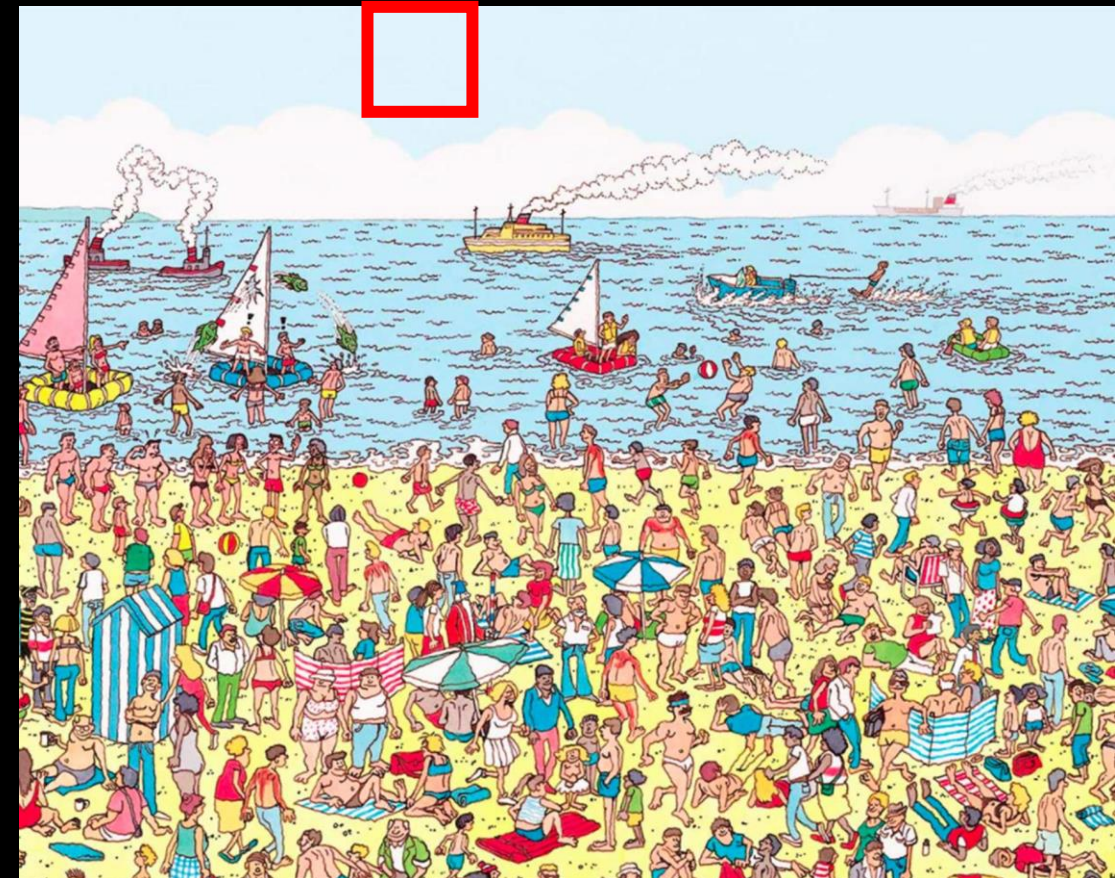
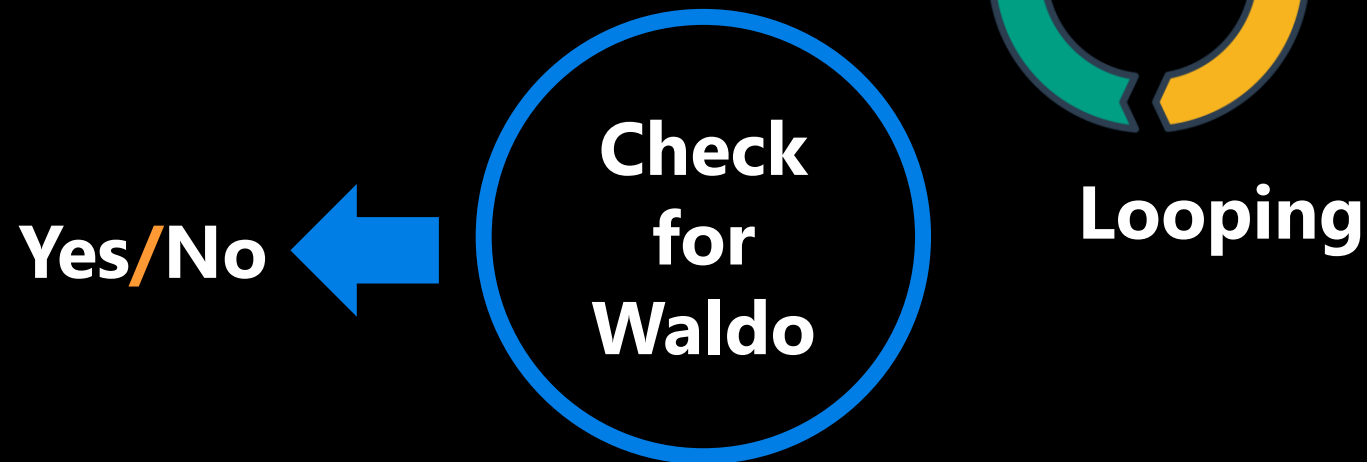
Looping (Iterating)

- Looping means repeating something over and over until a particular condition is satisfied.



