PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

An Roinn Oideachais agus Scileanna
Department of Education and Skills

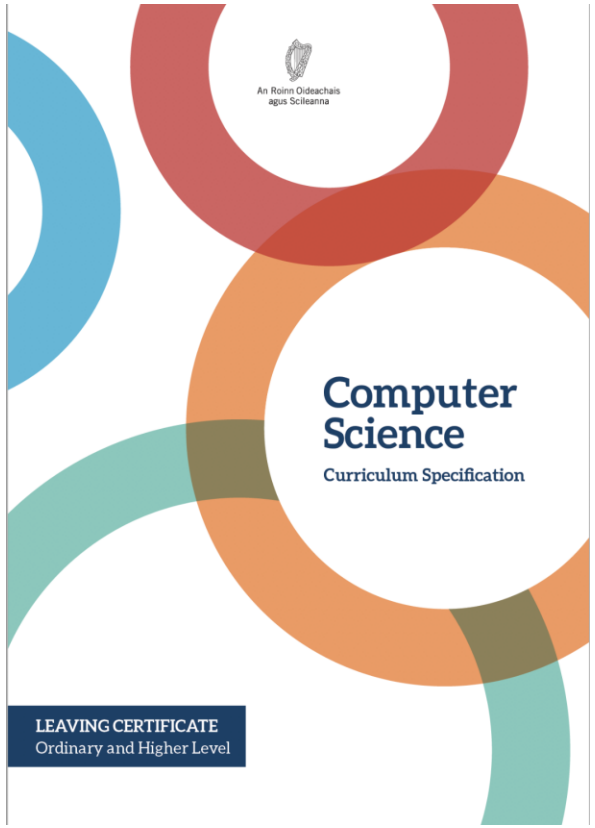
National Workshop 3

LEAVING CERTIFICATE
COMPUTER SCIENCE

PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Session 3

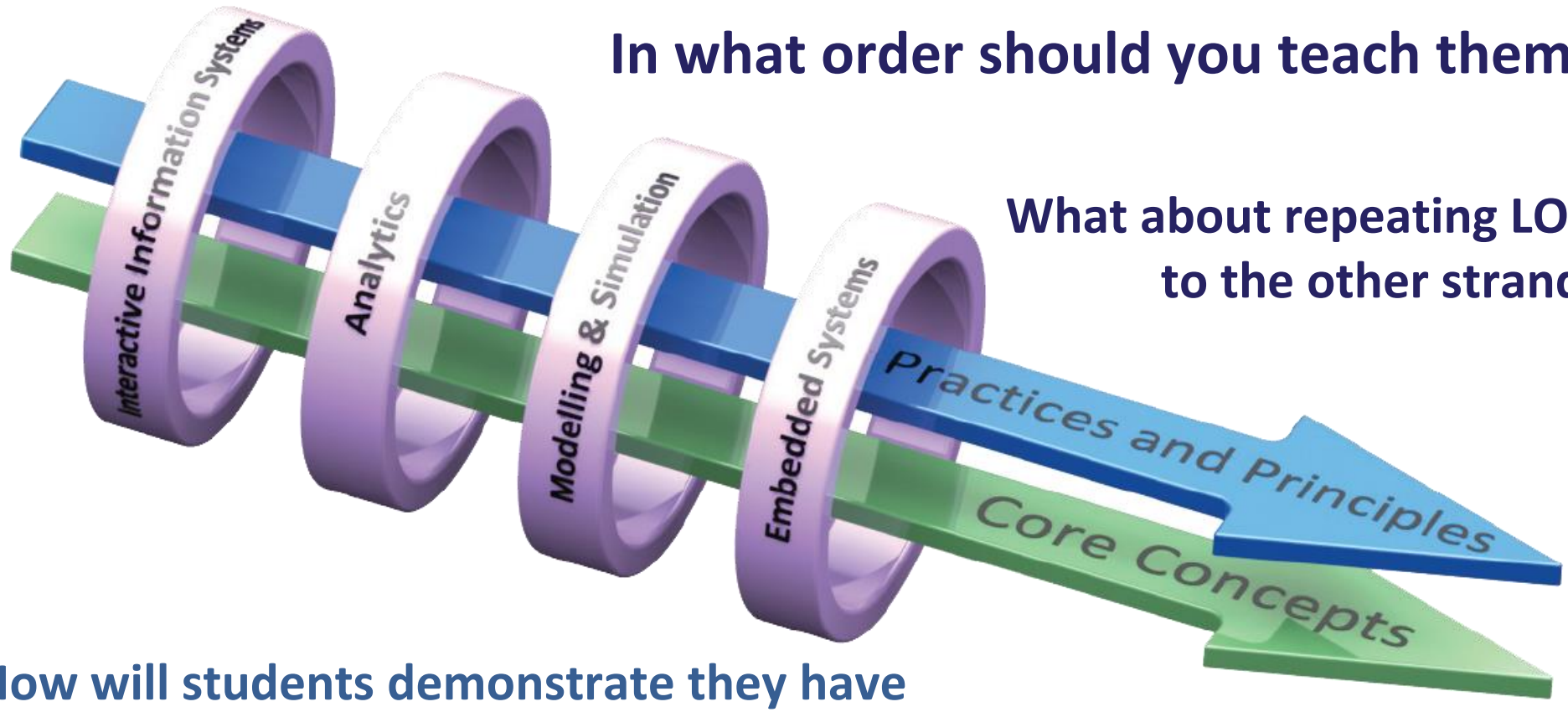
Curriculum planning



