A step by step guide to learning Python with mblock

This guide will help you begin your transition from block based programming to writing Python code

Getting started

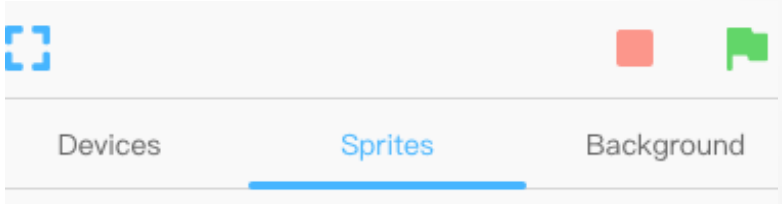
Step 1:

Paste the following link into your browser:

ide.makeblock.com

Step 2:

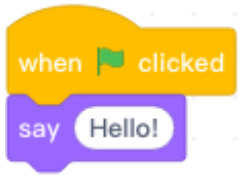
ensure that 'sprites' is selected in the toolbar



(As you walk through this tutorial it is a good idea to add each line of python code as you go along)

What are we going to do?

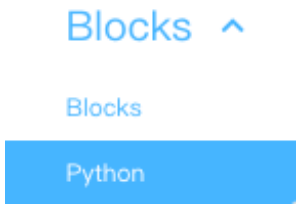
We want to write Python code for the following script:



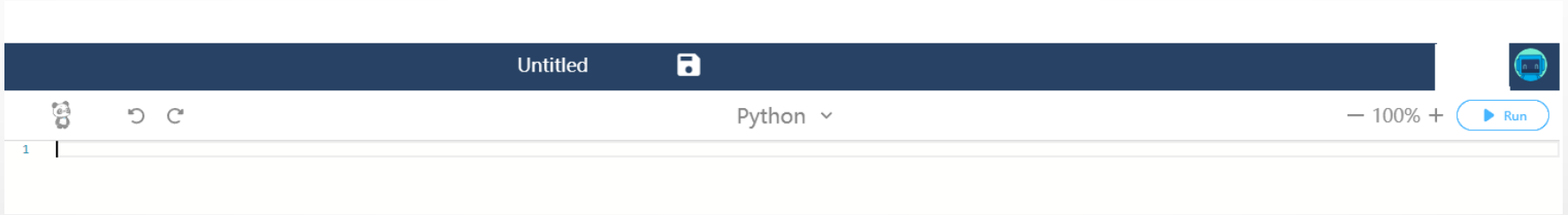
This will cause our panda sprite to say hello

Getting ready to write

Click on the drop down menu at the top of the screen



Select Python. This is the screen you will use to write your code

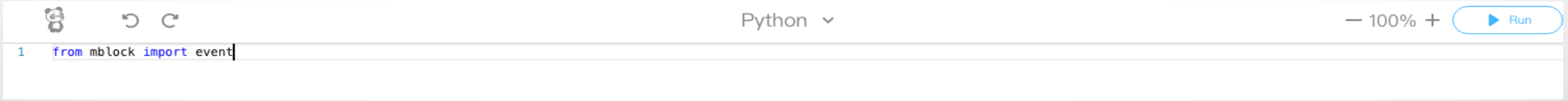


Understanding the first line of code

Although mblock supports Python, it is primarily a block based program. The sprite is being used so the code needs to be imported from mblock.

This means your first line of Python code, when working with a sprite in mblock, will *always* be:

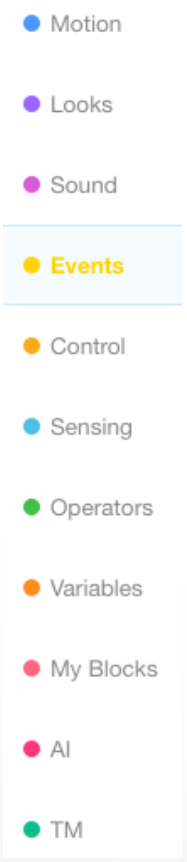
from mblock import



Adding an event

Using blocks, the 

is accessed using the events toolbar in mblock



Therefore we need to include this in the Python code.

An event is being imported from mblock:

from mblock import event

Adding an event

The event in this case is the green flag

This is communicated in Python as follows:

@event.greenflag

```
1  from mblock import event
2  @event.greenflag
```

Python also requires that the event is defined, so it is understood.