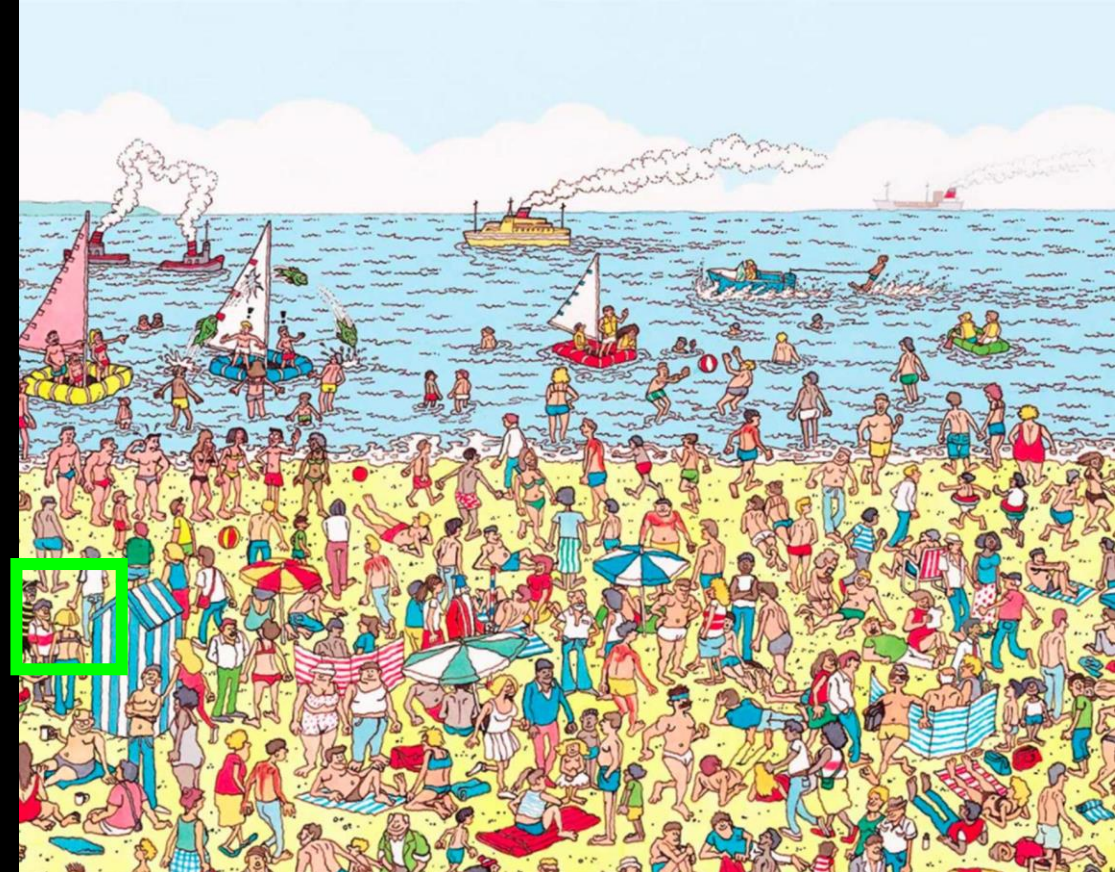
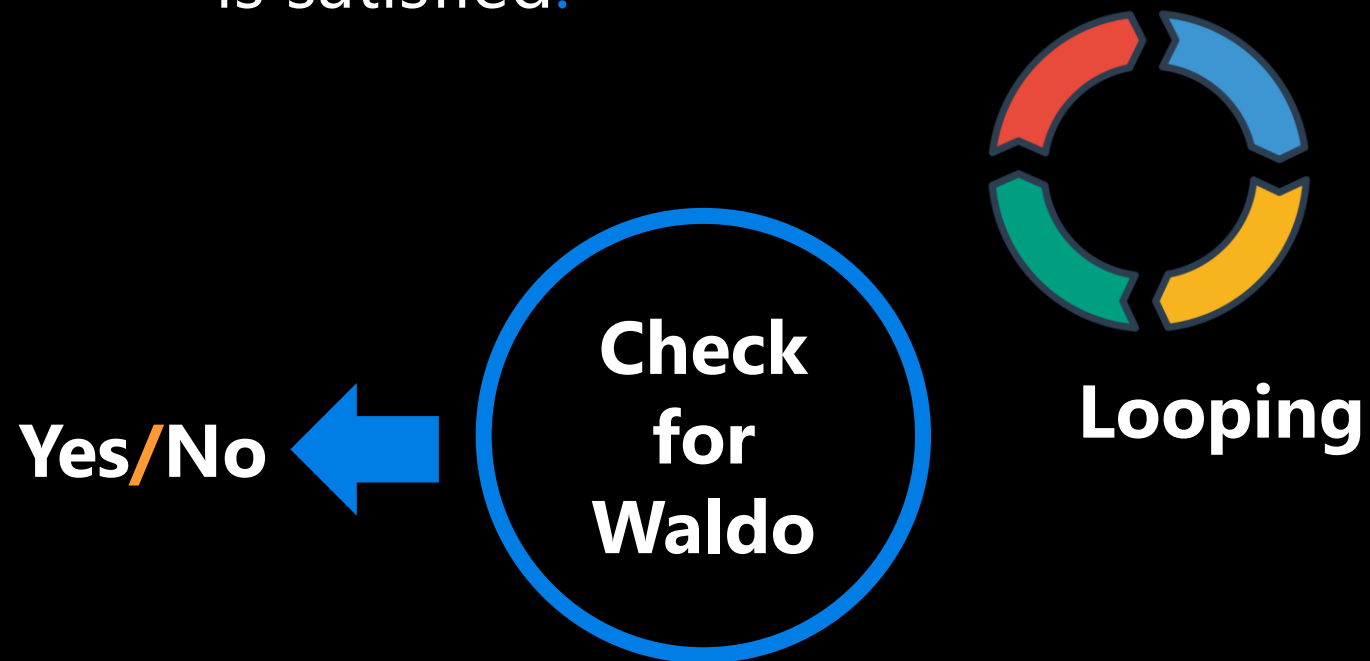
Looping (Iterating)