*'Learning outcomes can best be defined as statements of what a learner **knows**, **understands** and is **able to do** after completion of learning.'*

CEDEFOP (2009)

What will you do with LOs?



In what order should you teach them?

What about repeating LOs / linking to the other strands?

How will students demonstrate they have achieved the learning outcomes?

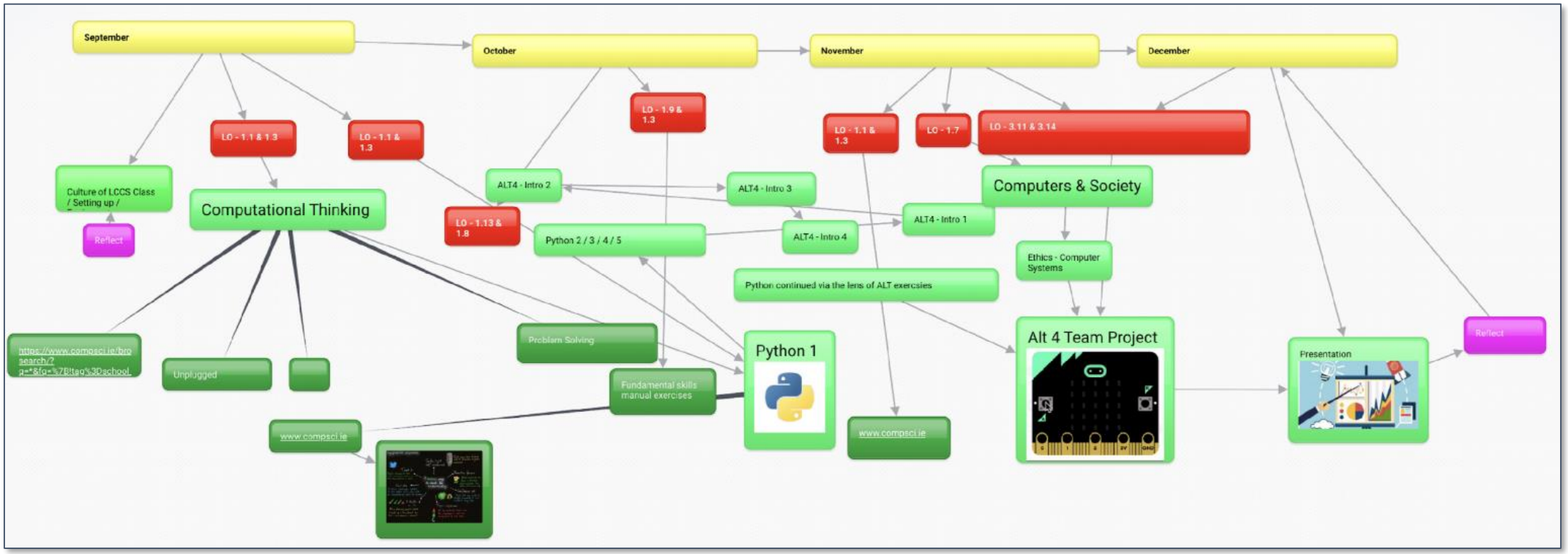
What content or resources do you need?

