

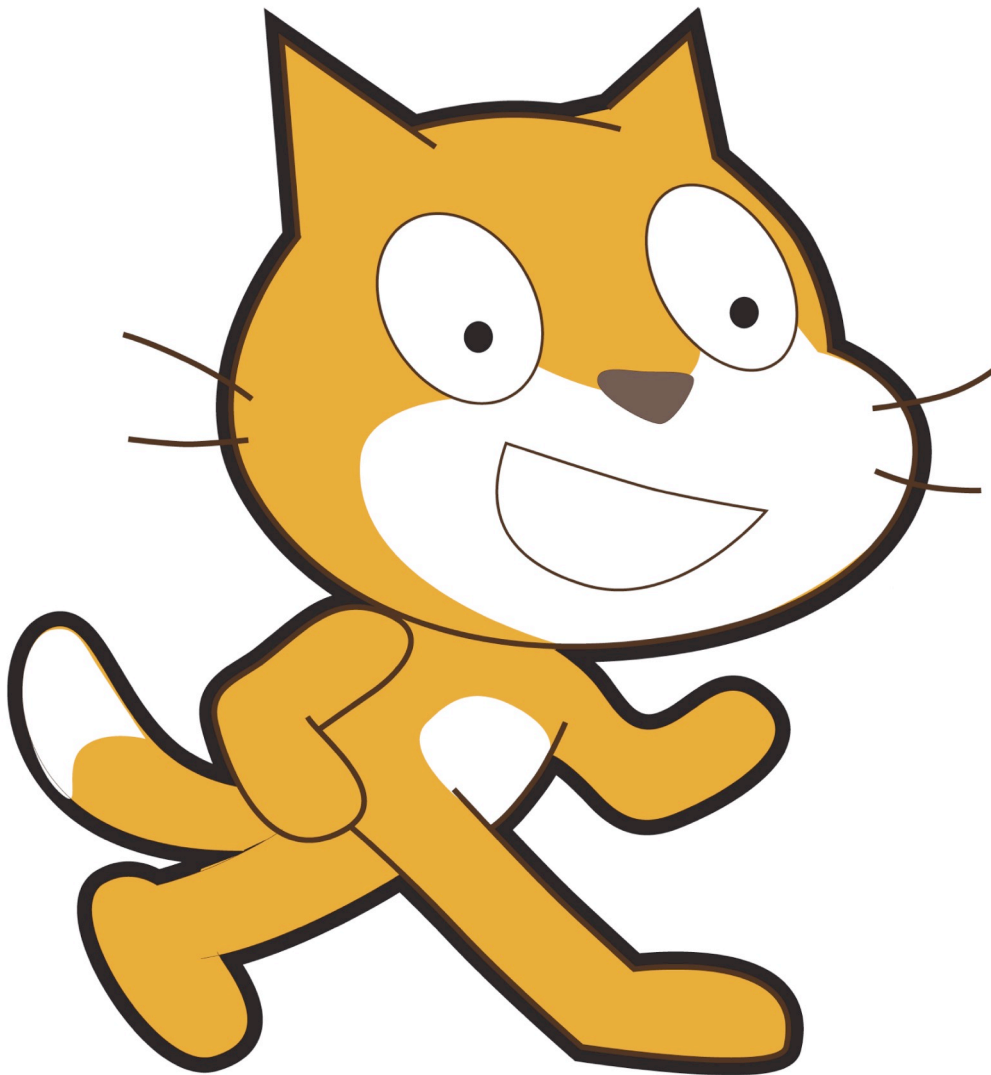
technocamps

Inspiring | Creative | Fun

Ysbrydoledig | Creadigol | Hwyl

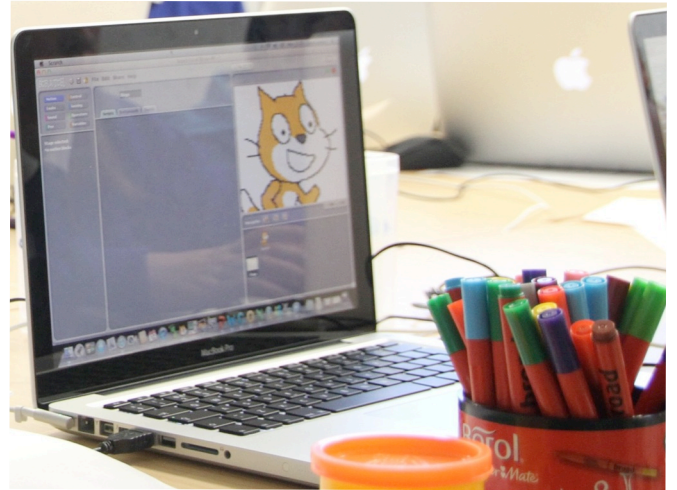


Scratch Workshop Content



Scratch

The benefit of the ‘drag-and-drop’ approach to programming is the lack of pressure to learn a programming language’s syntax. Scratch provides just that. In addition it also provides a friendly user-interface, attractive to both experienced users and newcomers, as well as the range of assistance available to users ranging from forums and help guides, to demo games, example code and helpful videos available on their webpage.



Despite this lack of a feel for a ‘real programming environment’, any user of the software will begin to understand some basic coding constructs required to program such as rules and iteration, these fundamental aspects of programming can be applied later to also more experienced users or coding environments.

Use of basic coding constructs will assist further learning of programming in general and can be applied to any given programming language. They are fundamental basics, this drag-and-drop style approach is a good introduction into understanding these elements.

Scratch is a colourful user friendly open source (free) software popular within educating ICT and Computing. It incorporates not only the basics of computer science, physics and maths but most importantly has a certain “fun factor” with the use of sound, imagery and the editing tools.

To begin we discuss some of the basics of using Scratch, for instance highlighting which areas of the interface contain the code to be used, where they are dragged to and how to access some of the image editing tools. Your “Sprite” (an object within your project) can be deleted, edited or even a new one can be imported or drawn from scratch.

Blocks of code can be dragged into the main script, these can apply movement to your Sprite and lay out the rules to your project and express any conditional statements needed.