- Looping means repeating something over and over until a particular condition is satisfied.



Looping (Iterating)

- Looping means repeating something over and over until a particular condition is satisfied.



While Loops

- Sometimes we need to keep looping as long as some condition is **True**, and stop when it becomes **False**.
- Let's say you want to ask the user a question.
 - "Do you think the Toronto Maple Leafs will win the Stanley Cup in your lifetime?"
- If the user answers 'y', print out "You are going to live for a very long time." If the user answers 'n', print out "Well, sometimes miracles happen."

Open your notebook

Click Link:

1. Asking the User a Question

While Loops

- Oh code kinda worked but if the user makes a typo, they can't participate in the questionnaire.
- The general solution is to loop: to execute the same lines of code more than once. This is also called iteration.
- We're going to talk about one loop construct today: the while-loop where you loop while some boolean expression is True.

While Loops

- The **while loop** keeps executing a piece of code as long as a particular condition is **True**.
- There must be a colon (:) at the end of the while statement.
- The action to be performed must be indented.

Must evaluate to
True or False

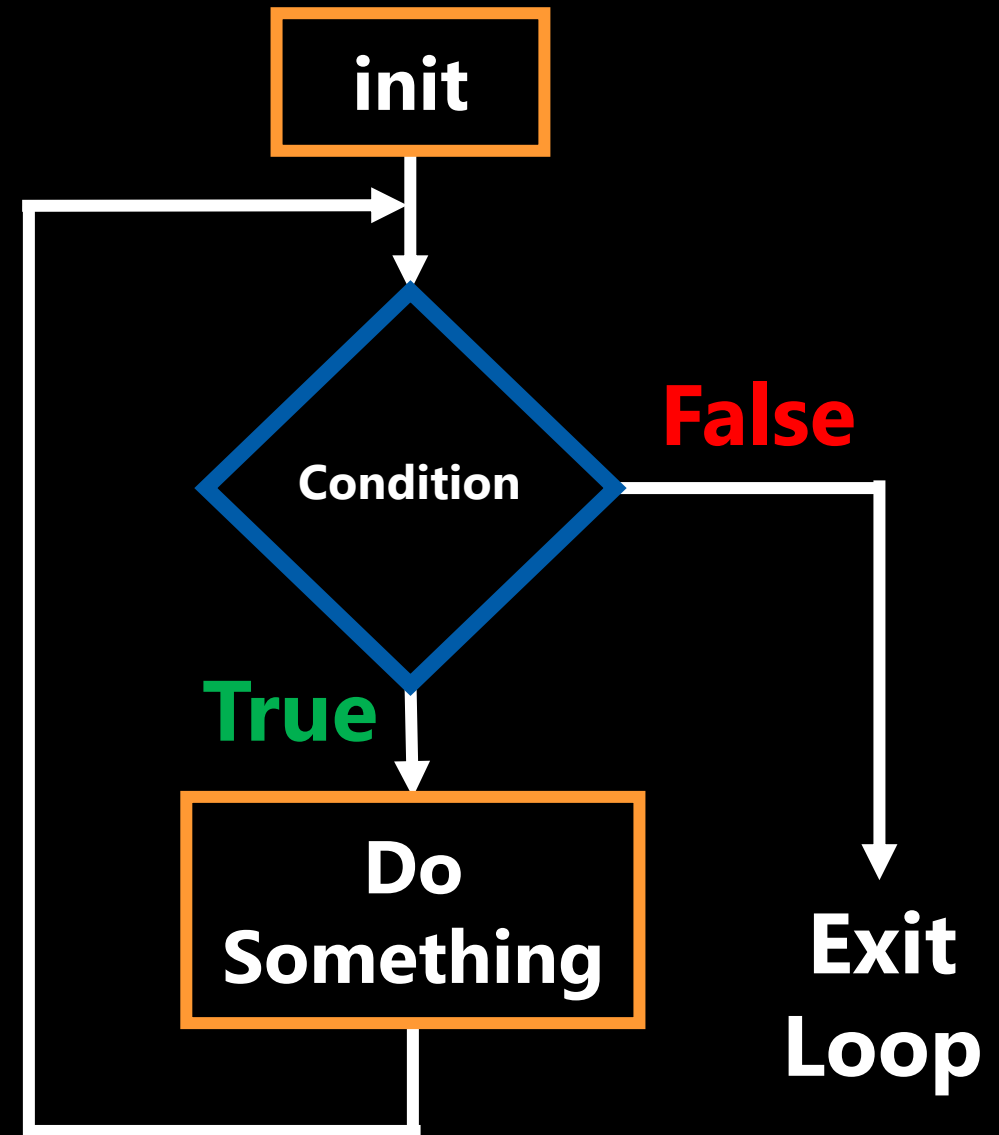
Colon

while expression:
do something.

Indent

While Loops

- The condition that gets evaluated is just a boolean expression.
