Coláiste na Tríonóide, Baile Átha Cliath
Trinity College Dublin
Ollscoil Átha Cliath | The University of Dublin

Fondúireacht Eolaíochta Éireann
Dá bhfuil romhainn
Science Foundation Ireland
For what's next

Ollscoil Teicneolaíochta Bhaile Átha Cliath
Technological University Dublin

technoteach
technocamps

Llywodraeth Cymru
Welsh Government

Prifysgol Abertawe
Swansea University

Cardiff University
Prifysgol Caerdydd

Prifysgol Bangor University

Cardiff Metropolitan University
Prifysgol Metropolitan Caerdydd

University of South Wales
Prifysgol De Cymru

Cyngor Cyllido Addysg Uwch Cymru
Higher Education Funding Council for Wales
hefcw

Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Prifysgol Aberystwyth University

Prifysgol Glyndŵr Wrecsam
Wrexham Glyndŵr University

institute of CODING in wales
technocamps

technoteach
technocamps

Branwen's Starling and Python

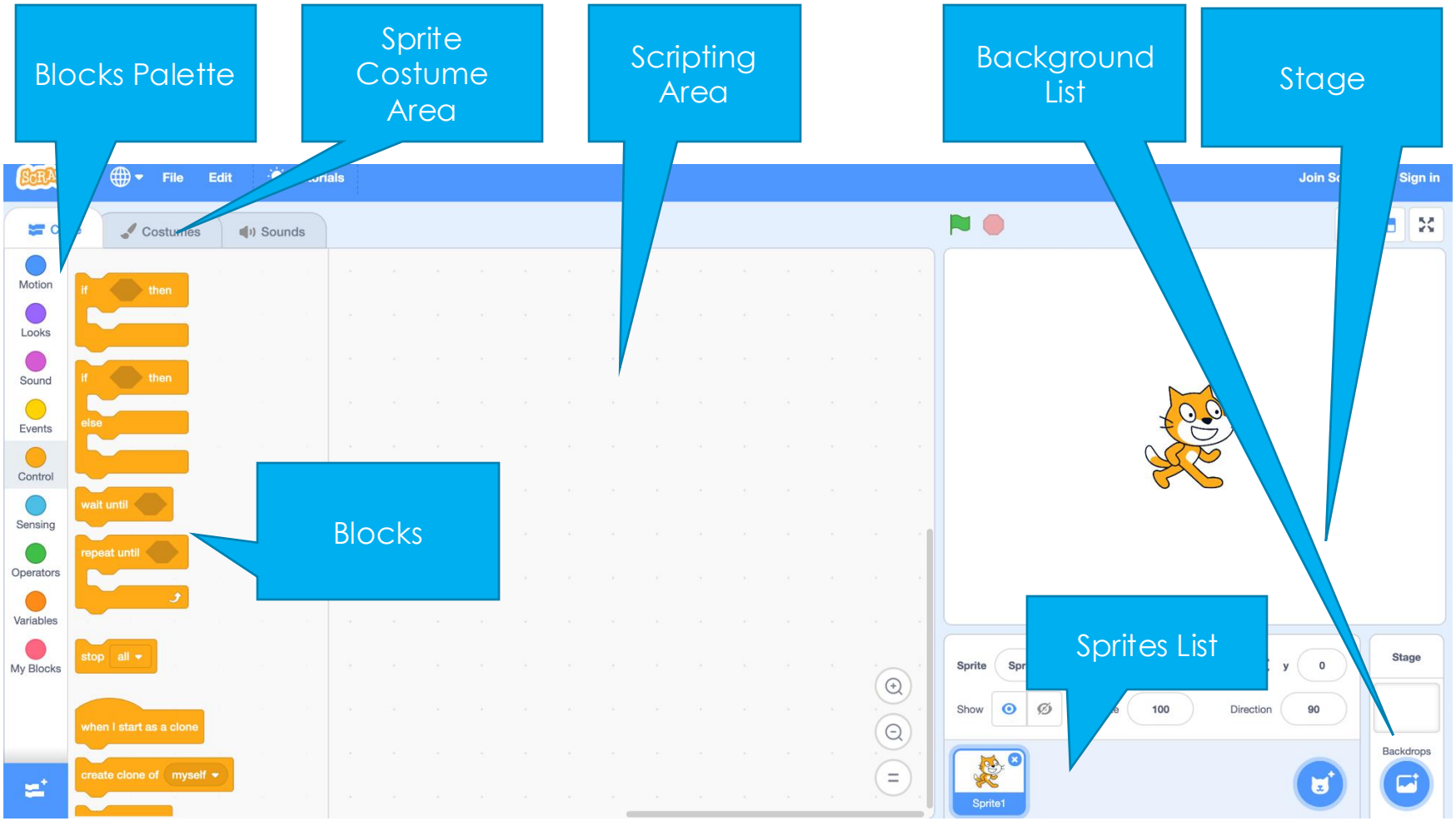
Branwen and the Starling - A Welsh Tale

- Branwen, Bendigeidfran's sister, had married Matholwch, the King of Ireland.
- This angered Efnysien, Branwen's other brother, as his permission was not asked, so he killed all of Matholwch's horses.
- Branwen and Matholwch managed to escape to Ireland, but because of the actions of Efnysien, Matholwch decided to imprison Branwen.
- In jail, Branwen reared a Starling to fly home to Wales carrying a letter for Bendigeidfran.