This is communicated as follows:

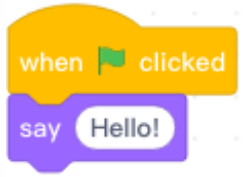
def on_greenflag():

```
1  from mblock import event
2  @event.greenflag
3  def on_greenflag():
```

NB: The parenthesis () are known as a tuple. They are used to execute a sequence of events that will remain unchanged.

Executing a command

When the green flag is clicked, something should happen, the sprite should say hello.



This is communicated in Python as follows:

```
sprite.say('Hello')
```

Important things to note:

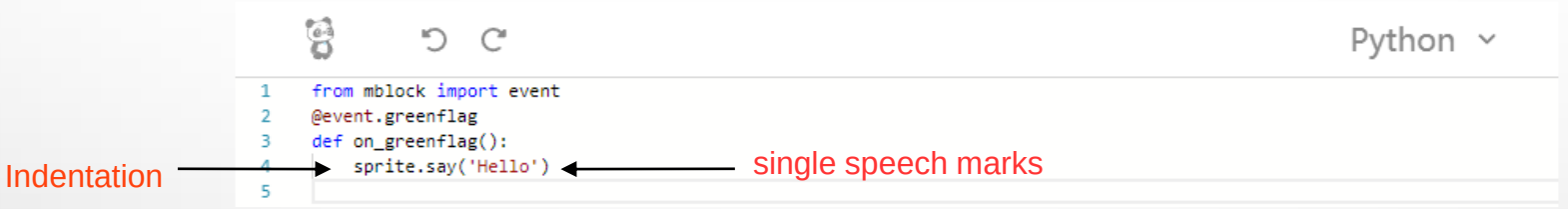
1. Indentation is important in Python, as it shows which block of code a statement belongs to.

```
sprite.say('Hello')
```

Four spaces is standard in Python

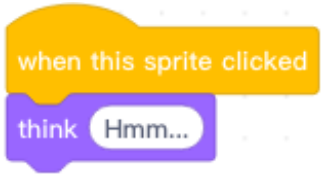
2. Speech should always be place inside single speech marks

```
sprite.say('Hello')
```

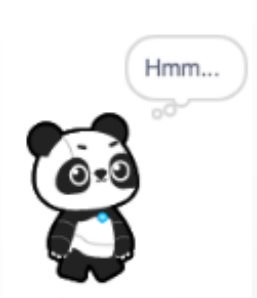


Try it yourself

Try writing you own Python code for the following blocks:



Your panda sprite should show a think bubble with 'hmm' inside when you click on him



Things to remember:

- The event in this case is clicked
- The event will need to be defined
- The speech is hmm

(This Python script follows the same format as the code in the tutorial)

Control blocks

The 

is a control block.

It checks for boolean conditions.

When the condition is true, something happens

When the condition is false, nothing happens

What do we want to do?

Write Python code for the following script

Breaking down the blocks

Like before, Python needs to understand that the code is imported from mBlock.

You have already written the code for the first block!

Python

```
1  from mblock import event
2  @event.greenflag
3  def on_greenflag():
4
5
```

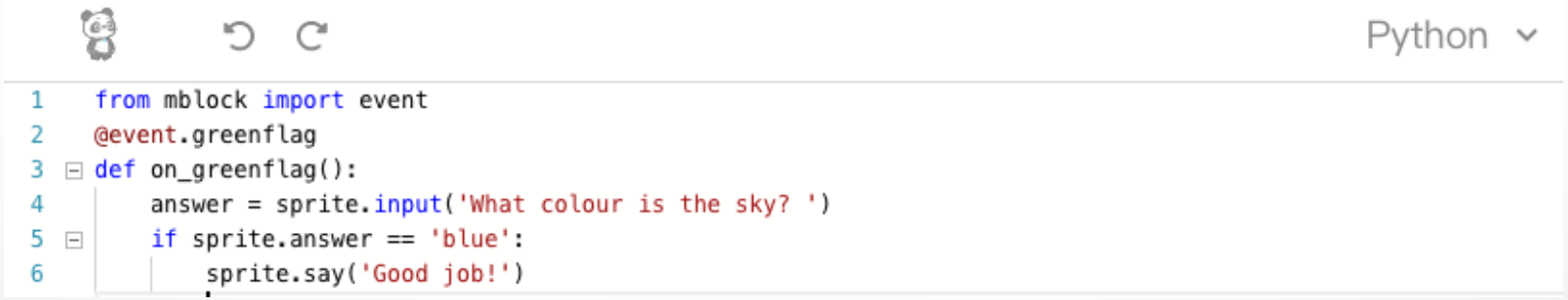
Breaking down the blocks

The next block is considered an input block

The question will be asked by the sprite, meaning an answer will be provided. This needs to be reflected in the Python code

Remember it is a new block so indentation matters!