- In particular it can include:
 - Something that evaluates to **True** or **False**.
 - logical operators (**and**, **or**, **not**)
 - comparison operators
 - function calls
- ... really anything that evaluates to **True** or **False**.



While Loops

Open your
notebook

Click Link:
2. While Loops

Must evaluate to
True or False

Colon

`while expression:`
`do something.`

Indent

while & for Loops

- In Python there are two types of loops **for** and **while**.
- For loops will be introduced in Week 6.
- What is the difference between **for** loops and **while** loops and when would we use one over the other?

```
for item in iterable:  
    do something.
```

```
while expression:  
    do something.
```

while & for Loops

- **for loop**
- The number of iterations to be done is already known.

```
for item in iterable:  
    do something.
```

```
cats = ['Persian', 'Siamese', 'Ragdoll']
```

```
for cat in cats:  
    print(cat)
```

```
>>> Persian  
>>> Siamese  
>>> Ragdoll
```

```
while expression:  
    do something.
```

while & for Loops

- **while** loop

- The number of iterations to be done is NOT known and iteration continues until a condition is met.

```
x = 0
while x*x < 200:
    print(x)
    x += 1
```

```
>>> 0
>>> ...
>>> 14
```

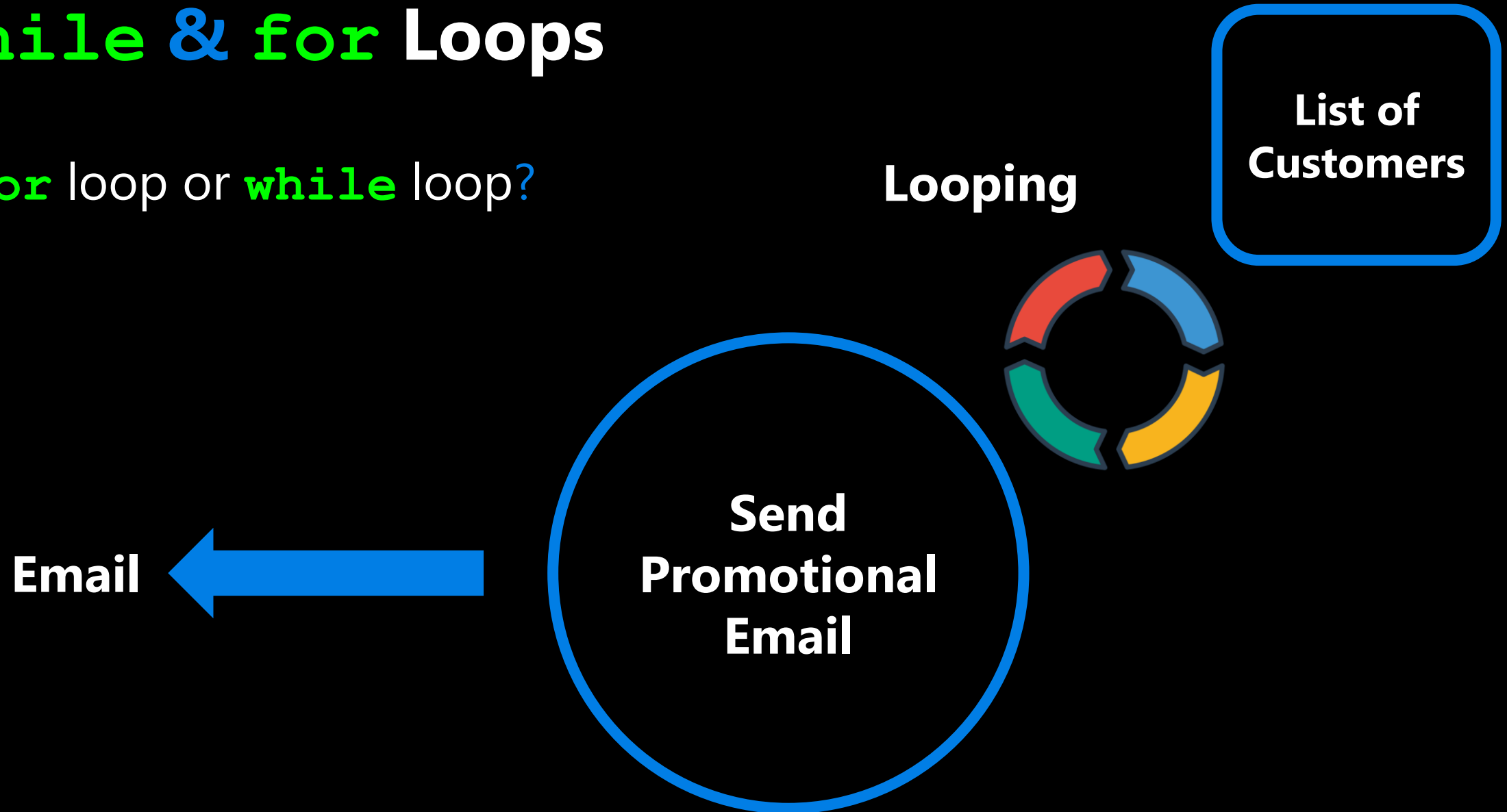
I don't know how many times I should iterate but I know when I should stop.

```
for item in iterable:
    do something.
```

```
while expression:
    do something.
```