technoteach
technocamps

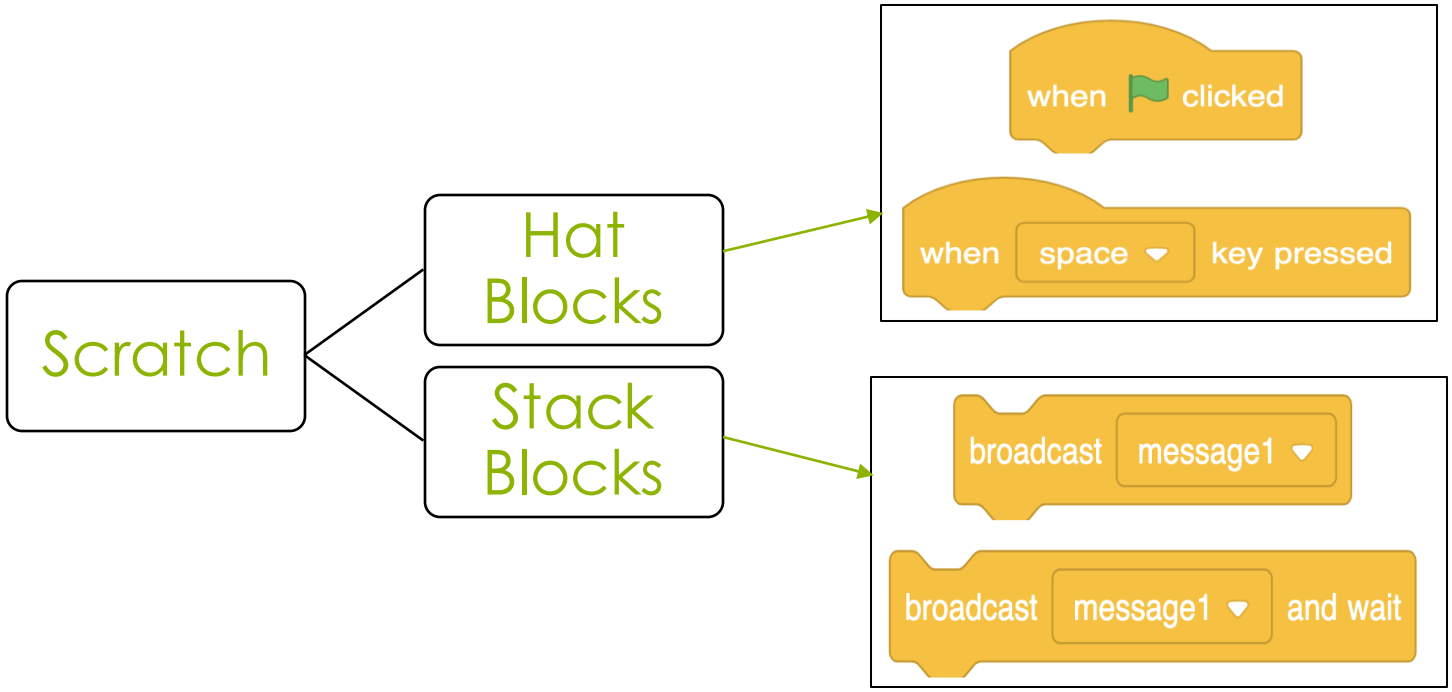
Levelling up Scratch to Python

Scratch: A Brief Reminder



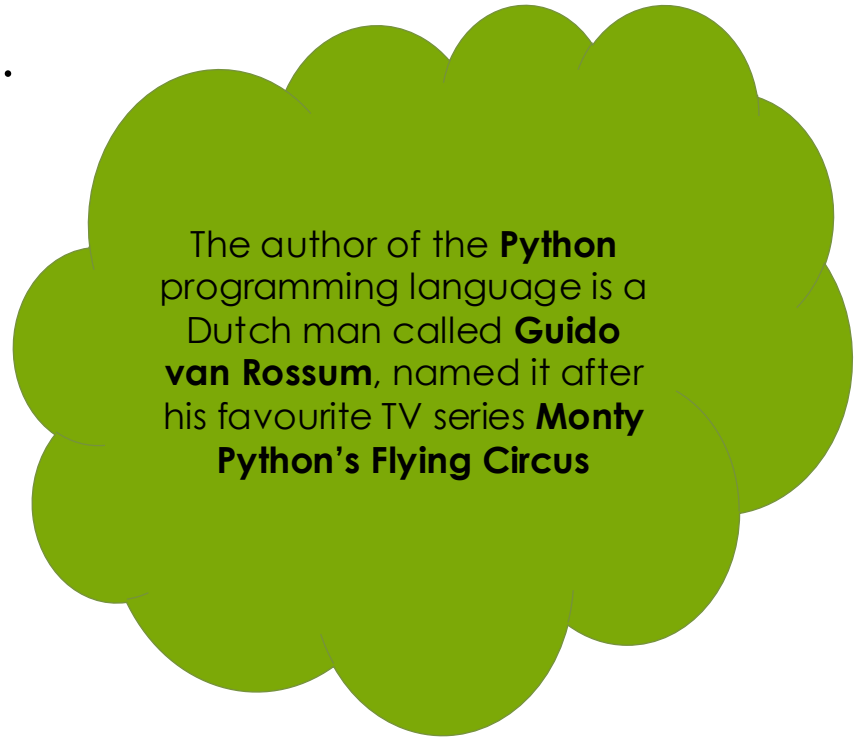
Scratch: A Brief Reminder

- Drag and drop blocks to give instructions
- Easy to create games
- User friendly



Introducing Python

- It is a programming language which tells the computer what to do using algorithms.
- It is free.
- It is easy to learn, read and code.
- It is interactive and portable.
- It is high level and flexible.



Python Basics

1.Syntax Simplicity 

1. Python emphasises clean, readable code.
2. Indentation is used for code blocks (no braces).

```
if True:
    print("Hello, Python!")
```

2. Easy Data and Variable Management

1. Store and manipulate various data
2. No need to declare data types explicitly

```
age = 25
name = "Alice"
```

Branwen's Starling in Scratch

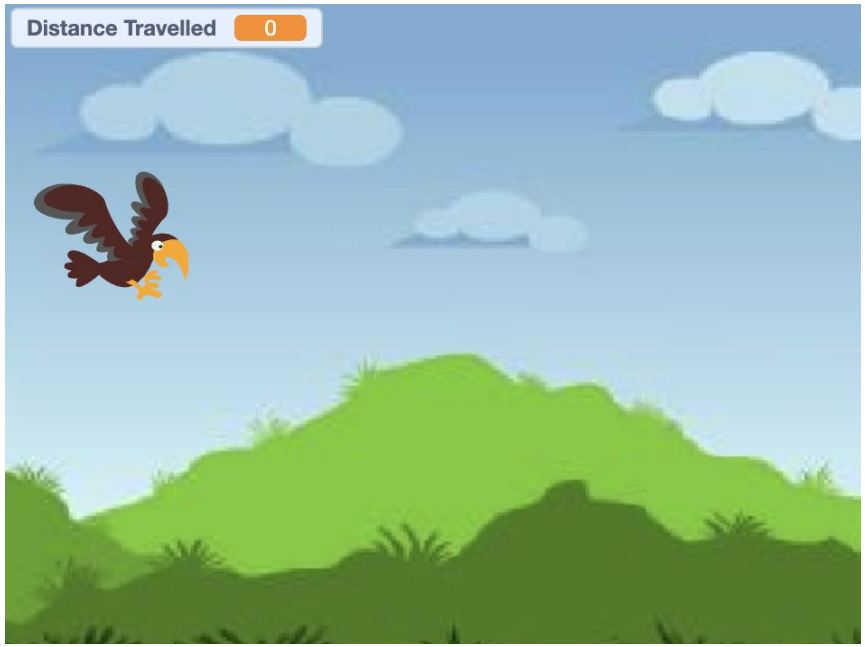
Opening Branwen's Starling in Scratch

Go to

TC1.me/BranwenScratch

And press "See Inside"

We should now see our game



Running and Playing the Game

Now we can explore our game, and see what different characters do

Try playing the game, and see what you need to do

By clicking on the Background, Starling and Osprey, we can look at the different bits of code, to work out how the game is played

Answer these **3** questions

1. How do you win and lose the game?
2. How and when do the enemies spawn?
3. How do we move our character?

Making Branwen's Starling in Python

Pytch: Converting Scratch to Python

Pytch is a Python library and IDE which allows us to create Scratch-Like applications in Python