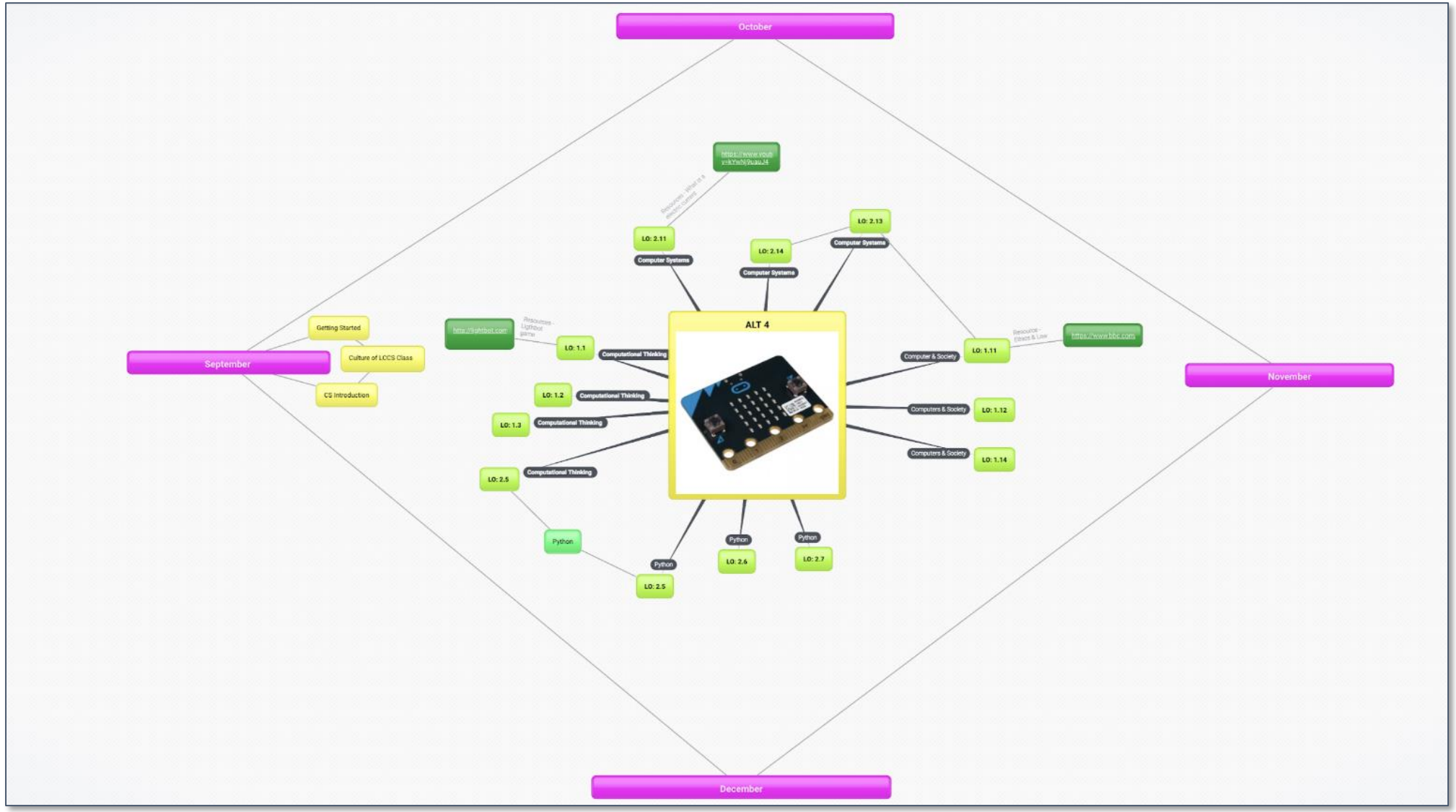
Key Message to remember:

Explore and teach the LOs through the lens of ALTs.

There are several ways to achieve this.

bubl.us


September
October
November
December
LO - 1.1 & 1.3
LO - 1.1 & 1.3
LO - 1.9 & 1.3
LO - 1.1 & 1.3
LO - 1.7
LO - 3.11 & 3.14
Culture of LCCS Class / Setting up /
Reflect
Computational Thinking
ALT4 - Intro 2
ALT4 - Intro 3
Computers & Society
LO - 1.13 & 1.8
Python 2 / 3 / 4 / 5
ALT4 - Intro 1
ALT4 - Intro 4
Ethics - Computer Systems
Python continued via the lens of ALT exercsies
https://www.compsci.ie/browse/?q=*&fq=%7Bltag%3Dschool_
Unplugged
Problem Solving
Python 1
Alt 4 Team Project
Presentation
Reflect
Fundamental skills manual exercises
www.compsci.ie
www.compsci.ie

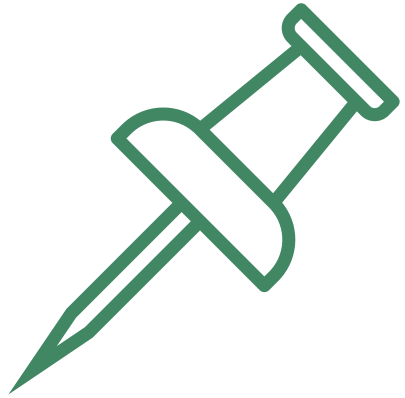

October
https://www.youtube.com/watch?v=kYw5jhvauJ4
Resources - what is a micro:bit?
LO: 2.11
Computer Systems
LO: 2.13
Computer Systems
LO: 2.14
Computer Systems
September
Getting Started
Culture of LCCS Class
CS Introduction
http://lightbot.com/
Resources - Lightbot game
LO: 1.1
Computational Thinking
ALT 4
LO: 1.2
Computational Thinking
LO: 1.3
Computational Thinking
LO: 2.5
Computational Thinking
Computer & Society
LO: 1.11
Resource - Ethics & Law
https://www.bbc.com/
Computers & Society
LO: 1.12
Computers & Society
LO: 1.14
November
Python
LO: 2.5
Python
LO: 2.6
Python
LO: 2.7
December