The most important of the instruction categories is the “control”, all instructions should either begin with a control or should be embedded within a control statement such as the use of loops or conditions.

Scratch

The production of a project should cover the majority of the basics required to use the Scratch software to its full potential. This could even be taken further and incorporate more complex dynamics such as gravity.

The following features of a project tend to review all the important aspects of Scratch to then gain enough confidence to try it out yourself:

- Creating a main character and editing the background using the familiar Paint Editor available.
- Adding additional sprites that can be hidden or made visible. These can be used as collectables for a game. The implementation of enemies or obstacles to ensure the game is of a challenge to the user. These can provide their own movements without user interaction and also include the variables category to create a score.
- You can then progress onto the use of broadcasting to send messages in your game e.g. telling your game to end if the score becomes zero.

Further features can easily be added to improve the overall quality of each of your projects, these can also be applied to use of levels or even buttons on your game:

Additional Features

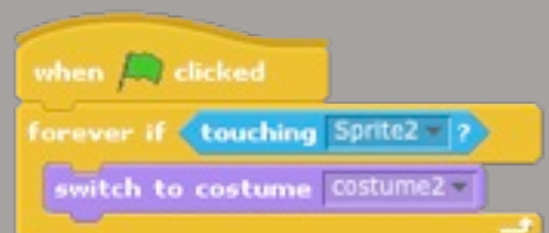
Adding Sound

Within the “Sound” category there is a range of commands available to you to add to your game. Controls that could supply your game with background music or even give your characters voices. These can also be imported from a range already available to you on file, but there is the option to record your own voice or sounds.



Costume change

The “Looks” category contains some handy commands you could use, especially if you want to do things like “collectables” for your game or even levels. By being able to change the costume you could design a new layer to “show” when a certain condition is met in your script. You could even simply “switch to the next costume”.





technocamps



www.technocamps.com



@Technocamps



Find us on
Facebook