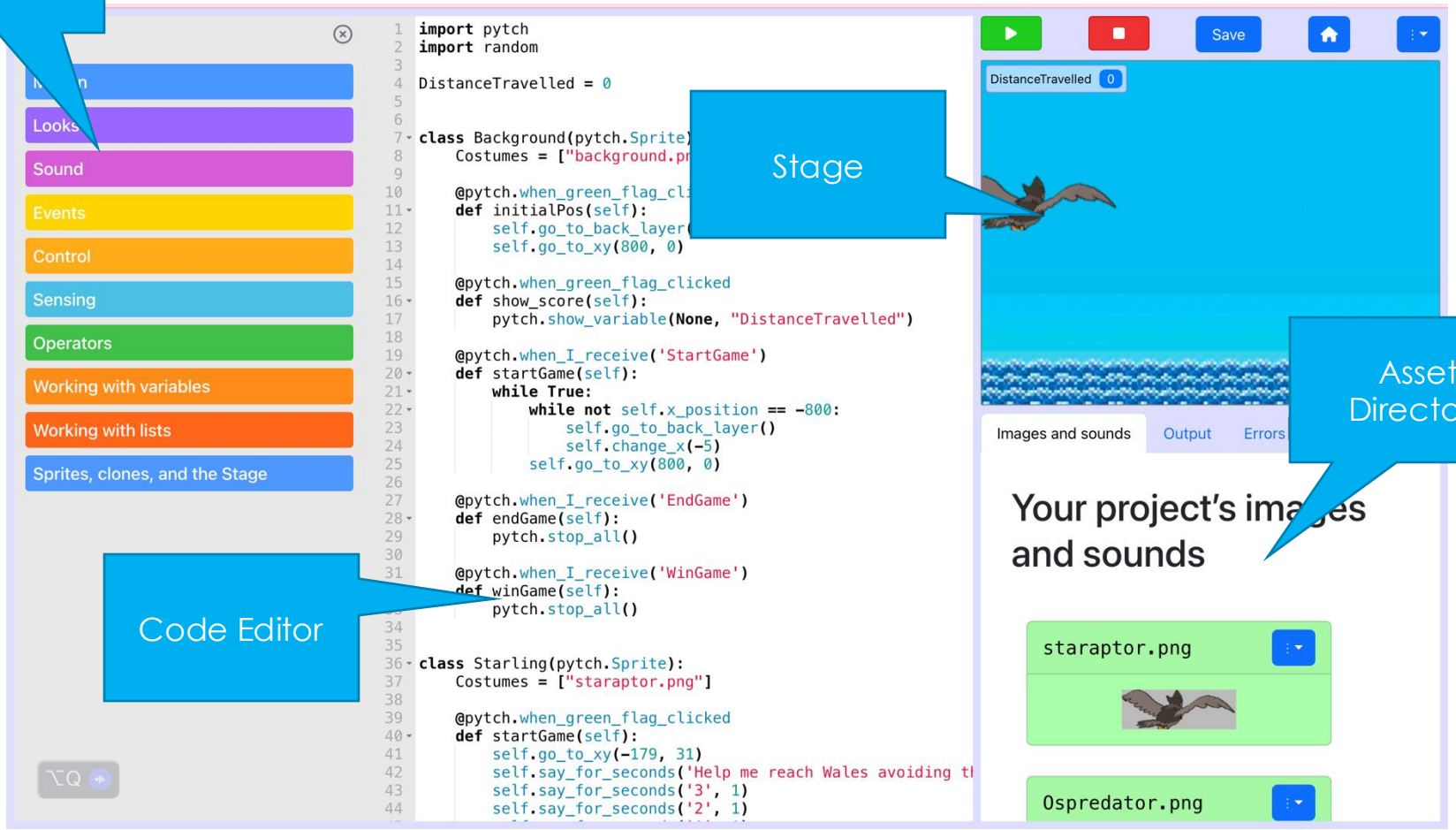
We are able to use many of the phrases used in Scratch for making this applications

We can also see blocks as a reference!

It simplifies much of the complexity from Python

It also means we can make games easily!

Pytch: Converting Scratch to Python

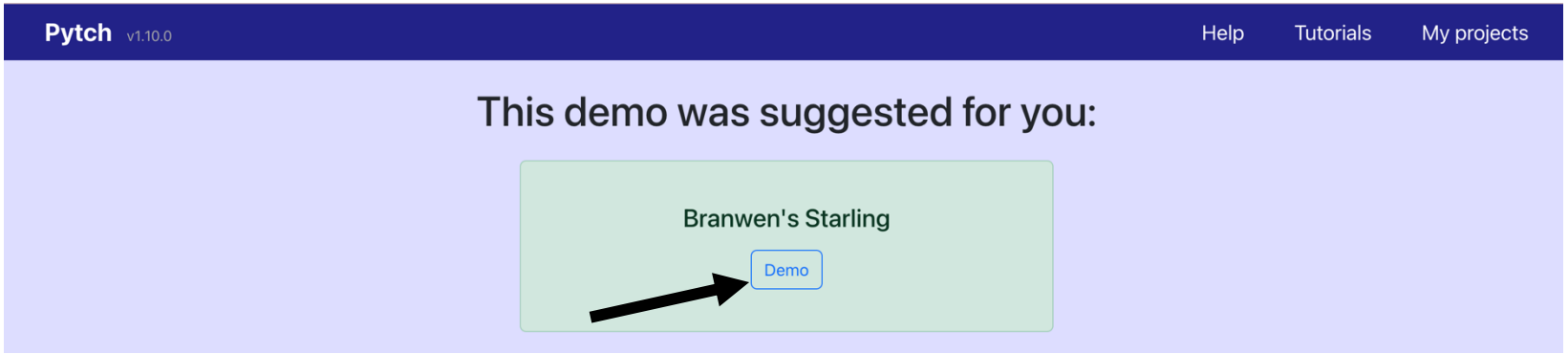


Opening Pytch

Go to

TC1.me/BranwenPytchSBS

And click **“Demo”** to see our starter game!

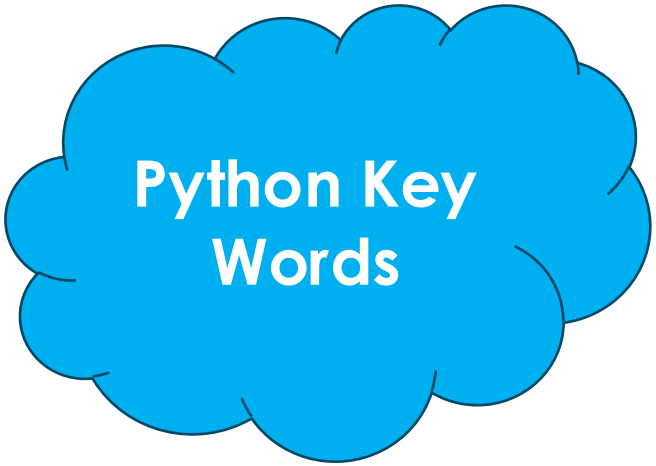


Code Structure

Have a **5 minute** explore of the code, but don't change anything…

Yet!