bubl.us

Group Activity

Develop a curriculum map for January to April

Focus on ALT2



Key Message to remember:

Explore and teach the LOs through the lens of ALTs.

There are numerous ways to achieve this.

Group Activity - Instructions

1. *Have a copy of the LCCS specification to hand.*
2. *In the Chat, click on the link to the Google doc.*
3. *In the Google doc, click on the link to the Bubbl.us diagram corresponding to your breakout room number.*
4. *Develop a detailed curriculum map for January to April – ALT2*
5. *Work in your group and consider…*

   Topics / LOs / Resources / Assessment / Build up to ALT2 / ALT2 / Equipment etc.
6. *Present back to the wider group.*

What will you do with the LOs for ALT2?

In what order should you teach them?

What about repeating LOs / linking to the other strands?

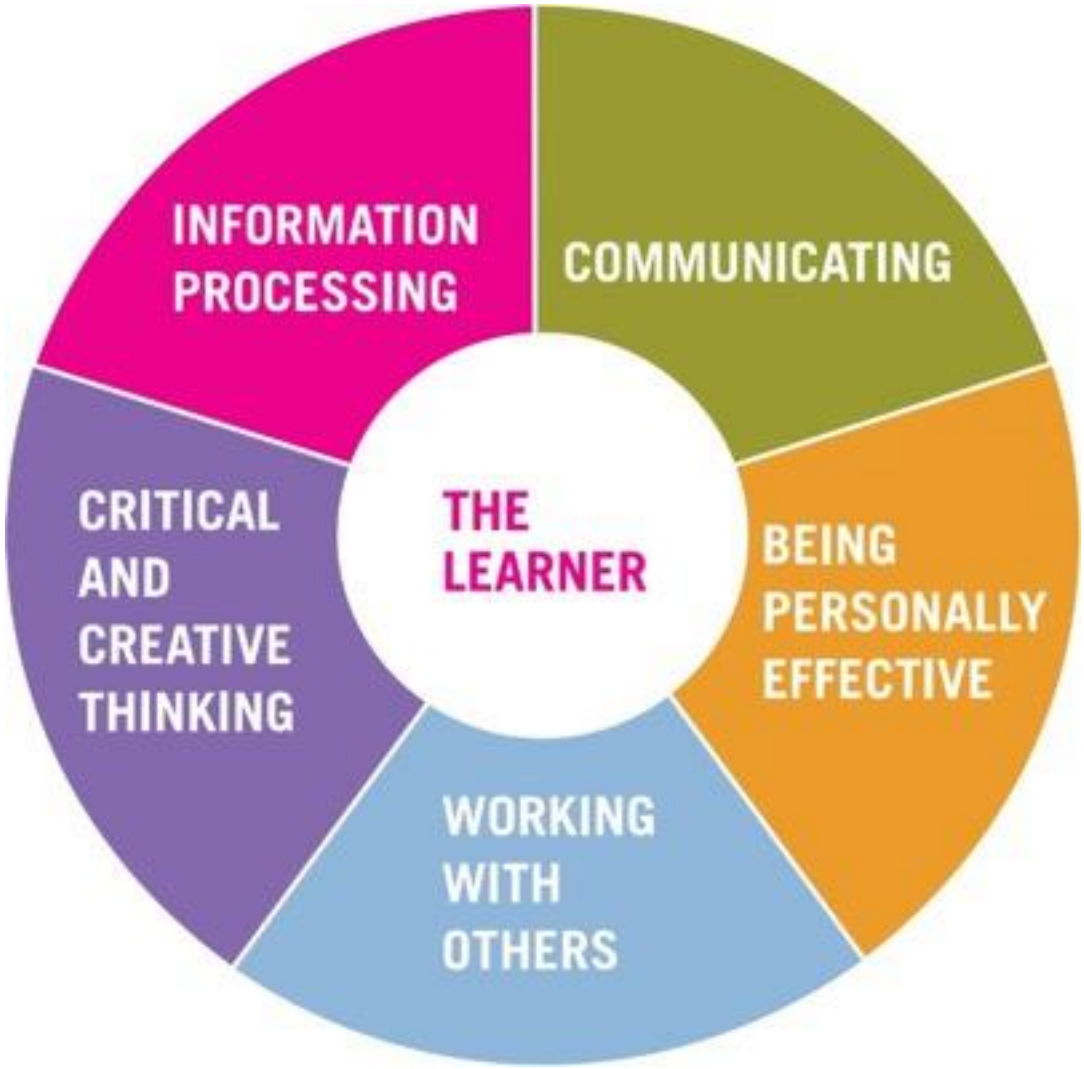
How will students demonstrate they have achieved the learning outcomes?