while & for Loops

- **for** loop or **while** loop?



while & for Loops

- **for** loop or **while** loop?

Looping

List of
Tweets



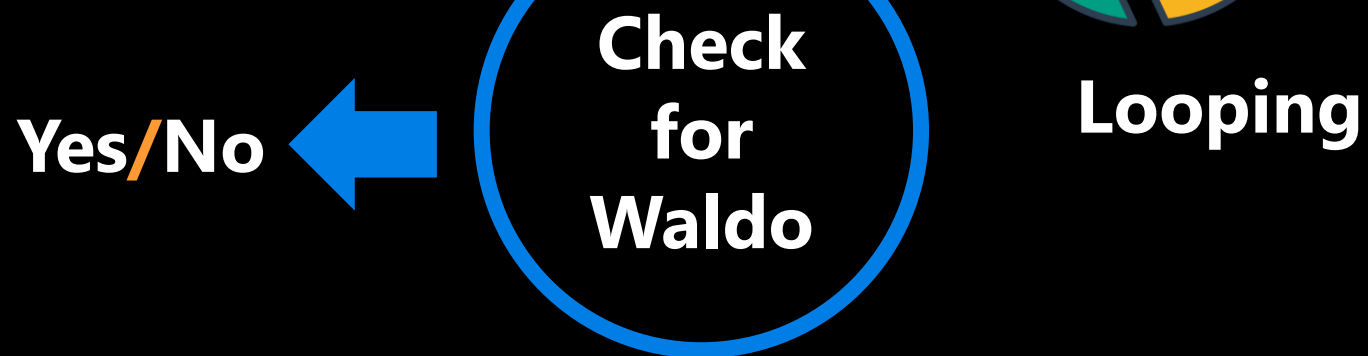
Yes/No



Does the
Tweet
contain
#cleancode

while & for Loops

- for loop or while loop?



Infinite Loops

- Remember that a **while** loop ends when the condition is satisfied (**True**).
- A common error when working with while loops is for the condition to never be satisfied and therefore, the loop to continue forever (till infinity).
- **We need some way inside the loop for the condition to become false.**

```
x = 0
while x < 10:
    print(x)
    x += 1
```

True

```
x = 0, 1, 2,
3, 4, 5, 6,
7, 8, 9
```

False

```
x = 10
```

Infinite Loops

- Remember that a **while** loop ends when the condition is satisfied (**True**).
- A common error when working with while loops is for the condition to never be satisfied and therefore, the loop to continue forever (till infinity).
- **We need some way inside the loop for the condition to become false.**

**Open your
notebook**

Click Link:

3. Infinite Loops

Variable Scope **and** Loops

```
def func(x):  
    x += 1
```

x (Local)



```
x = 0  
func(x)
```

x (Global)



x (Global)



```
x = 0
```

```
while x < 10:
```

```
    x += 1
```

x (Global)



While Loops

- Let's revisit our User Input code and see if the While Loop will solve out problem.

**Open your
notebook**

Click Link:

4. Back to User Input

Breakout Session 1

- Write code to print all the numbers from 0 to 20 that aren't evenly divisible by either 3 or 5.
- Zero is divisible by everything and should not appear in the output.

