

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 system, 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.

Each of our lessons in this module follows the PRIMM approach where we work in stages:

- **P**redicting what a program will do;
- **R**unning it to learn more;
- **I**nvestigating more deeply, prompted by questions;
- **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, and an eighth week of a multiple-choice quiz, for which solutions are provided. We have also included prompts for a portfolio piece/independent project in the “Week 8” folder, if you prefer.

Lesson plans

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

- There is a set of powerpoint slides. (We anticipate in future providing a video which presents the slides. The video will have (usually four) points in it where you can pause the video for students to work.)
- There are 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 provided. You can use these as a grading guide if you wish to grade the worksheets, but we recommend allowing students to treat them as formative assessment; you may still find the solutions useful (and may even share them to let students see our versions of the solutions).

Each lesson begins with a short presentation of something the students will learn (statements, variables, etc).

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**.

After this, there is a **Run** activity. On the slide there is a URL which will take the student to the Pytch web site with the “predict” code pre-loaded. They can run the program and **update Worksheet 1** to reflect their new understanding.

Next, students are given a set of prompts and questions to **Investigate** the code. They write their answers on **Worksheet 2**. Lesson 2 is an exception here — there are two blocks of questions one after the other (the questions do not fit on one slide). We have broken them into two rounds of “pause and answer”, but feel free to present both sets of questions and have the students work on them in one block if you prefer.

Next, student are given a **Modify and Make** activity. (In most of the lessons we have merged these two steps of PRIMM since the students generally don’t have enough scope to do a full making activity, though the portfolio piece prompt in lesson 8 can serve this role as well.) 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 cut and paste their final programs into **Worksheet 3**. It is not necessary that each student fully complete each task, but they should all attempt at least the first.

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