What content or resources will you need?

What can you include for Ordinary Level students?

Are there any considerations you should make for your students with SEN?

What about differentiation and extension of tasks?



Key Skills of Senior Cycle

LCCS Specification: p12

What LOs will your students experience?

Are there links to the other strands?

Why did you make these decisions?

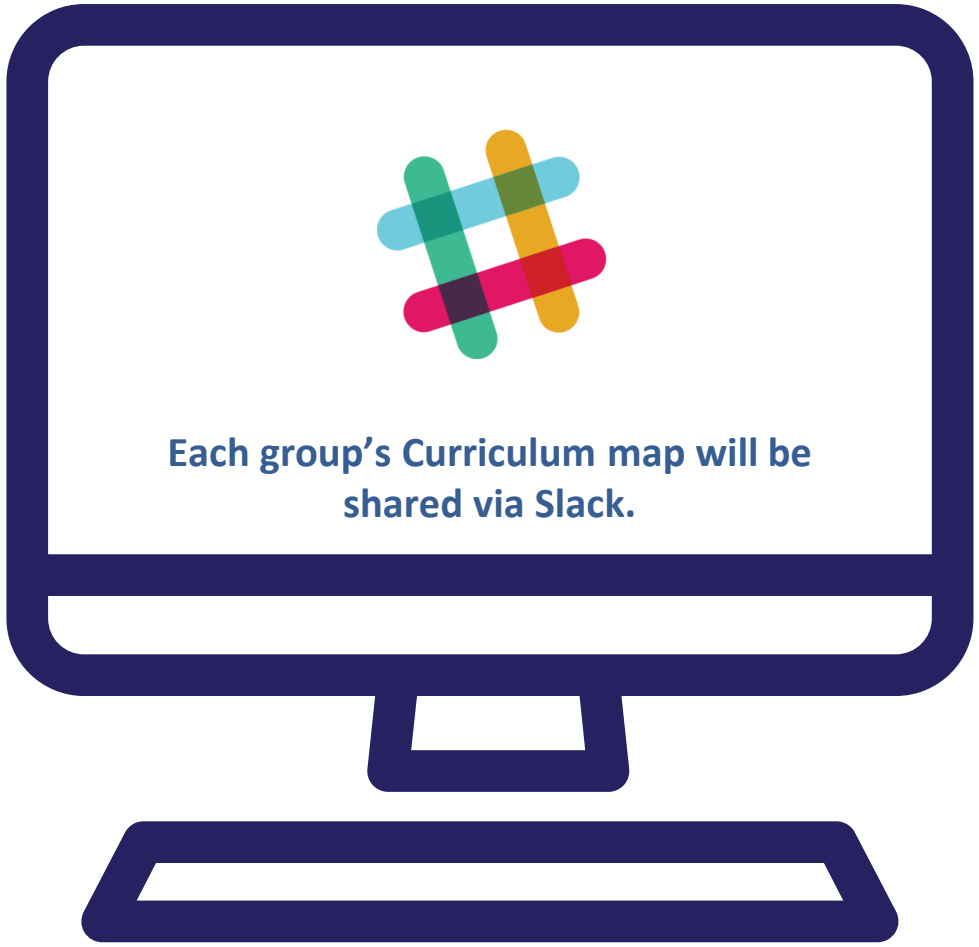
What learning experiences will help your students to achieve these LOs?

Where do you want to be in September 2022 in terms of the course?

What did you find challenging about this task?

