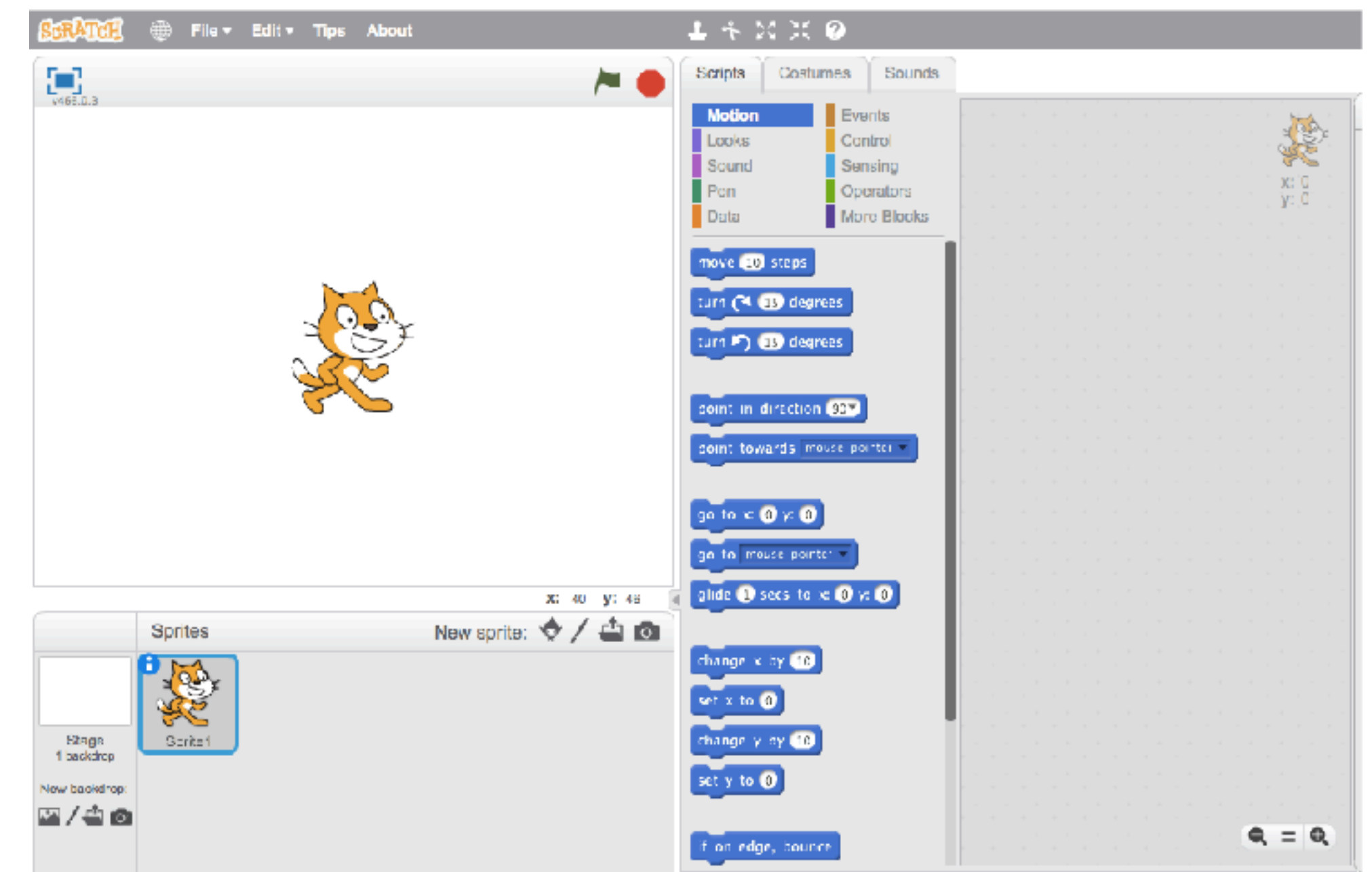


SCRATCH TO PYTHON PROGRAMMING



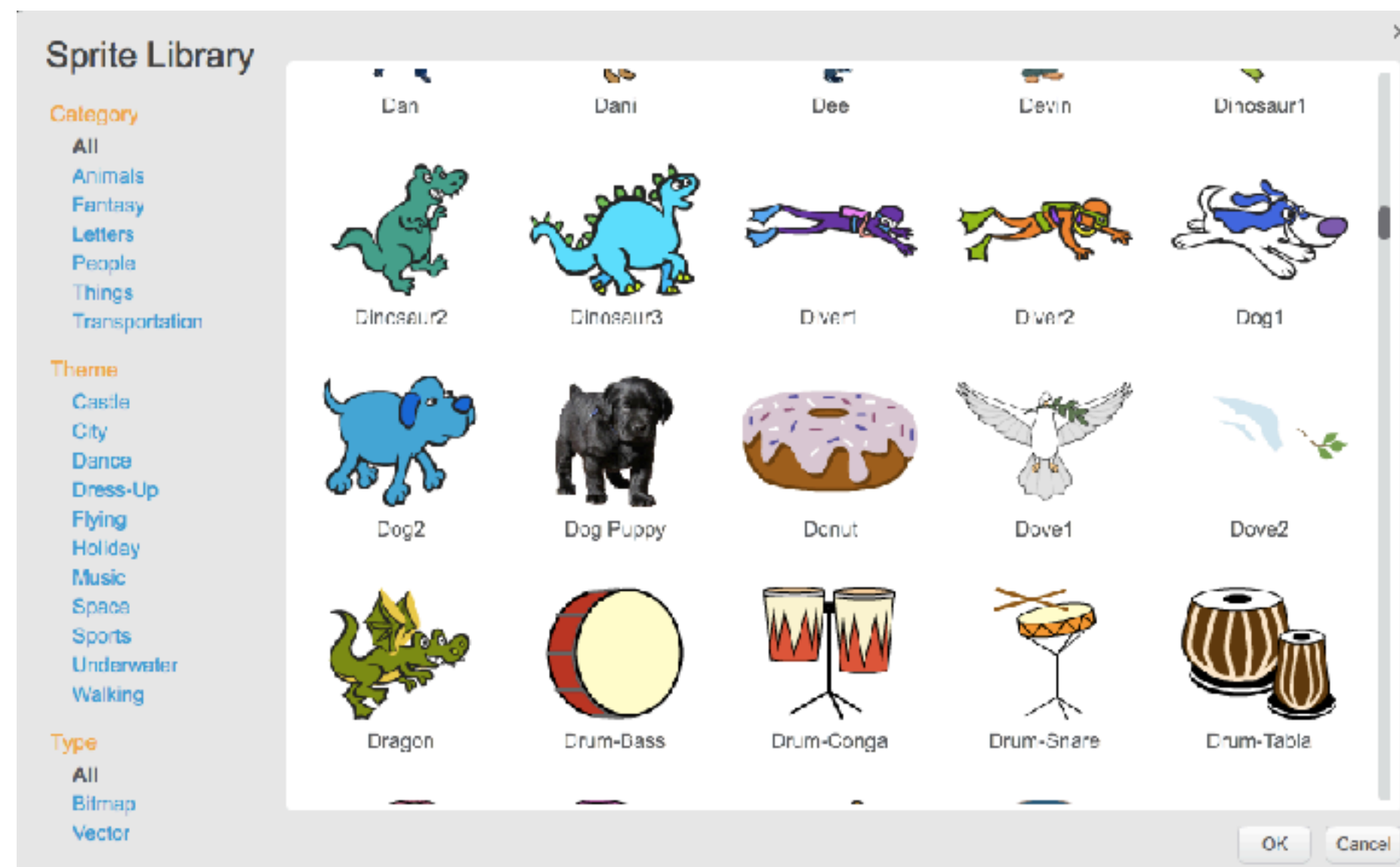
OVERVIEW: SOME QUESTIONS THAT NEED ANSWERING

- Scratch is a good basic tool to get visually pleasing animations running with little working time.
- However we have to ask; "Who actually made scratch?", "How was it made?".
- The programming world is much bigger than Scratch. Real projects use a lot of different languages working together - And a lot of different people working together.
- Are there other languages out there?