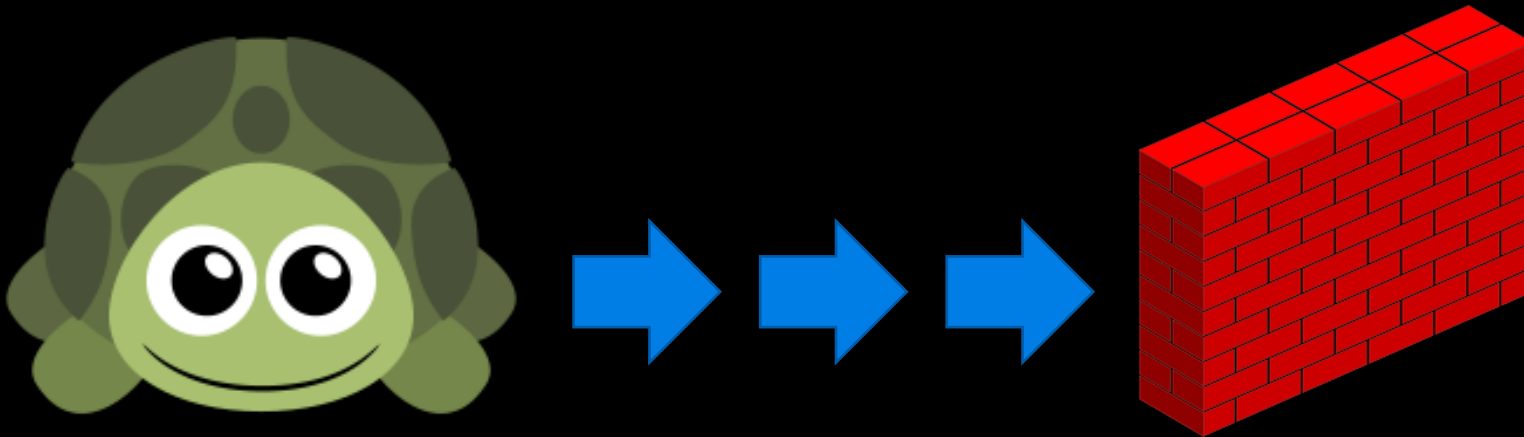
**Open your
notebook**

Click Link:

5. Breakout Session 1

Turtles and **while** loops

- I'm a little turtle and I want to take steps to the right until I get to the brick wall.
- However, I don't know how far away the brick wall I.



**Open your
notebook**

Click Link:
**6. Turtles and while
loops**

Random Module

- This module implements pseudo-random number generators for various distributions.

```
import random
```

```
random.uniform()
```

```
random.random()
```

```
random.randint()
```

```
...
```

**Open your
notebook**

Click Link:

7. Random Module

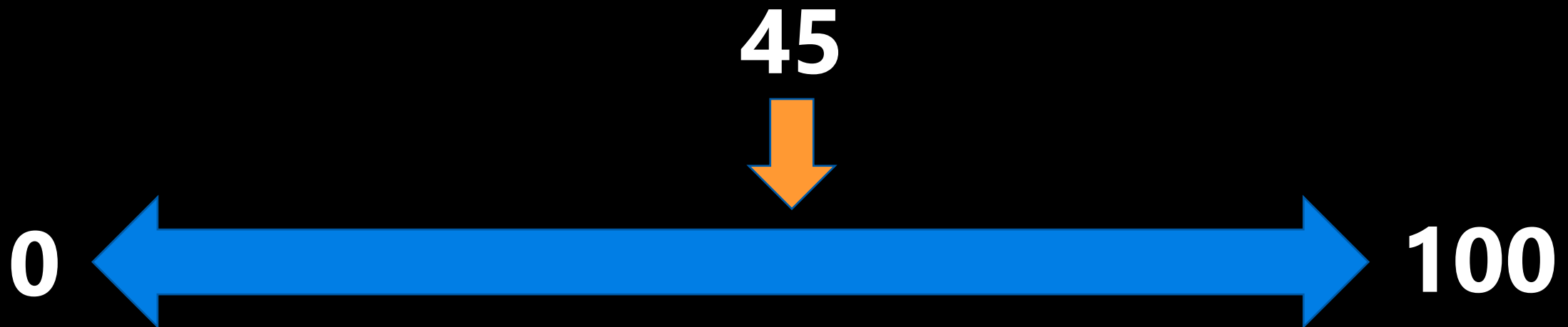
Guessing Game

- Let's build a simple guessing game.
 - Get the computer to choose a random integer from 0 to 100.
 - Ask the user for a guess and allow the user to input a guess or "q".
 - If the user inputs "q" print a nice message and end the program.
 - If the user enters a guess, tell them if they should guess higher, lower, or if they got it right.
 - If they got it right, print a nice message and quit.