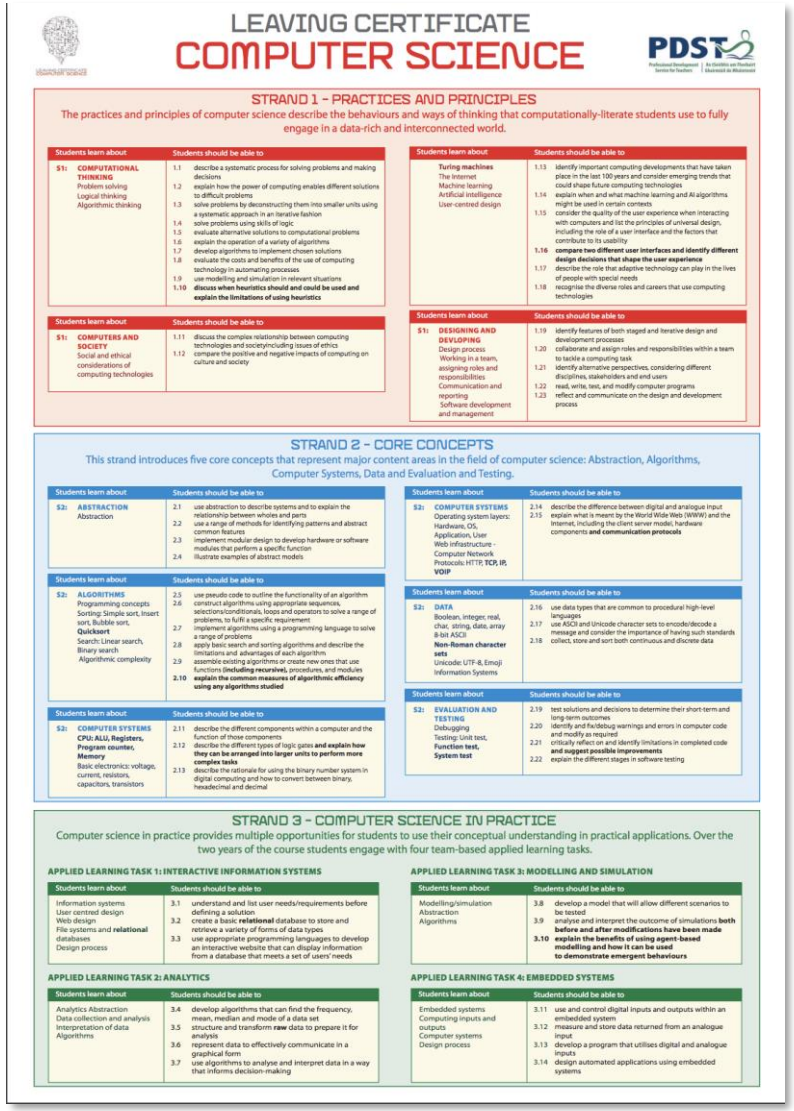
How was your thinking extended in relation to curriculum planning?

In what way will you teach the LOs through the lens of the ALTs?


Each group’s Curriculum map will be shared via Slack.

PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

LCCS Promotion in your school



LEAVING CERTIFICATE
COMPUTER SCIENCE

PDST

STRAND 1 – PRACTICES AND PRINCIPLES

The practices and principles of computer science describe the behaviours and ways of thinking that computationally-literate students use to fully engage in a data-rich and interconnected world.

Students learn about	Students should be able to
S1: COMPUTATIONAL THINKING Problem solving Logical thinking Algorithmic thinking	1.1 describe a systematic process for solving problems and making decisions 1.2 explain how the power of computing enables different solutions to difficult problems 1.3 solve problems by deconstructing them into smaller units using a systematic approach in an iterative fashion 1.4 solve problems using skills of logic 1.5 evaluate alternative solutions to computational problems 1.6 explain the operation of a variety of algorithms 1.7 develop algorithms to implement chosen solutions 1.8 evaluate the costs and benefits of the use of computing technology in automating processes 1.9 use modelling and simulation in relevant situations **1.10 discuss when heuristics should and could be used and explain the limitations of using heuristics**

Students learn about	Students should be able to
Turing machines The Internet Machine learning Artificial intelligence User-centred design	1.13 identify important computing developments that have taken place in the last 100 years and consider emerging trends that could shape future computing technologies 1.14 explain when and what machine learning and AI algorithms might be used in certain contexts 1.15 consider the quality of the user experience when interacting with computers and list the principles of universal design, including the role of a user interface and the factors that contribute to its usability **1.16 compare two different user interfaces and identify different design decisions that shape the user experience** 1.17 describe the role that adaptive technology can play in the lives of people with special needs 1.18 recognise the diverse roles and careers that use computing technologies

Students learn about	Students should be able to
S1: COMPUTERS AND SOCIETY Social and ethical considerations of computing technologies	1.11 discuss the complex relationship between computing technologies and society including issues of ethics 1.12 compare the positive and negative impacts of computing on culture and society

Students learn about	Students should be able to
S1: DESIGNING AND DEVLOPING Design process Working in a team, assigning roles and responsibilities Communication and reporting Software development and management	1.19 identify features of both staged and iterative design and development processes 1.20 collaborate and assign roles and responsibilities within a team to tackle a computing task 1.21 identify alternative perspectives, considering different disciplines, stakeholders and end users 1.22 read, write, test, and modify computer programs 1.23 reflect and communicate on the design and development process

STRAND 2 – CORE CONCEPTS

This strand introduces five core concepts that represent major content areas in the field of computer science: Abstraction, Algorithms, Computer Systems, Data and Evaluation and Testing.