See if we can identify some key words in our code, and any words which are repeated a lot!


Python Key
Words

Coding the Main Character

Our character: The Starling

Starting from line 36, we can see our **Starling** Class

This **class** is where all our code for our Starling will go

Inside our Class, we can break our code up into **functions**

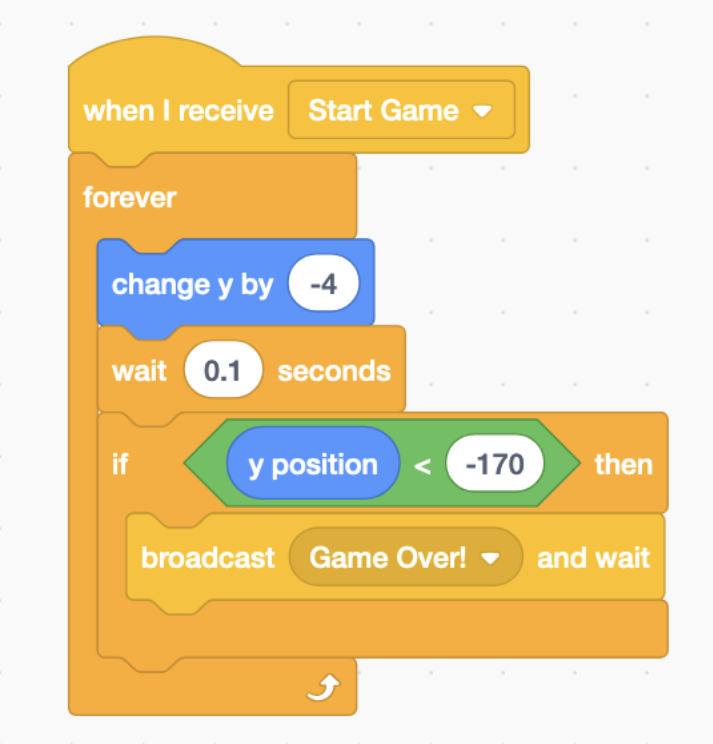
Each function uses the **Def** key word

A **function** in Pytch is like a script in Scratch

Main Character: Making Her Fall...

Our Starling currently stays still

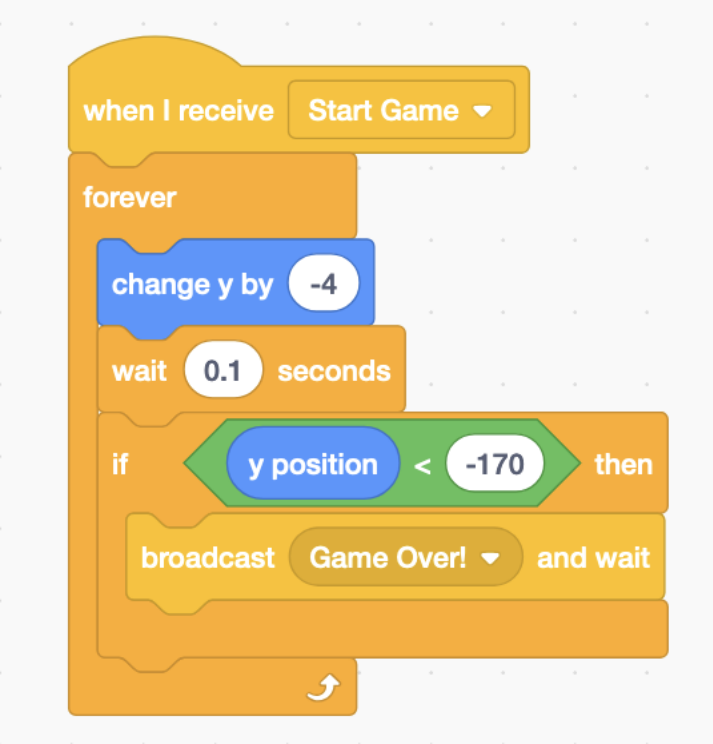
Lets convert this Scratch code to Python code



Main Character: Making Her Fall...

Our Starling currently stays still

Lets convert this Scratch code to Python code

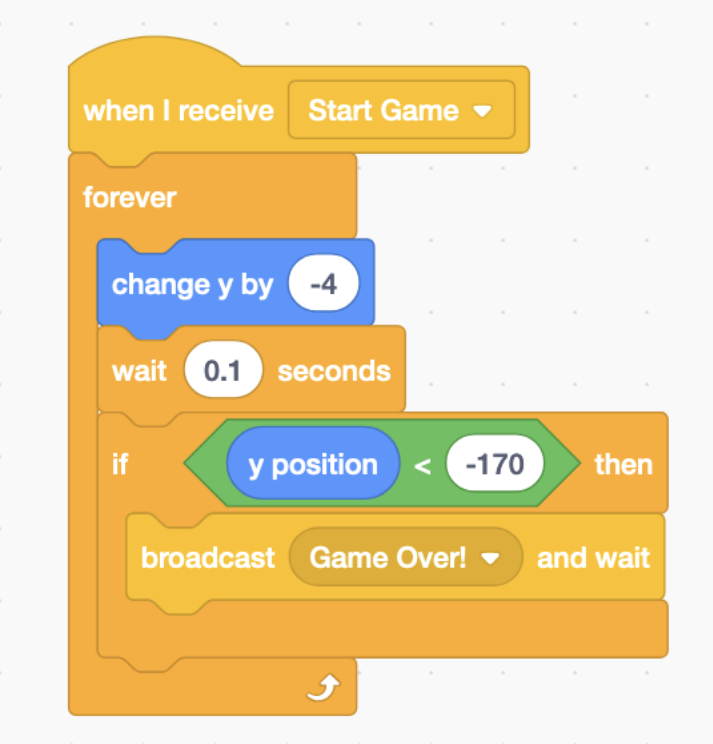


while True:

Main Character: Making Her Fall...

Our Starling currently stays still

Lets convert this Scratch code to Python code

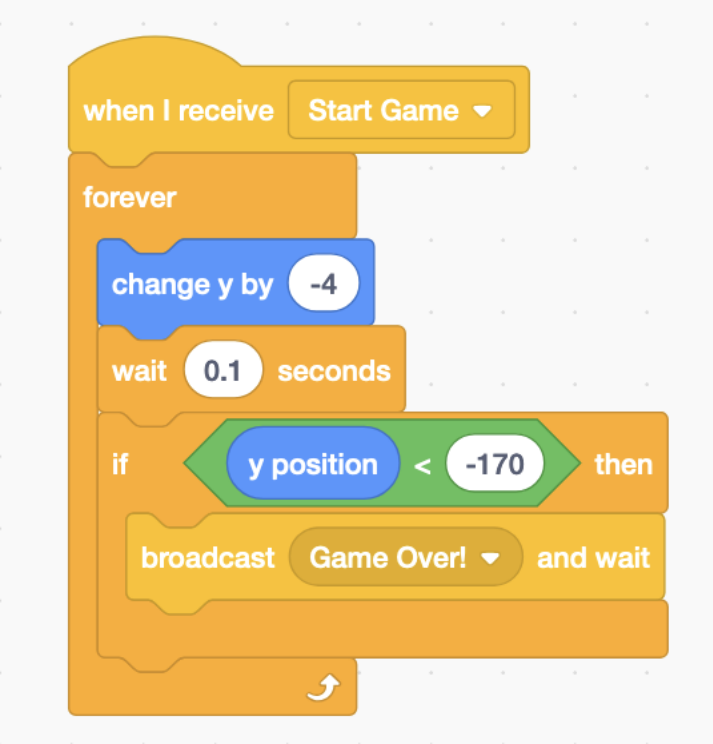



```
while True:
    self.change_y(-4)
```

Main Character: Making Her Fall...