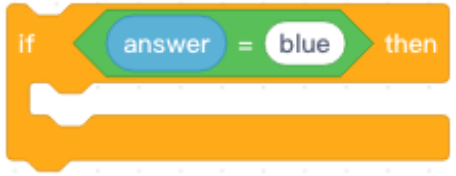
answer = sprite. input('What colour is the sky?')



Breaking down the blocks

Next, we want the sprite to react to the answers given to the question, 'What colour is the sky?'

So, if the answer is blue (which is the colour of the sky), the sprite should do something



== is a comparison operator, it checks for true and false

This is expressed as follows:

if sprite.answer == 'blue' :

Python

```
1  from mblock import event
2  @event.greenflag
3  def on_greenflag():
4      answer = sprite.input('What colour is the sky? ')
5      if sprite.answer == 'blue':
```

**indentation is not necessary as the 'answer =' block is within the 'if ,then' block*

Breaking down the blocks

If the answer to the question 'what colour is the sky?' is blue, we want the sprite to say 'Good job!'

This is expressed as :

sprite.say('Good job!')

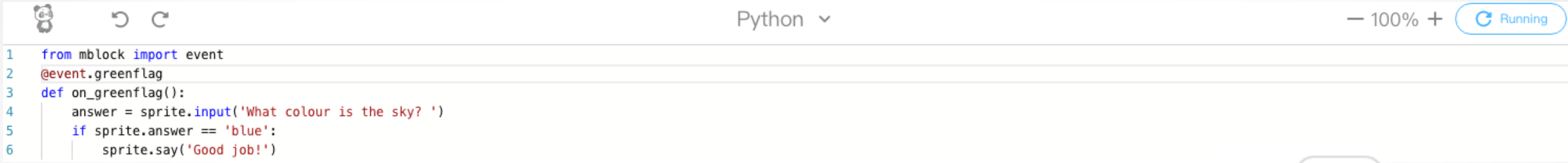
Python

```
1  from mblock import event
2  @event.greenflag
3  def on_greenflag():
4      answer = sprite.input('What colour is the sky? ')
5      if sprite.answer == 'blue':
6          sprite.say('Good job!')
```

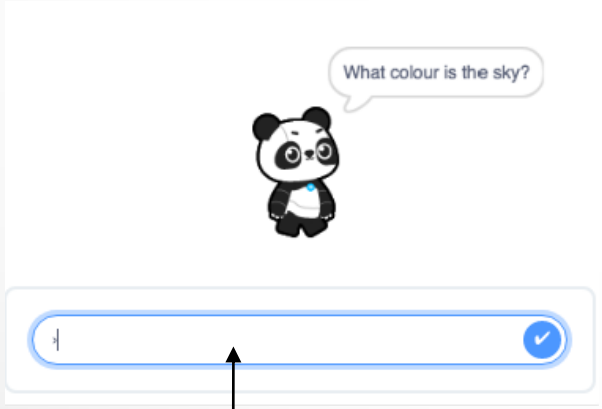
This is a new block and so indentation matters

Running your script

1.Now run your script



2.Click on the green flag



3. Type your answer to the question 'what colour is the sky' here.

4. If your answer is blue, the panda will say 'Good job!'

5. If your answer is anything other than blue, the panda will not react.

Try it yourself

Try to write a Python script for the following blocks:

When the arrow is clicked, your sprite should ask, 'How many days are in a week?

If the answer is 7, he should say 'Well done!'

If any other answer is given, he should say try again

Things to remember:

- you already know how to write the code for the event and define the event
- the if, then, else block is a control block
- if and else are the same block, so should have the same level of indentation
- the answer is 'inside' the if then block and so should also have the same level of indentation as if and else