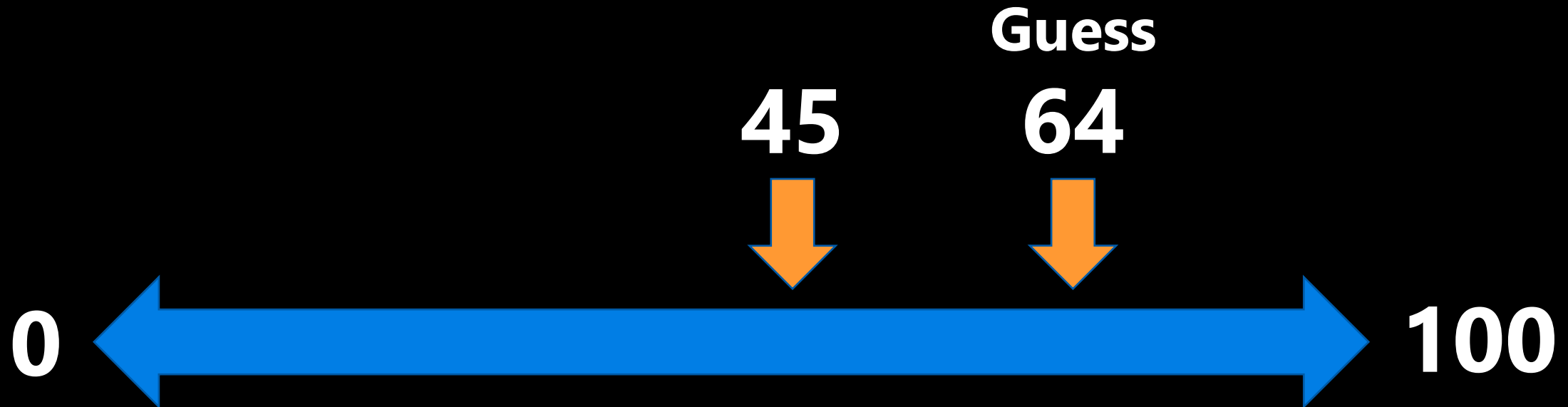
Guessing Game

- Get the computer to choose a random integer from 0 to 100.
 - The computer selects 45.

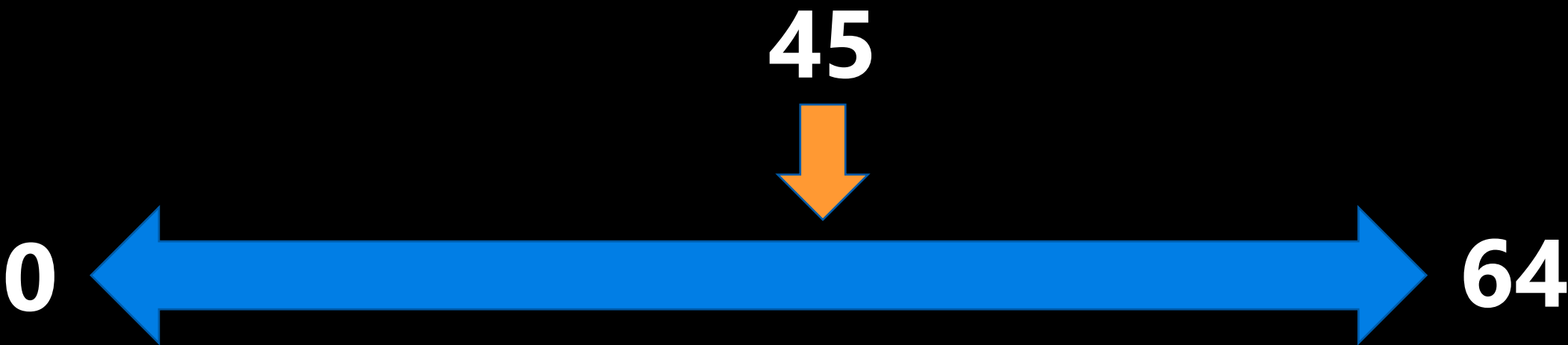


Guessing Game

- The user guesses 64.
 - The computer says **LOWER**.