Our Starling currently stays still

Lets convert this Scratch code to Python code

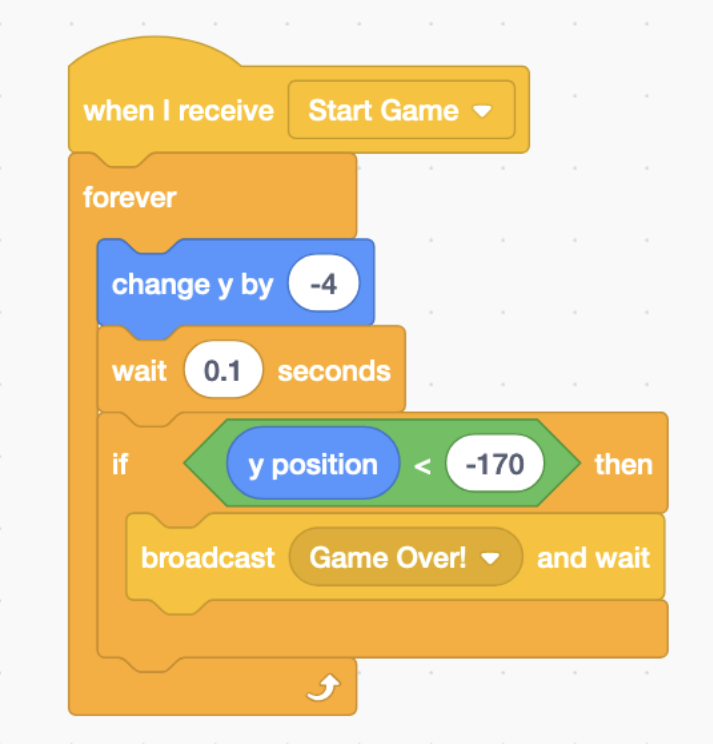



```
while True:
    self.change_y(-4)
    pytch.wait_seconds(0.1)
```

Main Character: Making Her Fall...

Our Starling currently stays still

Lets convert this Scratch code to Python code

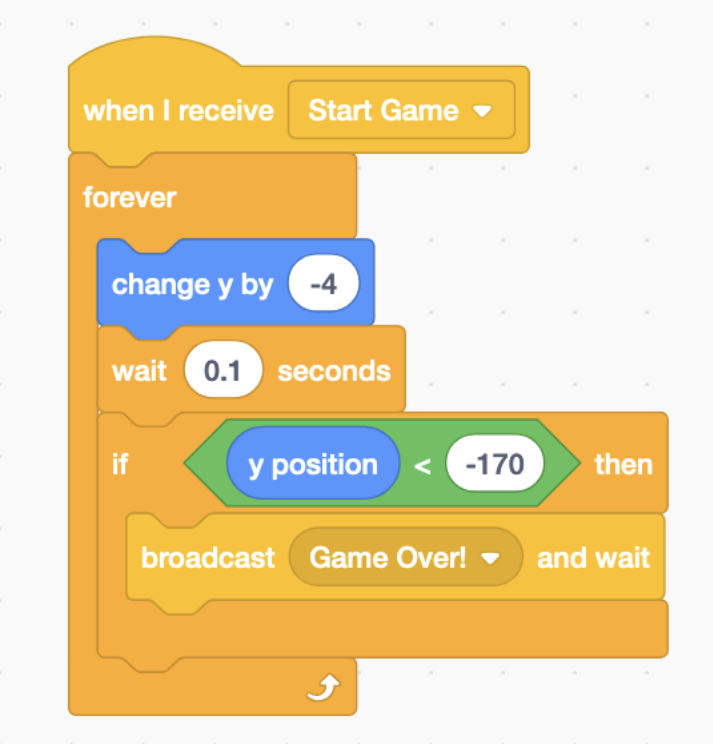



```
while True:
    self.change_y(-4)
    pytch.wait_seconds(0.1)
    if self.y_position < -170:
```

Main Character: Making Her Fall...

Our Starling currently stays still

Lets convert this Scratch code to Python code




```
while True:
    self.change_y(-4)
    pytch.wait_seconds(0.1)
    if self.y_position < -170:
        pytch.broadcast_and_wait('EndGame')
```

Main Character: Making Her Fall...

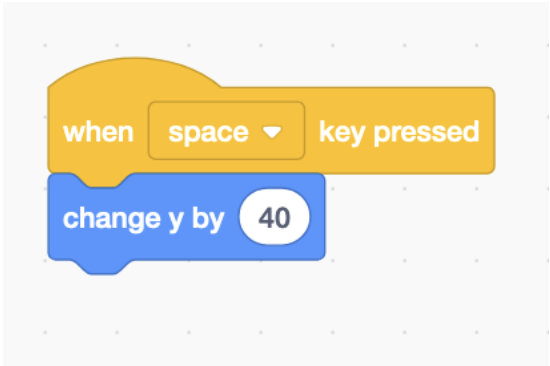
Lets now integrate this code!


```
# Falling code
while True:
    self.change_y(-4)
    pytch.wait_seconds(0.1)
    if self.y_position < -170:
        pytch.broadcast_and_wait('EndGame')
```

Main Character: Making Her Fly!

But now our Starling is just falling!

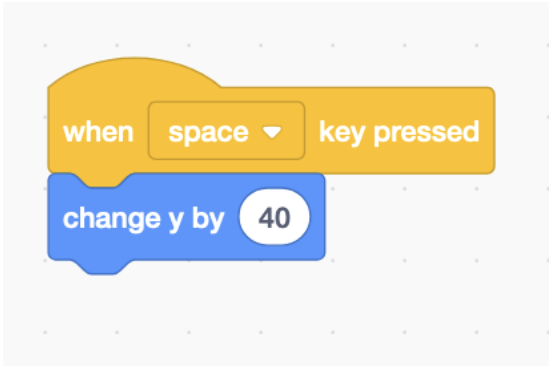
Lets convert this Scratch code to Python code



Main Character: Making Her Fly!

But now our Starling is just falling!