Students learn about	Students should be able to
S2: ABSTRACTION Abstraction	2.1 use abstraction to describe systems and to explain the relationship between wholes and parts 2.2 use a range of methods for identifying patterns and abstract common features 2.3 implement modular design to develop hardware or software modules that perform a specific function 2.4 illustrate examples of abstract models

Students learn about	Students should be able to
S2: COMPUTER SYSTEMS Operating system layers: Hardware, OS, Application, User Web infrastructure - Computer Network Protocols: HTTP, **TCP, IP, VOIP**	2.14 describe the difference between digital and analogue input 2.15 explain what is meant by the World Wide Web (WWW) and the Internet, including the client-server model, hardware components **and communication protocols**

Students learn about	Students should be able to
S2: ALGORITHMS Programming concepts Sorting: Simple sort, Insert sort, Bubble sort, **Quicksort** Search: Linear search, Binary search Algorithmic complexity	2.5 use pseudo code to outline the functionality of an algorithm 2.6 construct algorithms using appropriate sequencies, selections/conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement 2.7 implement algorithms using a programming language to solve a range of problems 2.8 apply basic search and sorting algorithms and describe the limitations and advantages of each algorithm 2.9 assemble existing algorithms or create new ones that use functions (**including recursive**), procedures, and modules **2.10 explain the common measures of algorithmic efficiency using any algorithms studied**

Students learn about	Students should be able to
S2: DATA Boolean, integer, real, char, string, date, array 8-bit ASCII **Non-Roman character sets** Unicode: UTF-8, Emoji Information Systems	2.16 use data types that are common to procedural high-level languages 2.17 use ASCII and Unicode character sets to encode/decode a message and consider the importance of having such standards 2.18 collect, store and sort both continuous and discrete data

Students learn about	Students should be able to
S2: COMPUTER SYSTEMS **CPU: ALU, Registers, Program counter, Memory** Basic electronics: voltage, current, resistors, capacitors, transistors	2.11 describe the different components within a computer and the function of those components 2.12 describe the different types of logic gates **and explain how they can be arranged into larger units to perform more complex tasks** 2.13 describe the rationale for using the binary number system in digital computing and how to convert between binary, hexadecimal and decimal

Students learn about	Students should be able to
S2: EVALUATION AND TESTING Debugging Testing: Unit test, **Function test, System test**	2.19 test solutions and decisions to determine their short-term and long-term outcomes 2.20 identify and fix/debug warnings and errors in computer code and modify as required 2.21 critically reflect on and identify limitations in completed code **and suggest possible improvements** 2.22 explain the different stages in software testing

STRAND 3 – COMPUTER SCIENCE IN PRACTICE

Computer science in practice provides multiple opportunities for students to use their conceptual understanding in practical applications. Over the two years of the course students engage with four team-based applied learning tasks.

APPLIED LEARNING TASK 1: INTERACTIVE INFORMATION SYSTEMS

Students learn about	Students should be able to
Information systems User centred design Web design File systems and **relational** databases Design process	3.1 understand and list user needs/requirements before defining a solution 3.2 create a basic **relational** database to store and retrieve a variety of forms of data types 3.3 use appropriate programming languages to develop an interactive website that can display information from a database that meets a set of users' needs