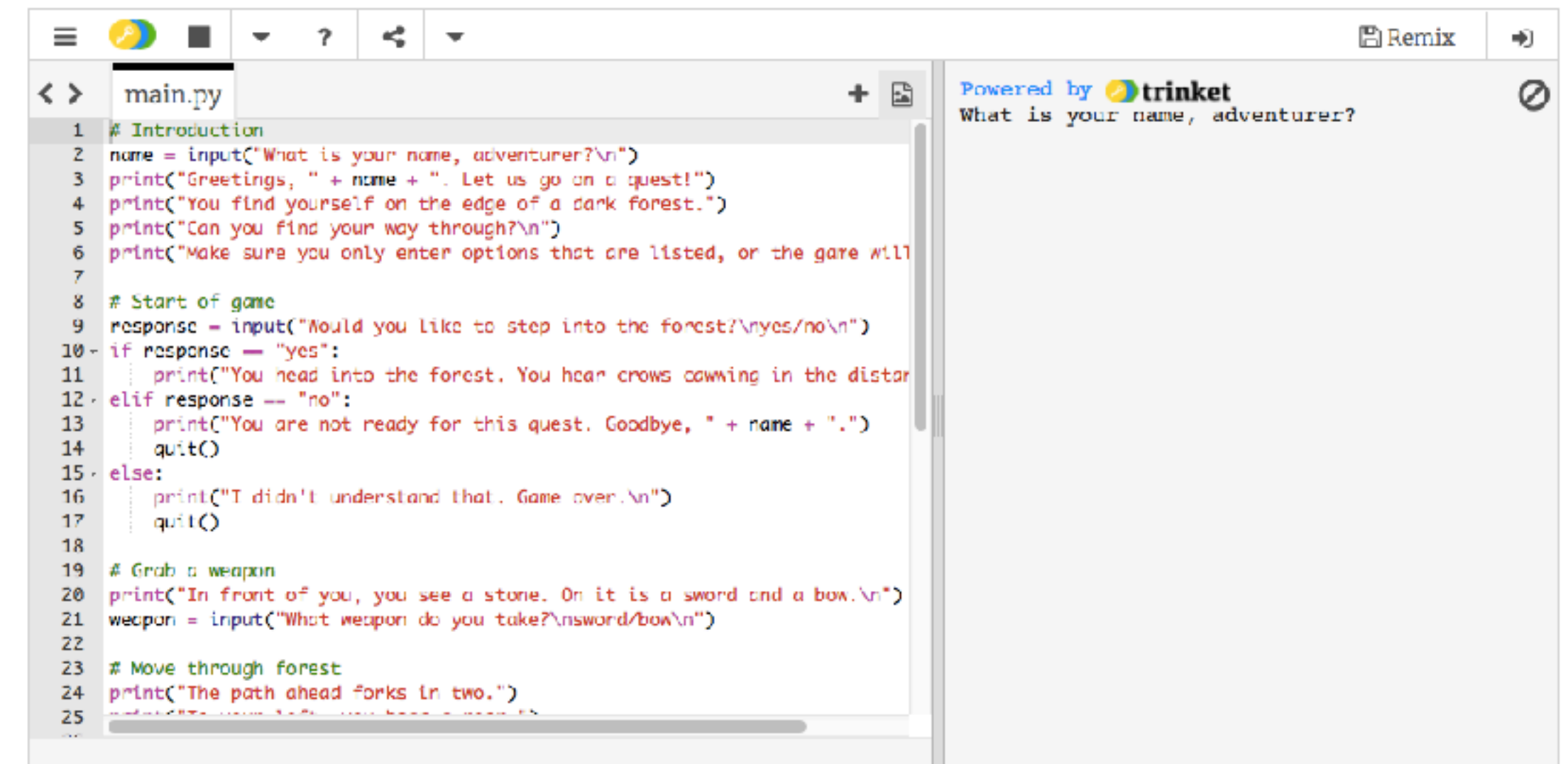


BASIC DIFFERENCES BETWEEN SCRATCH AND PYTHON

In Scratch each sprite has it's own code. Sprites interact with each other through shared resources such as variables and messages. This is often confusing and leads to poorly maintainable code.