Lets convert this Scratch code to Python code

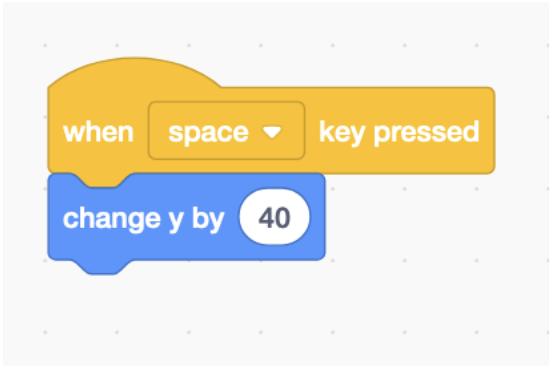



```
self.change_y(40)
```

Main Character: Making Her Fly!

But now our Starling is just falling!

Lets convert this Scratch code to Python code



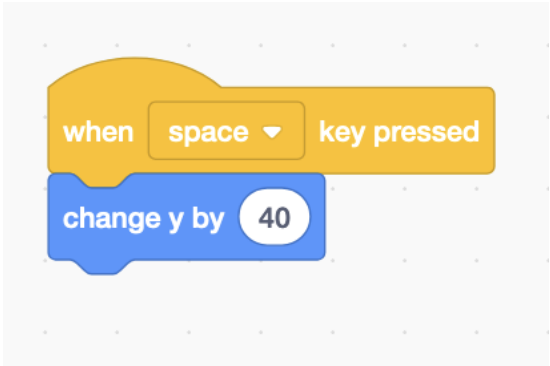
self.chan

Is there a problem with the game?

Main Character: Making Her Fly!

But now our Starling is just falling!

Lets convert this Scratch code to Python code




```
if self.y_position < 180:
        self.change_y(40)
```

Main Character: Making Her Fly!

Let's make her fly by adding this code to our **Flying** function


```
# Flying code
if self.y_position < 180:
        self.change_y(40)
```

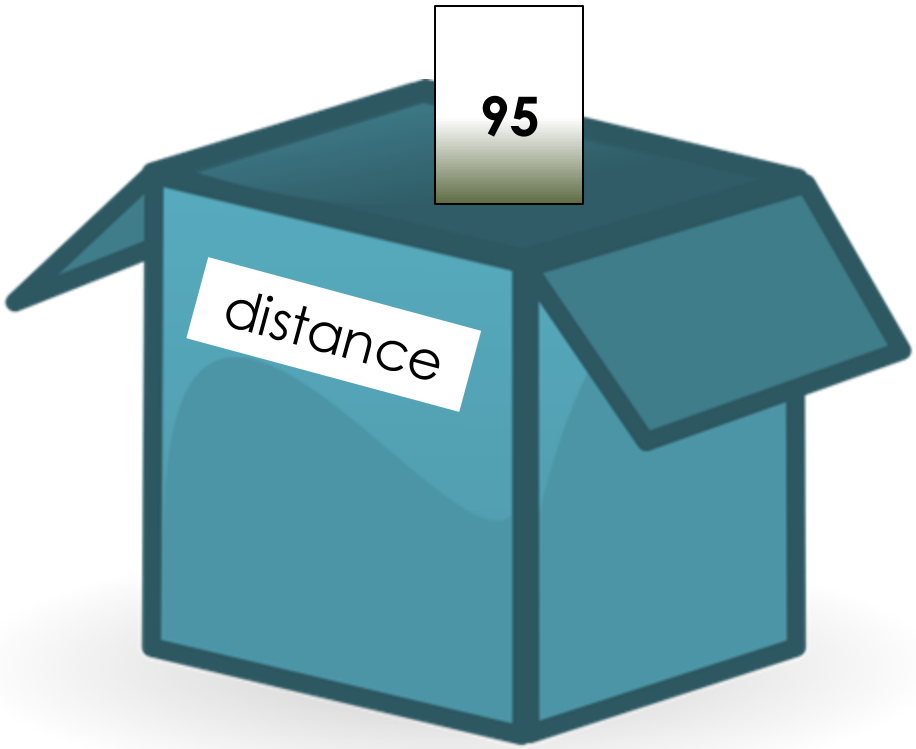
Main Character: Fixing Our Variable!

Now she flies, but our **distanceTravelled** variable never changes.

A **variable** is something that stores data in our program. It is like a box with a label on it.

I can store different things in the box, but the label stays the same.

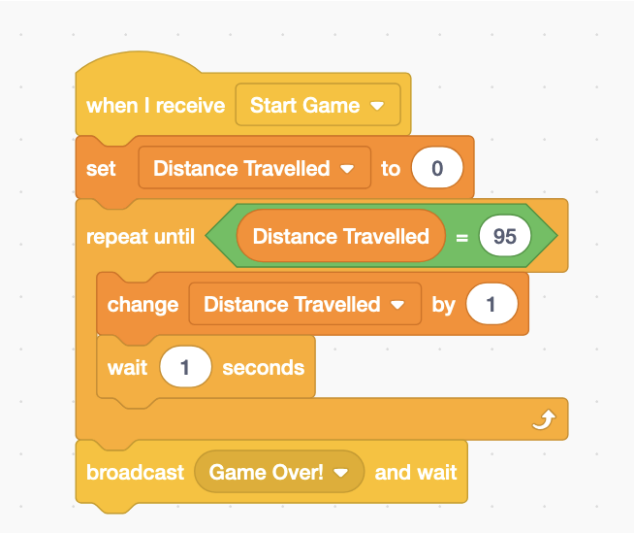
For example, I have stored the number "95" in my **variable** which is labelled "**distanceTravelled**".



Main Character: Making Her Travel!

Now she flies, but our **distanceTravelled** variable never changes

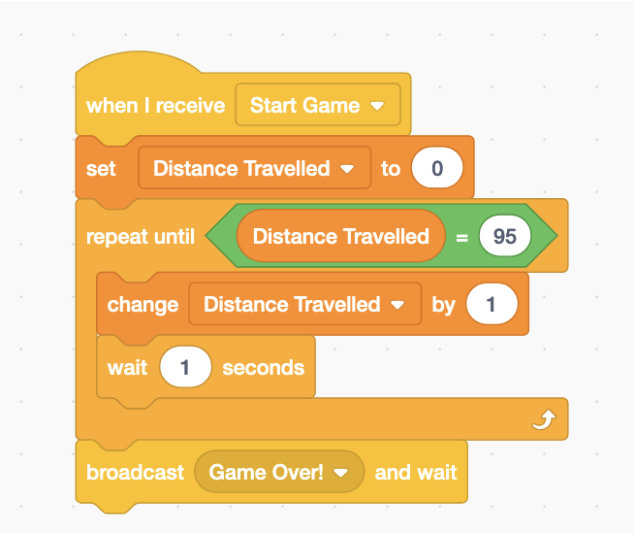
Lets convert this Scratch code to Python code



Main Character: Making Her Travel!

