

## Pre-course documentation

Welcome to the module Introducing Python Coding with Pytch. This document gives an overview of the module and how each lesson is structured.

Pytch is a web-based environment that allows users to write Python programs. It adds the kinds of Sprites, graphics, sounds, and interactivity that students may be familiar with from MIT's Scratch language, although no previous experience with Scratch is required. There is a short intro video that shows off the basic features of Pytch in the General section at the top of the module. Pytch is a research project run out of Trinity College Dublin – there is a separate document describing how you could assist us with our research if you so wish. You can learn more about the Pytch project at our project home page: <https://pytch.scss.tcd.ie/>

Your students will need laptops or chromebooks with internet access to use Pytch (iPads are not really suitable as the on-screen keyboard takes too much space). If you can access the website <https://pytch.org/> that's all you need. There is an intro video in the General section that shows the basics of the main Pytch interface. Each lesson will give the students a link to follow to access the relevant code.

Each of our lessons in this module follows the PRIMM approach where we work in stages, first **P**redicting what a program will do, then **R**unning it to learn more, then **I**nvestigating more deeply prompted by questions, and finally **M**odifying the program and possibly **M**aking our own program. You can read more about PRIMM here: <https://suesentance.net/primm-project/>

This course consists of seven weeks of structured lessons, with a quiz for the eighth week. We have also included prompts for a portfolio piece/independent project in the week eight folder, if you wish to assess students that way instead.

## Lesson plans

Each lesson is structured around the PRIMM pedagogy, following the same basic structure:

- A set of Powerpoint slides, and an optional video which presents the slides. The video has (usually four) points in it where you can pause the video for students to work.
- Three worksheets for each lesson, one for the Predict and Run sections, one for the Investigate section, and one for the Modify and Make section. We recommend having students work in pairs or small groups so that they can discuss the work.
- Worksheets with sample solutions are also on the Moodle page, hidden from students. You can use these as a grading guide if you wish, but we recommend allowing students to treat them as formative assessment; you may still find the solutions useful (or even un-hide them, so students can see our solutions). If you wish to run the code from the sample answers on [Pytch](https://pytch.org/), we recommend you copy and paste the code from the Word versions of the solutions, in the “Alternative formats” folder.

Each lesson begins with a short presentation of some aspects of the curriculum which the students will learn (e.g., statements, variables). There is a script you may wish to use (in the speakers notes in the Powerpoint versions of the files) if you don’t want to use the videos.

There is then a **Predict** activity where a program (or program fragment) is shown, and the students must write their predictions for what it does using **worksheet 1**. (If you are using the videos, you pause the video here while they work – there is a ten-second countdown in the video presentation at this point to give you time to pause it.)

After this (restart the video, if appropriate), there is a **Run** activity. On the slide there is a URL which takes the student to the Pytch website with the “predict” code pre-loaded. They can run the program and **update worksheet 1** to reflect their new understanding. Again, pause the video here (there is a timer and a prompt in the video at the appropriate moment).

Next, students are given a set of prompt questions to **Investigate** the code, writing their answers on worksheet 2. Again, pause the video here (it has a timer and prompt). Lesson 2 is an exception – there are two blocks of questions one after the other (they do not fit on one slide). We have broken them into two rounds of “pause and answer”, but you can present both sets of questions and have the students work on them in one block if preferred.

Next, students are given a **Modify and Make** activity. (In most of the lessons we have merged these two steps of PRIMM, as the students generally don’t have scope to do a full making activity, though the portfolio piece prompt in lesson 8 can serve this role.) Students are given some prompts to improve the program and can work (individually or in pairs) to add features. Each week there are also *extension activities* (optional extra tasks) for students who finish early, or for classes which have more time to explore. The students are prompted to copy and paste their final programs into **worksheet 3**. It is not necessary that each student fully completes each task, but they should all attempt at least the first.

The lesson wraps up with a single-slide recap of the lesson contents.