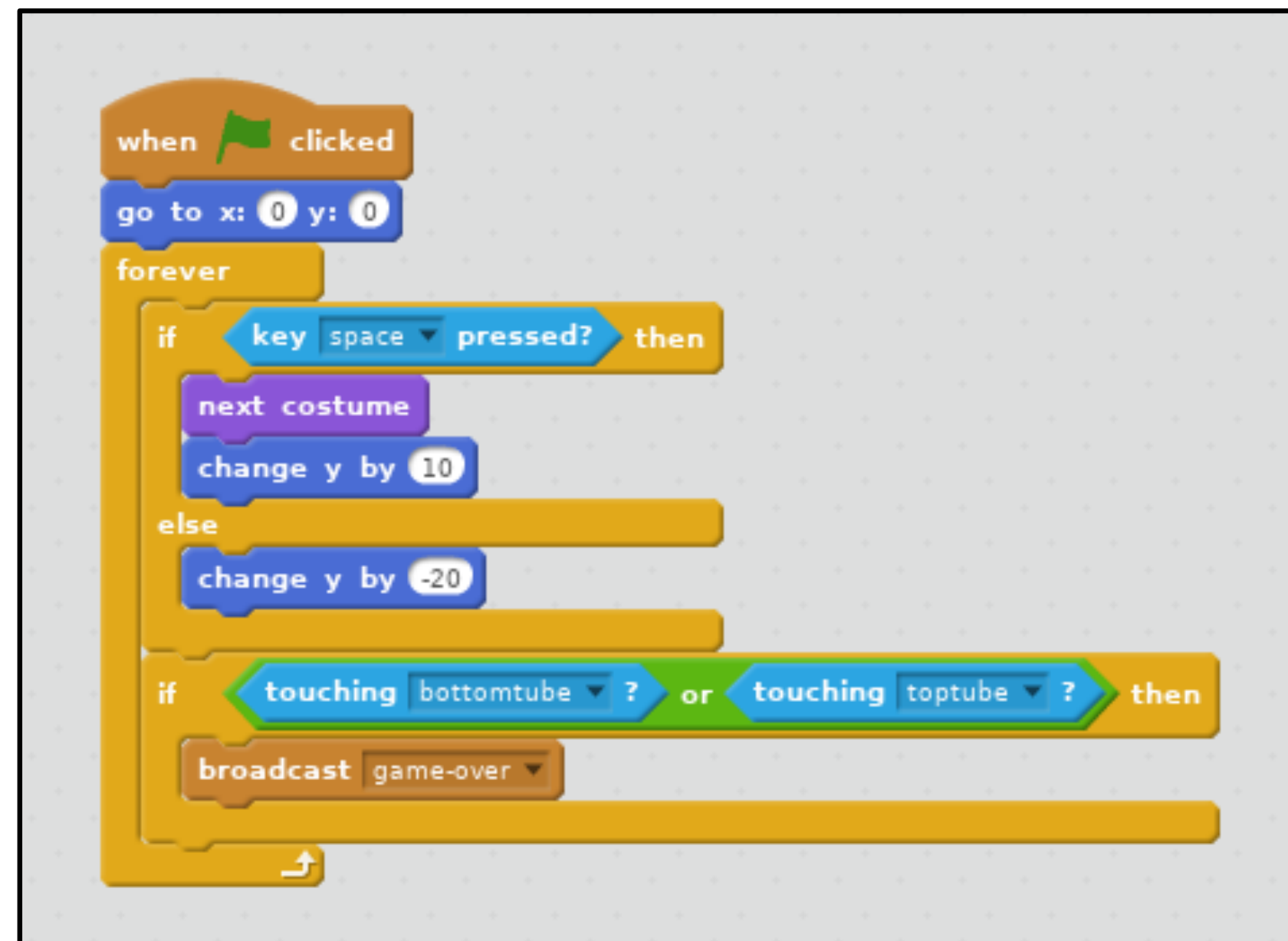
In Python each line of code is read by a Python "Interpreter" and the computer executes the instructions directly and in sequential order.



Expand

BASIC DIFFERENCES BETWEEN SCRATCH AND PYTHON

In Scratch each block is read in sequential order. The code starts with an event or "Hat" block. Placing blocks within outer "C" blocks is used to group blocks together. Each sprite has its own set of blocks which may or may not interact with each other.



In Python every line is read; line by line from top to bottom, reading each line from left to right, evaluating each statement in turn and if necessary assigning its return value to a variable. Indentation groups statements together.

```
1 import turtle
2
3 painter = turtle.Turtle()
4
5 painter.pencolor("blue")
6
7 for i in range(50):
8     painter.forward(50)
9     painter.left(123) # Let's go counterclockwise this time
0
1 painter.pencolor("red")
```

LETS SEE HOW OUR FAMILIAR SCRATCH CODE LOOKS IN PYTHON...



VARIABLES

Scratch:



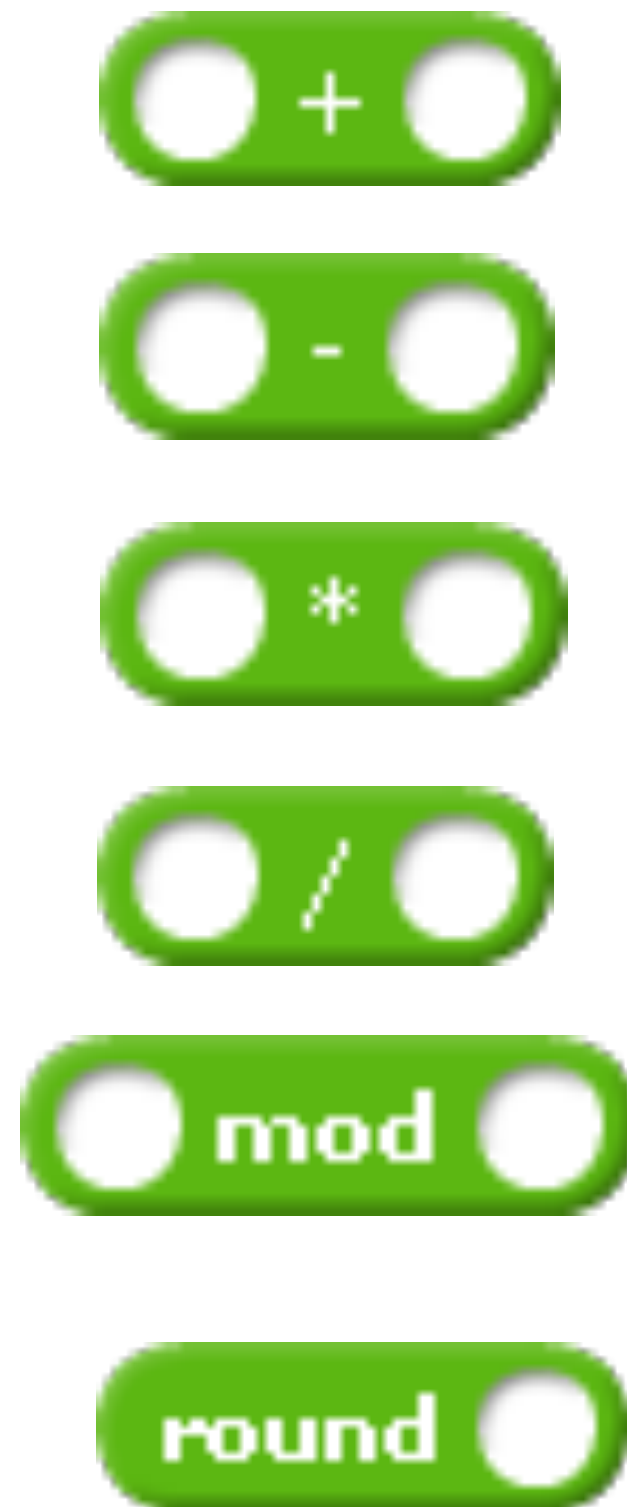
In Python variables have to be given an initial value:

```
variable = variable + 1
```

```
variable = 0
```

ARITHMETIC

Scratch:



In Python the modulus operator or "mod" is symbolised by a "%" symbol. We use it to get the remainder of a division

```
var1 + var2
```

```
var1 - var2
```

```
var1 * var2
```

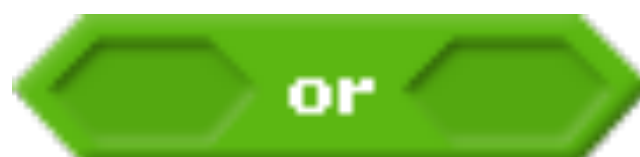
```
var1 / var2
```

```
var1 % var2
```

```
round(var1)
```

OPERATORS

Scratch:



Python:

```
var1 < var2
```

```
var1 == var2
```

```
var1 > var2
```

```
var = (True) and (False)
```

```
var = (True) or (False)
```

```
var = not (False)
```


CONDITIONALS

Scratch:



Python:

```
if (True):  
    # Do something  
else:  
    # Do something else
```

```
if (True):  
    # Do something
```

LOOPS

Scratch:



Python:

```
for i in range(10):  
    # Do something
```

```
while (True):  
    # Do something forever
```

```
while(not condition) # Waits until  
                    # condition is true
```

```
exit()
```

```
while(not condition): # Repeats until condition is true  
    # Do something
```

RANDOM NUMBERS

Scratch:



Python:

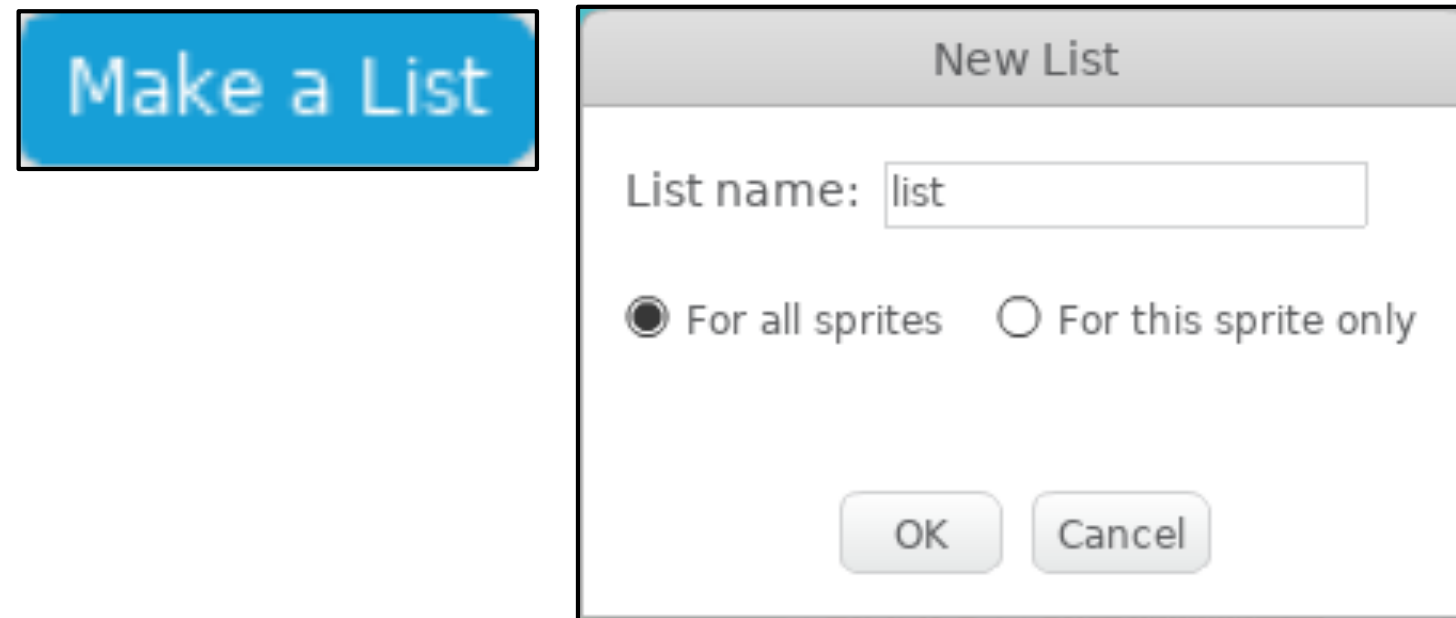
```
import random  
randnum = random.randint(1,10)
```

Random is a whole library of functions.

We have so much more we can do with random numbers in Python than in Scratch!

LISTS

Scratch:



Python:

```
List = []
```

Lists in Python start at index 0

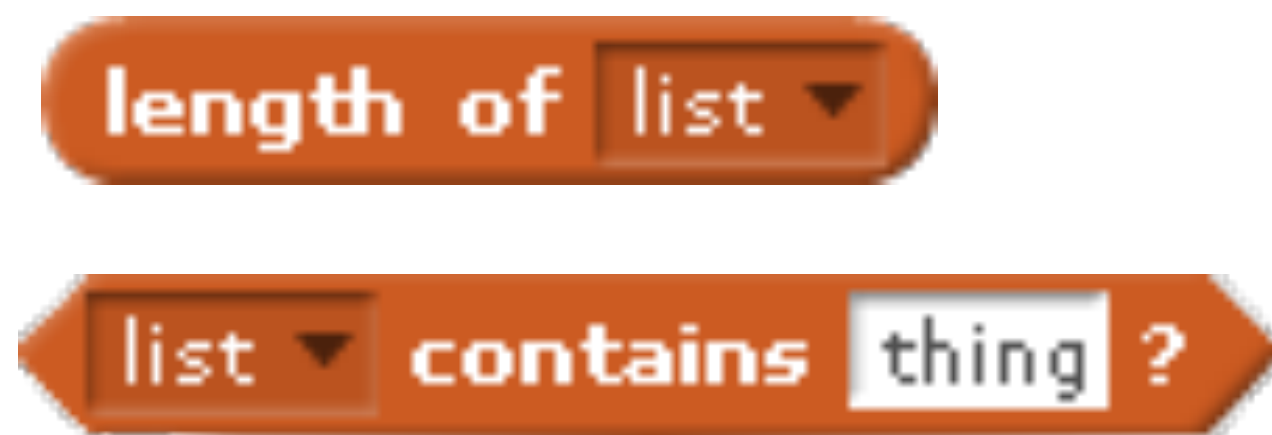
Let's add some objects to our lists



```
List.append(thing)  
del List[0]    or    List.pop(0)  
List.insert(0, thing)  
List[0] = thing  
List[0]
```

LISTS

Scratch:



Python:

```
len(List)
```

```
thing in List
```

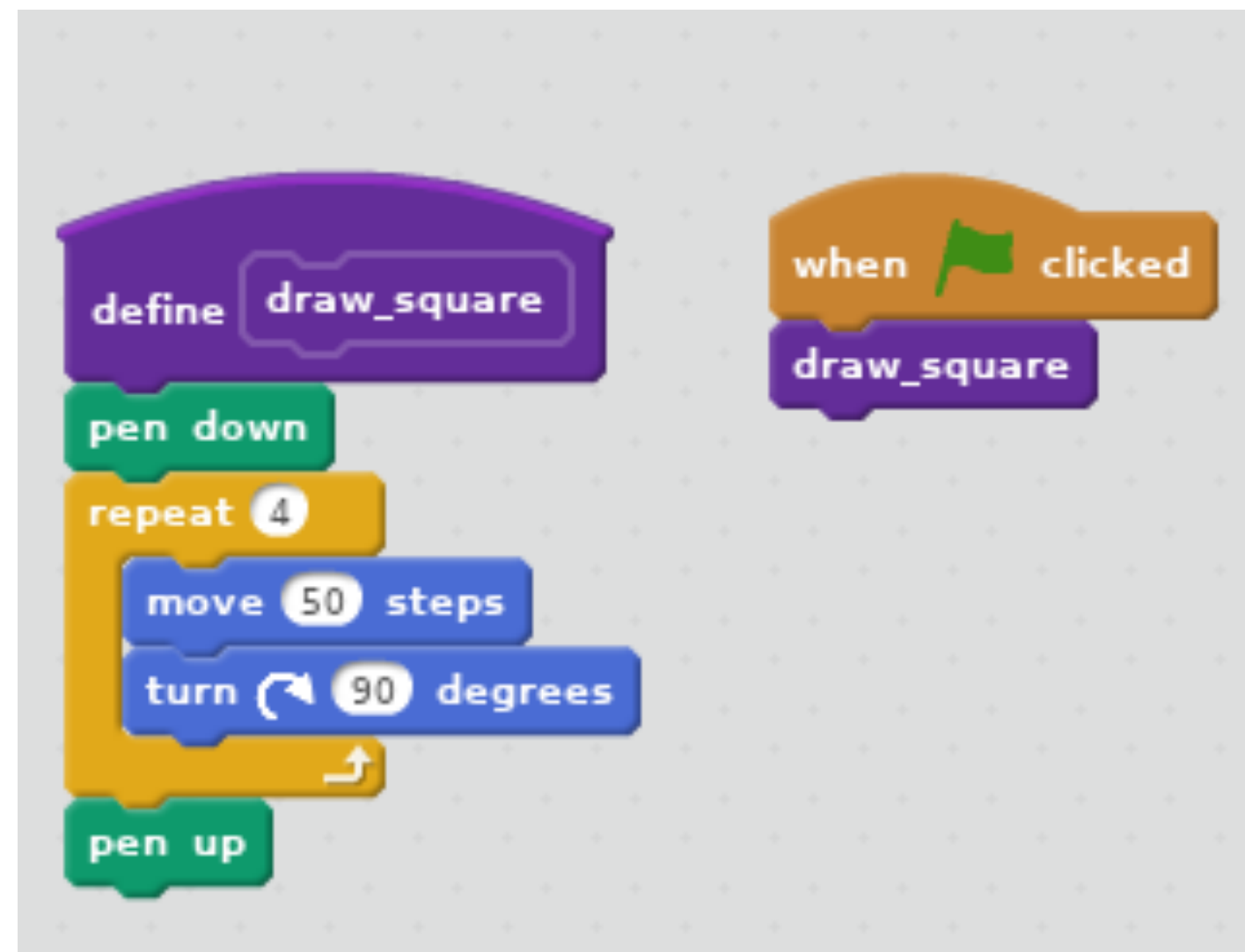
We can do so much more in Python than in Scratch. We can sort the list, reverse the order of the list etc...

```
List.reverse()
```

```
List.sort()
```


FUNCTIONS

Scratch:



Python:

```
def draw_square():
```

```
import turtle
turtle = turtle.Turtle()

def draw_square():
    for i in range(1,4):
        turtle.forward(10)
        turtle.left(90)

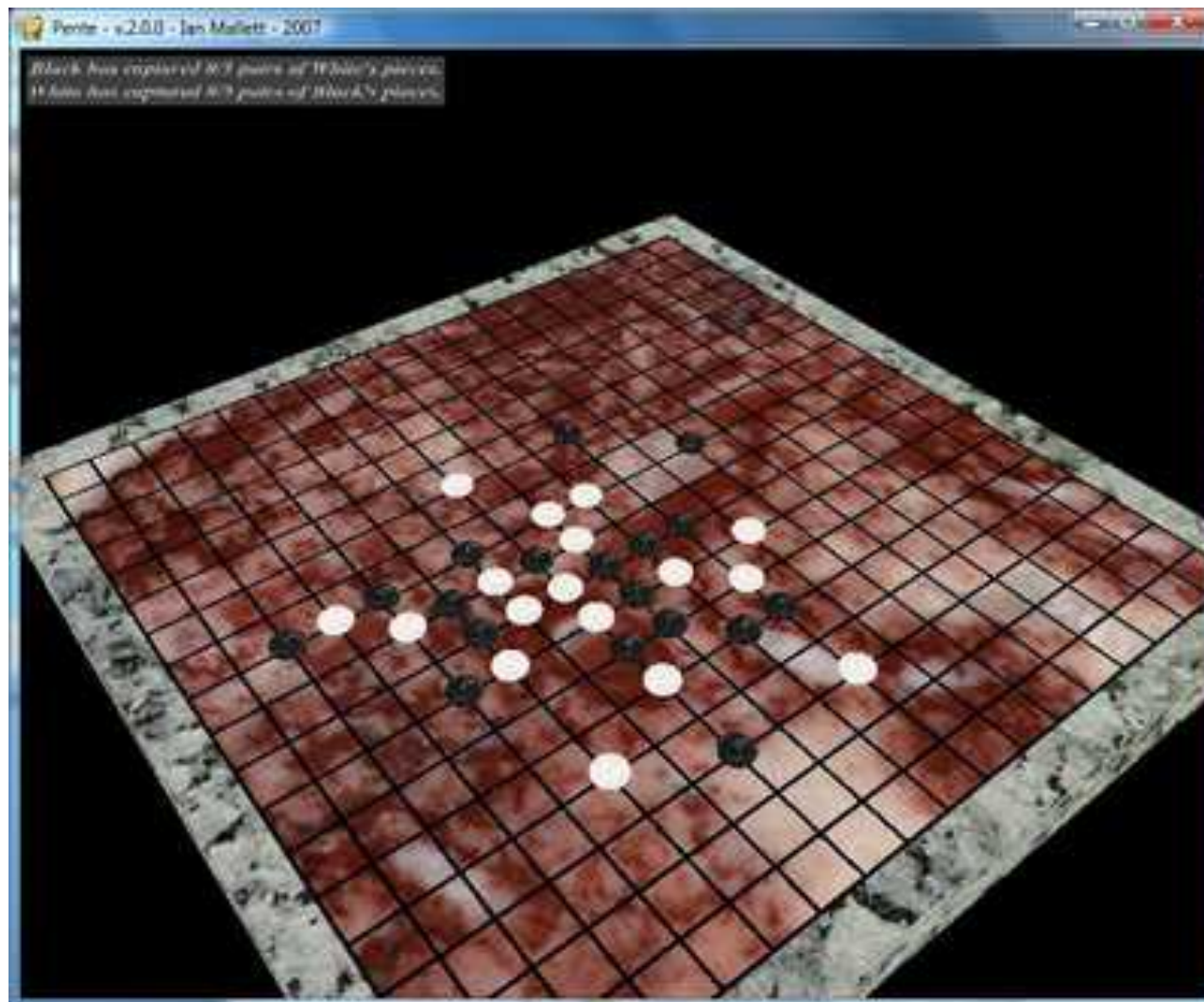
draw_square()
```

"BUT PYTHON IS BORING. I WANT TO MAKE GAMES IN SCRATCH"

You *can* make games in Python.

You could remake the entire Scratch interface using Python if you really wanted to.

Here are some examples of some games that utilise the PyGame library. But you need to build up your skill level and master the basics before you attempt large projects.



WHAT ELSE IS PYTHON USED FOR?

Python is actually not used that often for making games as it has many "resource intensive" features. When used correctly, you will rarely run into limitations with these features and they will in fact speed up how long it takes you to write code!

Python is often used for data analytics in the STEM community although this definitely not the only use for Python.. Some common packages used by these communities include NumPy, SciPy and Matplotlib but there are MANY MORE...

You can write your own library and share with the community...

*This is why we use Python, it is highly extensible and has a massive user base.
It is a good general purpose programming language to learn!*



SOME FAMOUS SERVICES THAT UTILISE PYTHON IN THEIR CODE BASE




TIOBE INDEX FOR VARIOUS PROGRAMMING LANGUAGES IN AUGUST 2017

Aug 2017	Aug 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.961%	-6.05%
2	2		C	6.477%	-4.83%
3	3		C++	5.550%	-0.25%
4	4		C#	4.195%	-0.71%
5	5		Python	3.692%	-0.71%
6	8	⬆	Visual Basic .NET	2.569%	+0.05%
7	6	⬇	PHP	2.293%	-0.88%
8	7	⬇	JavaScript	2.098%	-0.61%
9	9		Perl	1.995%	-0.52%
10	12	⬆	Ruby	1.965%	-0.31%

EMILY DE LA PEÑA

Founder of Coding Kids, Advance Queensland's Community Digital Champion

 codingkids.com.au

 facebook.com/codingkidstutoring

 @EmilyFdelaPena & @coding_kids

 linkedin.com/in/emilydelapena/

 @emily.f.delapena