Now she flies, but our **distanceTravelled** variable never changes

Lets convert this Scratch code to Python code

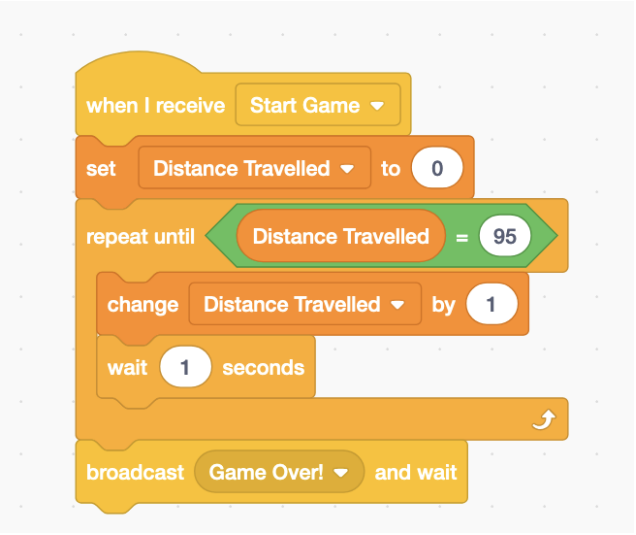


while Stage.DistanceTravelled **<** 95:

Main Character: Making Her Travel!

Now she flies, but our **distanceTravelled** variable never changes

Lets convert this Scratch code to Python code

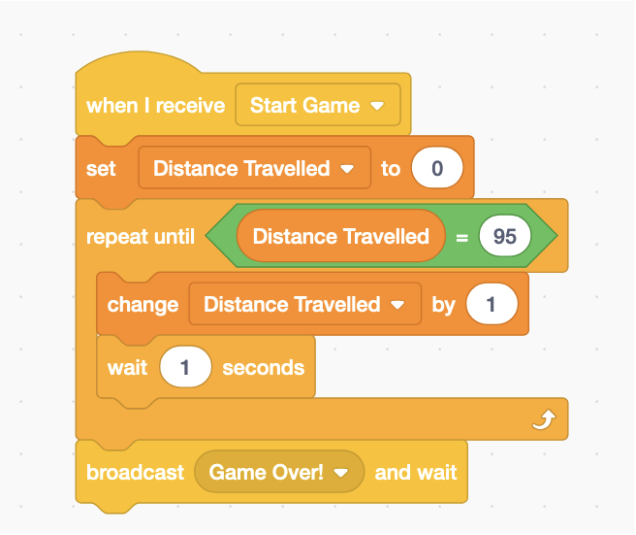



```
while Stage.DistanceTravelled < 95:
    Stage.DistanceTravelled = Stage.DistanceTravelled + 1
```

Main Character: Making Her Travel!

Now she flies, but our **distanceTravelled** variable never changes

Lets convert this Scratch code to Python code

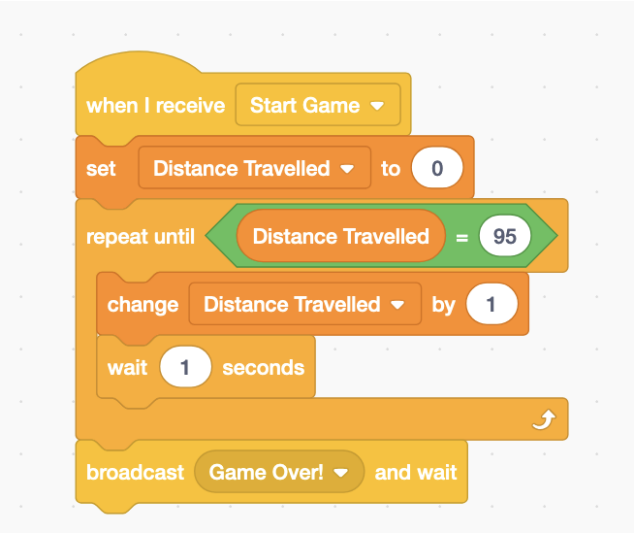



```
while Stage.DistanceTravelled < 95:
    Stage.DistanceTravelled = Stage.DistanceTravelled + 1
    pytch.wait_seconds(1)
```

Main Character: Making Her Travel!

Now she flies, but our **distanceTravelled** variable never changes

Lets convert this Scratch code to Python code




```
while Stage.DistanceTravelled < 95:
    Stage.DistanceTravelled = Stage.DistanceTravelled + 1
    pytch.wait_seconds(1)
pytch.broadcast_and_wait('WinGame')
```

Main Character: Making Her Travel!

Let's fix the score in our **distanceTravelled** function


```
# Distance code
while Stage.DistanceTravelled < 95:
    Stage.DistanceTravelled = Stage.DistanceTravelled + 1
    pytch.wait_seconds(1)
pytch.broadcast_and_wait('WinGame')
```

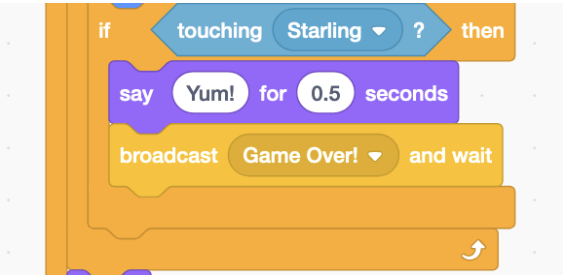
technoteach
technocamps

Creating Our Enemies

Creating our enemies

Now we are finished with our main character, we can move to our **Osprey Class**

There are three things we need to do: **Spawn our enemies**, **Move our enemies,** and **end the game if we are hit**



Spawning Enemies

Now we are finished with our main character, we can move to our **Osprey Class**

Lets convert this Scratch code to Python code

Spawning Enemies

Now we are finished with our main character, we can move to our **Osprey Class**

Lets convert this Scratch code to Python code

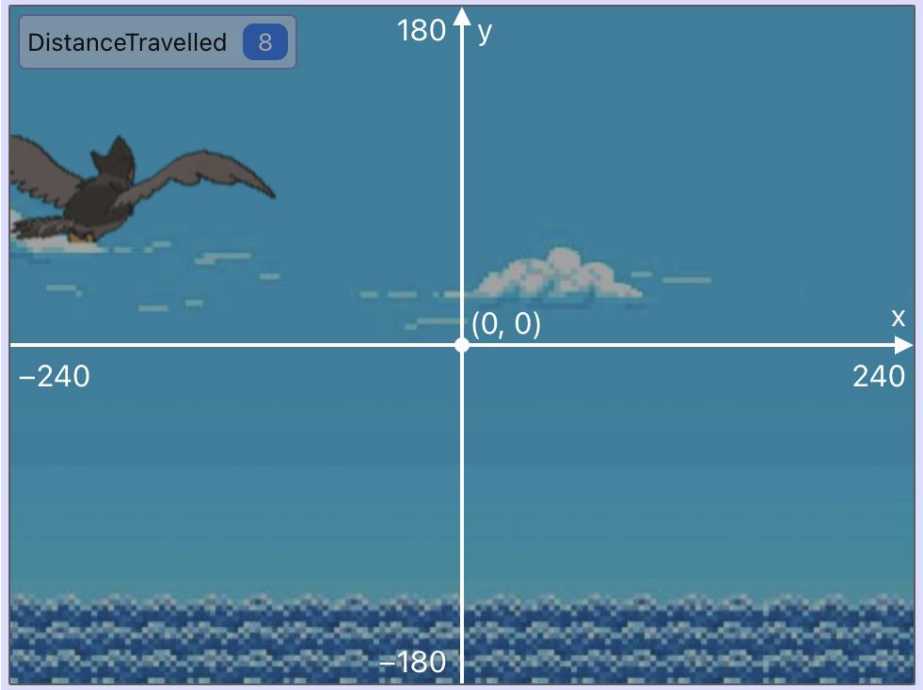
While True:

Spawning Enemies: A Closer Look

When programming games, our sprites must interact with the screen

Therefore, when spawning and moving characters, we need to be aware of the games **coordinate system**

Pytch lets us see this easily



Spawning Enemies

We now want to spawn our enemies in our **Osprey Class**

Lets convert this Scratch code to Python code


```
While True:
        self.go_to_xy(280, random.randint(-180,180))
```

Spawning Enemies

We now want to spawn our enemies in our **Osprey Class**

Lets convert this Scratch code to Python code


```
While True:
        self.go_to_xy(280, random.randint(-180,180))
        self.show()
```

Spawning Enemies

We now want to spawn our enemies in our **Osprey Class**


```
# Spawning code
while True:
    self.go_to_xy(240, random.randint(-180, 180))
    self.show()
```

Moving our Enemies

Now our enemies spawn, lets make them move towards us!

Lets convert this Scratch code to Python code

Moving our Enemies

Now our enemies spawn, lets make them move towards us!

Lets convert this Scratch code to Python code


```
while self.x_position > -220:
```

Moving our Enemies

Now our enemies spawn, lets make them move towards us!

Lets convert this Scratch code to Python code


```
while self.x_position > -220:
    self.change_x(-8)
```

Moving our Enemies

Now lets add this code undeath our spawning code

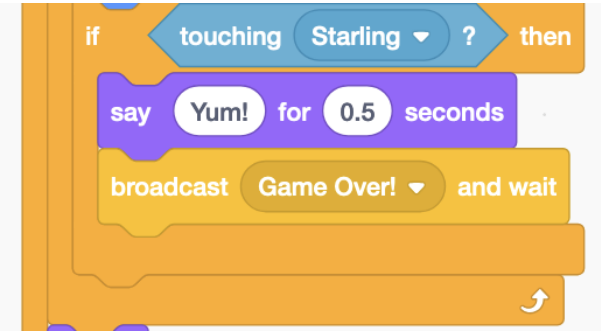
This will **make our enemies move towards us!**

```
...
while self.x_position > -220:
        self.change_x(-1)
```

Defeating Our Starling

Now our enemies spawn, lets make them move towards us!

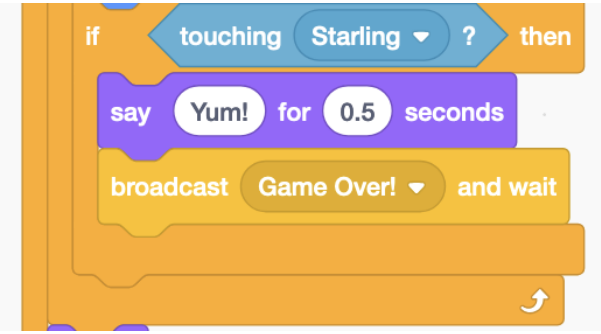
Lets convert this Scratch code to Python code



Defeating Our Starling

Now our enemies spawn, lets make them move towards us!

Lets convert this Scratch code to Python code

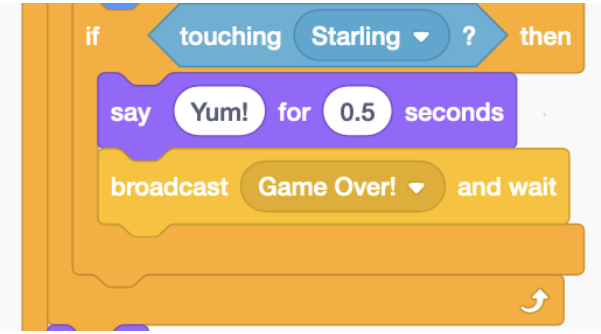



```
if self.touching(Starling):
```

Defeating Our Starling

Now our enemies spawn, lets make them move towards us!

Lets convert this Scratch code to Python code

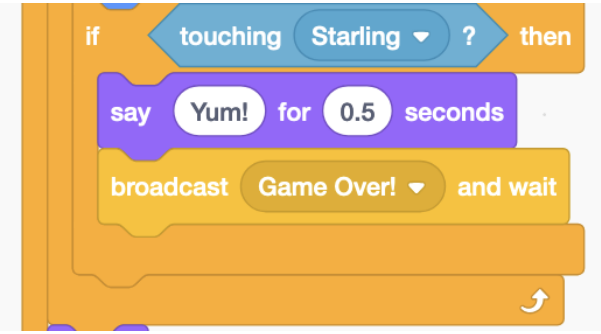



```
if self.touching(Starling):
        self.say_for_seconds("Yum!", 0.5)
```

Defeating Our Starling

Now our enemies spawn, lets make them move towards us!

Lets convert this Scratch code to Python code




```
if self.touching(Starling):
        self.say_for_seconds("Yum!", 0.5)
        pytch.broadcast_and_wait("GameOver")
```

Defeating Our Starling

Now lets add this code undeath our moving code

This will **end the game if we are hit!**

```
...
if self.touching(Starling):
        self.say_for_seconds('Yum!', 1)
        pytch.broadcast_and_wait('EndGame')
```

Complete Enemies Code


```
# Spawning code
while True:
    self.go_to_xy(240, random.randint(-180, 180))
    self.show()
    while self.x_position > -220:
        self.change_x(-1)
        if self.touching(Starling):
            self.say_for_seconds('Yum!', 1)
            pytch.broadcast_and_wait('EndGame')
```

Extensions

Modifying the Initial Difficulty

We can change various aspects of our game to make it a bit harder

Consider changing the following aspects, and experiment with some of the values:

- The speed at which our Starling falls
- The height at which our Starling jumps
- The speed at which the Ospreys move towards us

Adding a Difficulty Curve

We can add a difficulty curve to the game, to make it harder after we have traveled a certain distance

We will need to do the following:

- Turn some of the speeds (Osprey approaching speed, falling speed etc.) into **variables**
- Add an **if statement** that once we get past a certain travel distance, we change these variables
- Add this if statement to every **function** responsible for the variable we are changing

Adding Multiple Enemies

We could make multiple enemies spawn at once, to make the game much harder

Try duplicating the osprey class, and rename it

This will cause multiple enemies to spawn

Is the game now too difficult? Do we need to change anything else?

This is called **balancing**