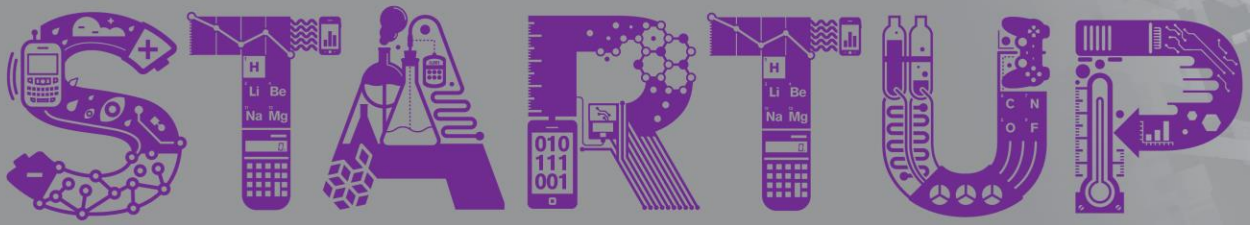
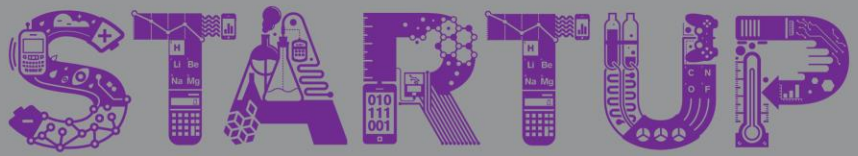
Bloomberg



Bloomberg



Python Lesson 4

Loops

Review Conditionals

What does this code do?

```
def am_or_pm(hour):
    if hour > 11:
        print("It is now PM")
    else:
        print("It is now AM")

am_or_pm(23)
```

```
It is now PM
```

Review Conditionals

What does this code do?

```
def am_or_pm(hour):
    if hour > 11:
        print("It is now PM")
    else:
        print("It is now AM")

am_or_pm(6)
```

```
It is now AM
```

Loops: Introduction

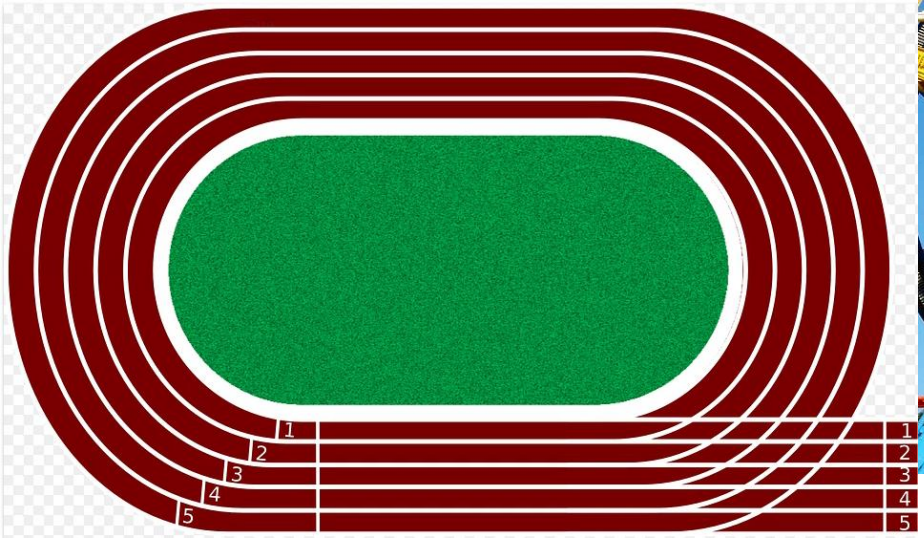
What is a loop?

Loops: Introduction

What is a loop?

Loops: Introduction

What is a loop?



Loops: Introduction

Loops are a way of repeating code over and over again until a specific condition has been met (or forever!)

Why loops?

If we want to write a program that prints out every number between 1 and 10?

Why loops?

If we want to write a program that prints out every number between 1 and 10?

```
print("1")
print("2")
print("3")
...
print("10")
```

Why loops?

- What do you want to do over and over?
  - Print a number:

```
print(number)
```

- Where do you want to start?
  - Number starts at 1

```
number = 1
```

- What do you want to change?
  - Increase number by 1:

```
number = number + 1
```

- How long do you want to do it for?
  - Until the number goes beyond 10:

```
while number <= 10:
```

Why loops?

If we want to write a program that prints out every number between 1 and 10:

```
number = 1
while number <= 10:
    print(number)
    number = number + 1
```

Anatomy of a While loop

while indicates that the block will repeat while the condition is true

The **condition** works just like in an 'if' statement

```
number = 1
while number <= 10:
    print(number)
    number = number + 1
```

The : tells the program you are starting the code for the while loop.

The loop code goes here. It must all be indented the same amount.

Incrementing the number is very important!

While loops

Not very different from an if statement:

```
num_bananas = 10
if num_bananas > 5:
    print("You have a bunch of bananas!")
```

```
num_bananas = 10
while num_bananas > 5:
    print("You have a bunch of bananas!")
```

Infinite loops

Beware of infinite loops!

```
counter = 1
while counter > 0:
    print(counter)
    counter = counter + 1
```

Remember: How long you want to run it for is a good question!

While loop

Loops are not limited to counting things!

```
user_reply = ""
while user_reply != "no":
    user_reply = input("Would you like to continue?")
```

Group Practice

Write 3 short programs using while loops to achieve the following results:

1.) Print all numbers between 1 and 100 in descending order
2.) Print all even numbers between 1 and 100 in ascending order
3.) Print all odd numbers between 1 and 100 in ascending order

Python Explorer

```
Please enter a direction: N
Walking towards north
Please enter a direction: S
Walking towards south
Please enter a direction: North
Looking around
Please enter a direction: look
Looking around
Please enter a direction: end
The game has ended!
```

```
user_input = input("Please enter a direction: ")
user_input_upper = user_input.upper()

while user_input != "END":

    if user_input_upper == 'N':
        print("Walking towards north")
    else:
        if user_input_upper == 'S':
            print("Walking towards south")
        else:
            print("Looking around")

    user_input = input("Please enter a direction: ")
    user_input_upper = user_input.upper()

print("The game has ended!")
```

Recap

- A while loop helps you avoid repeating code
- It works just like if-statements, but repeats the code block until the condition is False.
- Make sure that any loop you write will end!