# Code Appetizer

## What does this code do?

```
def odds_and_evens(num):
   for i in range(num):
     if(i%2==0):
       print i, "is even"
     else:
       print i, "is odd"
odds_and_evens(5)
```

# Code Appetizer

What does this code do?

```
def odds_and_evens(num):
    for i in range(num):
        if(i%2==0):
            print i, "is even"
        else:
            print i, "is odd"
odds_and_evens(5)
```

```
0 is even
1 is odd
2 is even
3 is odd
4 is even
```

Hackbright
Academy

# Loops Lecture

## While Loops

Hackbright
Academy

# While Loops

While loops are used when you want to complete a task until a certain *condition* is met.

Useful when we *don’t know* how many times we need to repeat something.

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
num_list = [2, 4, 2, 6, 3]
index = 0
while(num_list[index] <= 5):
   index = index + 1
print “Found a number greater than 5!”
```

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
⇒ num_list = [2, 4, 2, 6, 3]
  index = 0
  while(num_list[index] <= 5):
      index = index + 1
  print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]

Hackbright Academy

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
   num_list = [2, 4, 2, 6, 3]
⇒  index = 0
   while(num_list[index] <= 5):
       index = index + 1
   print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]
index ⇒ 0

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
   num_list = [2, 4, 2, 6, 3]
   index = 0
⇒  while(num_list[index] <= 5):
       index = index + 1
   print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]
index ⇒ 0
num_list[index] ⇒ 2

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
   num_list = [2, 4, 2, 6, 3]
   index = 0
   while(num_list[index] <= 5):
⇒      index = index + 1
   print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]
index ⇒ 1

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
  num_list = [2, 4, 2, 6, 3]
  index = 0
⇒ while(num_list[index] <= 5):
      index = index + 1
  print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]  
index ⇒ 1  
num_list[index] ⇒ 4

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
  num_list = [2, 4, 2, 6, 3]
  index = 0
  while(num_list[index] <= 5):
⇒     index = index + 1
  print “Found a number greater than 5!”
```

num_list ⇒  [2, 4, 2, 6, 3]
index ⇒ 2

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
   num_list = [2, 4, 2, 6, 3]
   index = 0
⇒  while(num_list[index] <= 5):
       index = index + 1
   print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]
index ⇒ 2
num_list[index] ⇒ 2

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
  num_list = [2, 4, 2, 6, 3]
  index = 0
  while(num_list[index] <= 5):
⇒     index = index + 1
  print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]
index ⇒ 3

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
   num_list = [2, 4, 2, 6, 3]
   index = 0
⇒  while(num_list[index] <= 5):
       index = index + 1
   print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]
index ⇒ 3
num_list[index] ⇒ 6

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
    num_list = [2, 4, 2, 6, 3]
    index = 0
    while(num_list[index] <= 5):
        index = index + 1
⇒   print “Found a number greater than 5!”
```

num_list ⇒ [2, 4, 2, 6, 3]
index ⇒ 3
num_list[index] ⇒ 6
⇒ “Found a number greater than 5!”

Hackbright Academy

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
num_list = [2, 4, 2, 6, 3]
index = 0
while(num_list[index] <= 5):
   index = index + 1
print “Found a number greater than 5!”
```

**What could go wrong?**

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
num_list = [2, 4, 2]
index = 0
while(num_list[index] <= 5):
   index = index + 1
print “Found a number greater than 5!”
```

**If there is no number greater than 5 in the list, then we would eventually get an `IndexError`.**

Hackbright Academy

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
num_list = [2, 4, 2]
index = 0
while(index<len(num_list) and num_list[index] <= 5):
    index = index + 1
print “Found a number greater than 5!”
```

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
num_list = [2, 4, 2]
index = 0
while(index<len(num_list) and num_list[index] <= 5):
  index = index + 1
print “Found a number greater than 5!”
```

**But now we don’t know if we found a number or not!**

# While Loops - Example

Finding a number that’s greater than 5 in a list.

```
num_list = [2, 4, 2]
found_num = False
index = 0
while(not found_num and index<len(num_list)):
    if(num_list[index] > 5):
        found_num = True
    else:
        index = index + 1
if(found_num):
    print “Found a number greater than 5!”
```

# While Loop Dangers - Infinite Loops

We need to be careful that our while loops eventually stop, or else we will get stuck in an *infinite loop*.

To exit a program with an infinite loop, you have to press `Control+C` on your keyboard.

# While Loop Dangers - Infinite Loops

This is an infinite loop:

```
i=0
while(i<100):
  print i
```

# While Loop Dangers - Infinite Loops

This is **not** an infinite loop because we are incrementing the *counter* variable `i`:

```
i=0
while(i<100):
   print i
   i = i+1
```

Hackbright
Academy

# While Loop Dangers - Infinite Loops

## Infinite Loop (bad!):

```
i=0
while(i<100):
   print i
```

## Finite Loop:

```
i=0
while(i<100):
   print i
   i = i+1
```

# Extremely Useful While Loop

While loops are very useful when asking for input from users.

```
items = []
while(True):
    print "Type exit to exit"
    item = raw_input("Add an item: ")
    if(item == "exit"):
        break
    else:
        items.append(item)
print “Your items are”, items
```

Hackbright Academy

# Extremely Useful While Loop

While loops are very useful when asking for input from users.

```
items = []
while(True):
    print "Type exit to exit"
    item = raw_input("Add an item: ")
    if(item == "exit"):
        break
    else:
        items.append(item)
print “Your items are”, items
```

Repeats forever

Immediately gets out of the loop

Hackbright Academy

# Converting between For and While Loops

## FOR:

```
for num in my_list:
    print num
```

## WHILE:

```
index = 0
while(index<len(my_list)):
    print num_list[index]
    index += 1
```

**Shorthand for index = index + 1**

# Exercise Time!

1. Do Loops Exercises - While Loops
2. Finish Loops Exercises - For Loops
3. Finish List Exercises 2
4. Finish List Exercises 1
    a. Flag me down when you are done so I can check it.
5. Answer the questions from Project Euler.

    https://projecteuler.net/archives