



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

pytch_



Lesson 5

Finishing off the “Chase game”

Developed by:

pytch.team

<https://pytch.org/>

<https://pytch.scss.tcd.ie/>

Booleans – True/False

- The basic rules of “Booleans” were formulated in 1847 by George Boole (an English mathematician who had moved to Queen’s College Cork)
- A Boolean value can be either False or True. In Python, Boolean values are capitalized: False and True.
- To define a variable with a Boolean value in Python: `my_variable = True`
- Why do we care about Booleans in Pytch? We need Python to make choices and apply different behaviours when a specific condition is True or False

Is the bird touching the star?

`if it is True -> do something`

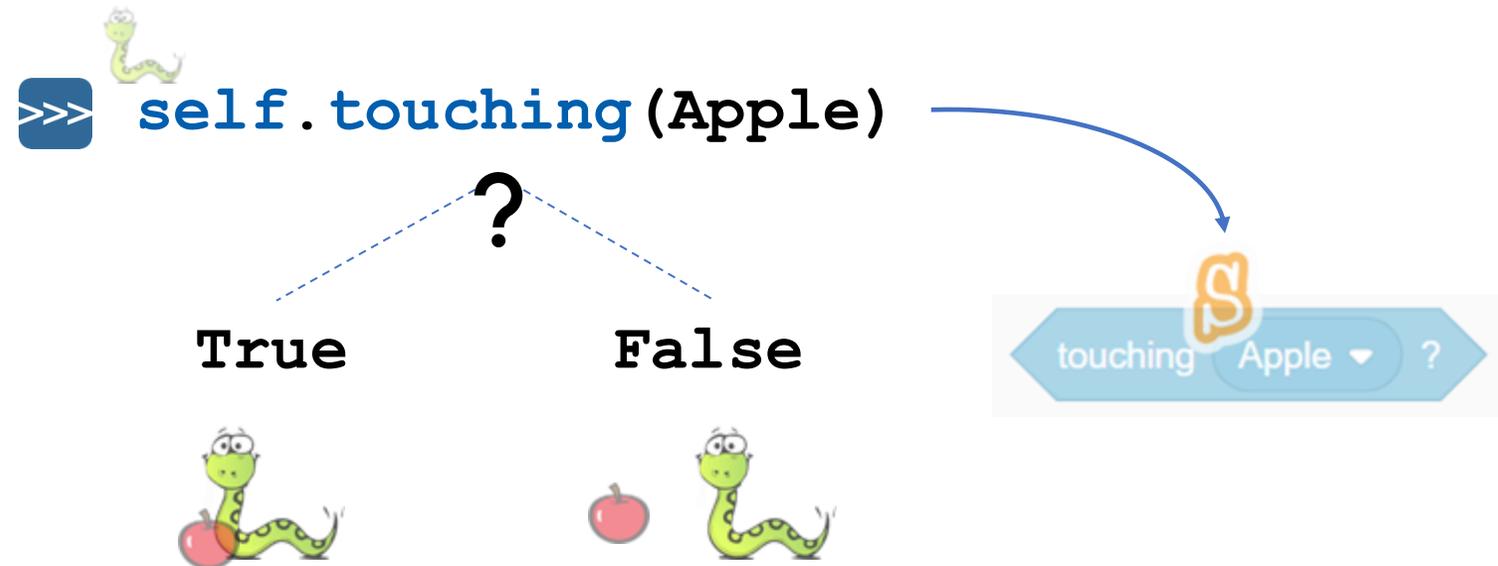
`if it is False -> do something else`



Booleans and Sensing

A Pytch program can report a Boolean value to tell whether it senses certain conditions happening. For example, it can report True or False according to whether the current Sprite is touching another Sprite.

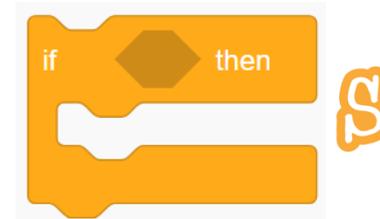
You can write this into a Pytch program any place where a Boolean is needed, and you will get True or False depending on what is happening in the project at that moment.



Python conditions – True/False

The if statement is used for conditional execution of some code. It is used by Python to make choices. If the **condition** is true, the **code contained within the "if" statement** will run.

```
if <condition>:  
    code_to_run_if_condition_true()
```



In Pytch you can use the touching sense with conditionals to ask Pytch to make a choice and run different code depending on the condition.

```
>>> if self.touching(Apple):  
    self.say_for_seconds("An apple!", 2.0)  
    self.glide_to_xy(0, 0, 2)
```



Worksheet 1

Now work in pairs:

- Let's add two new scripts, one in each sprite:
 - The first one to the Bird
 - The second one to the Star
- What do these two new scripts do?
- Write your answers on worksheet 1

[Bird Sprite]



when green flag clicked

```
1 pytorch.wait_seconds(0.1)
2 while True:
3     if self.touching(Star):
4         self.say_for_seconds("Got you!", 2.0)
```

[Star Sprite]



when green flag clicked

```
1 pytorch.wait_seconds(0.1)
2 while True:
3     if self.touching(Bird):
4         self.hide()
```



Try it out

- Follow this link to get a Pytch project that you can run
- Run the program
- Do the Bird and the Star Sprites do *exactly* what you thought they would do?
- If not:
 - Look at the differences
 - Correct your answer on worksheet 1

<https://pytch.org/app/lesson/sbys/5>



Questions to do in pairs – Worksheet 2

1. Why do we need to have the "*while True*" loops in these scripts? What do you think the Sprites would do if we removed them (you can try it and see — you will need to remove the indentation).
(Hint: how many times does the Bird check whether it is touching the Star?)
2. What happens in the Bird's script if you change `while True` to `while False`?
3. What happens if you remove the `pytch.wait_seconds(0.1)` commands?
(Hint: it matters *right at the start* of the program, when you press the green flag).
4. You may have noticed we added a `self.show()` in the Star's script. What do you think this does, and how does it work together with the `self.hide()` in the other script?

<https://pytch.org/app/lesson/sbys/5>



Tasks – Worksheet 3

 14 minutes

This is now a playable GAME, well done! Let's work on making it more fun.

Work in pairs on these activities:

1. Add a variable for the duration of the star movement with a random value from 1 up to 3 — change that variable each time the star glides to a random location.
2. Make the Bird and the Star a little smaller than they are now.
3. Make the game more difficult by decreasing the bird speed each time it collects a star.
 - You will have to add some code in the `while True – if self.touching(Star)` section
 - To decrease the speed, you can update the Bird's speed variable: `self.speed = self.speed * 0.9`

Extension

Finished early? Challenges:

1. Can you add a variable “score” to keep track of how often the player catches the star? Make the bird say the score (instead of "Got you!") during the game.
2. Add a Sprite for the player to *avoid* — if you bump into it, you lose a point from your score.

+

To debug your code, as well as “saying” a value, you can use the Python function `print(value)` to ask Python to tell you the value. Pytch will show you the value in the “Output” window, underneath the scripts. For example, if you write `print(self.score)`, Python will print the value of the score there.



Recap

Today we have

1. Learned about Booleans
2. Python conditionals (“if”)
3. Pytch touching sense methods
4. Finished off the creation of your Pytch playable game!

In the next lesson we will learn more about Python and Pytch and we will start a new Pytch project: “Catch the apple”.