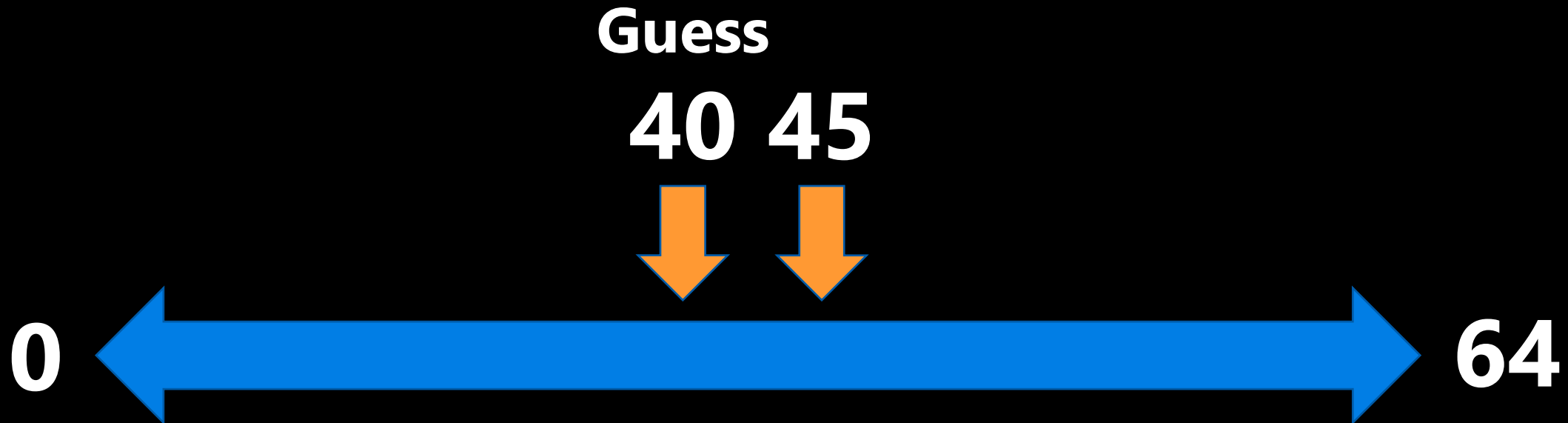


Guessing Game

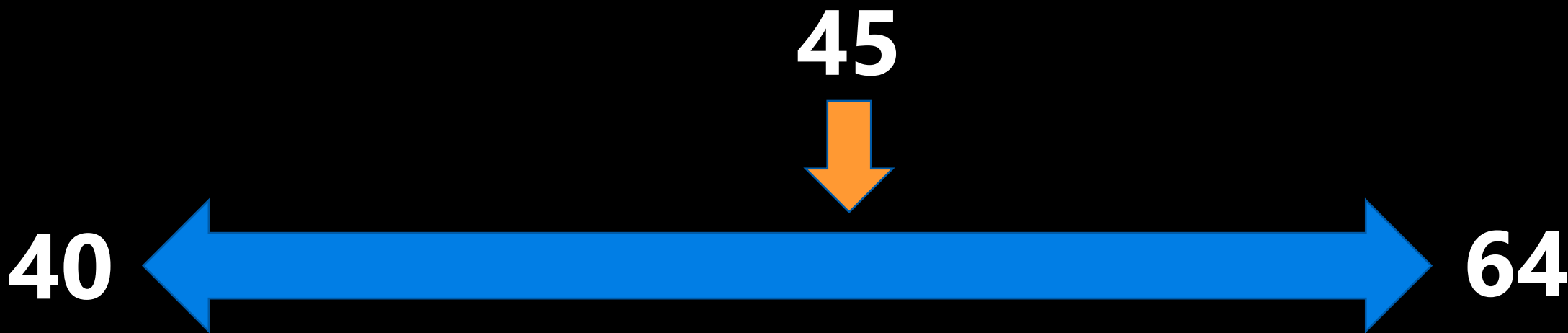


Guessing Game

- The user guesses 40.
 - The computer says **HIGHER**.

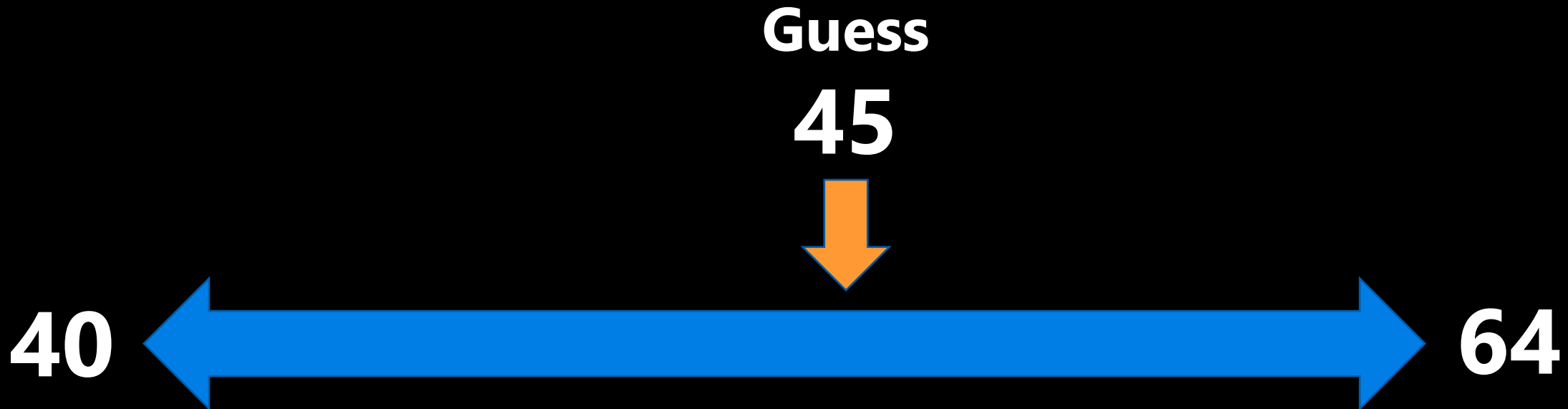


Guessing Game



Guessing Game

- The user guesses 45.
 - The computer says **YOU WIN.**



Guessing Game

- Let's build a simple guessing game.
 1. Get the computer to choose a random integer from 0 to 100.
 2. Ask the user for a guess and allow the user to input a guess or "q".
 3. If the user inputs "q" print a nice message and end the program.
 4. If the user enters a guess, tell them if they should guess higher, lower, or if they got it right.
 5. If they got it right, print a nice message and quit.

**Open your
notebook**

Click Link:

**8. A Simple Guessing
Game**

Lecture Recap

Practice!

- Looping (aka iteration) is the second key control structure in programming (if-statements/branching was the first).
- The basic idea of loops is to repeatedly execute the same block code.
- Looping is a very powerful idea.
- While loops is one of two loop types in Python.

functions, input & output, importing modules.

Week 4 | Lecture 1 (4.1)

if nothing else, write `#cleancode`