APPLIED LEARNING TASK 2: ANALYTICS

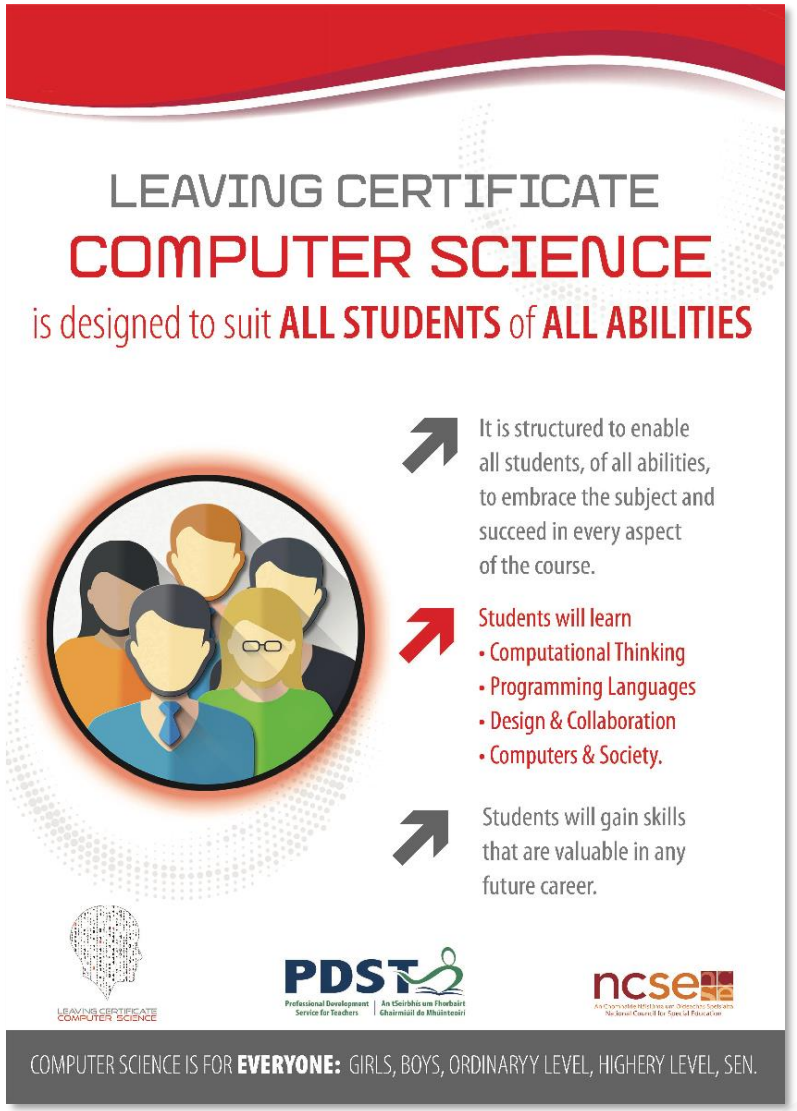
Students learn about	Students should be able to
Analytics Abstraction Data collection and analysis Interpretation of data Algorithms	3.4 develop algorithms that can find the frequency, mean, median and mode of a data set 3.5 structure and transform **raw** data to prepare it for analysis 3.6 represent data to effectively communicate in a graphical form 3.7 use algorithms to analyse and interpret data in a way that informs decision-making

APPLIED LEARNING TASK 3: MODELLING AND SIMULATION

Students learn about	Students should be able to
Modelling/simulation Abstraction Algorithms	3.8 develop a model that will allow different scenarios to be tested 3.9 analyse and interpret the outcome of simulations **both before and after modifications have been made** **3.10 explain the benefits of using agent-based modelling and how it can be used to demonstrate emergent behaviours**

APPLIED LEARNING TASK 4: EMBEDDED SYSTEMS

Students learn about	Students should be able to
Embedded systems Computing inputs and outputs Computer systems Design process	3.11 use and control digital inputs and outputs within an embedded system 3.12 measure and store data returned from an analogue input 3.13 develop a program that utilises digital and analogue inputs 3.14 design automated applications using embedded systems



PDST LCCS Posters – LOs & CS for Everybody

LCCS Information Videos

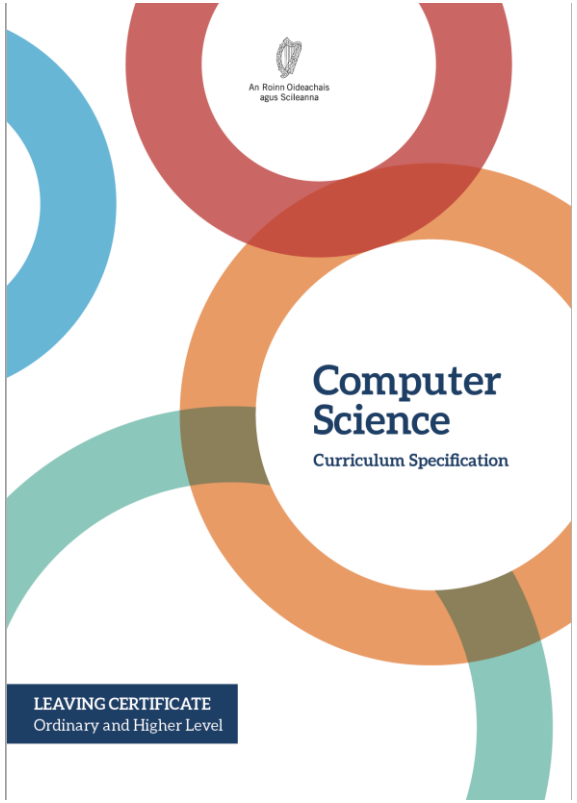
In September 2019, six phase 1 schools, in conjunction with NCCA, created videos to inform and advise schools starting out on Leaving Certificate Computer Science in the national rollout. The videos are intended for students, teachers and principals. They are also suitable for informing the school community about the subject in general and also about a particular main theme indicated below.

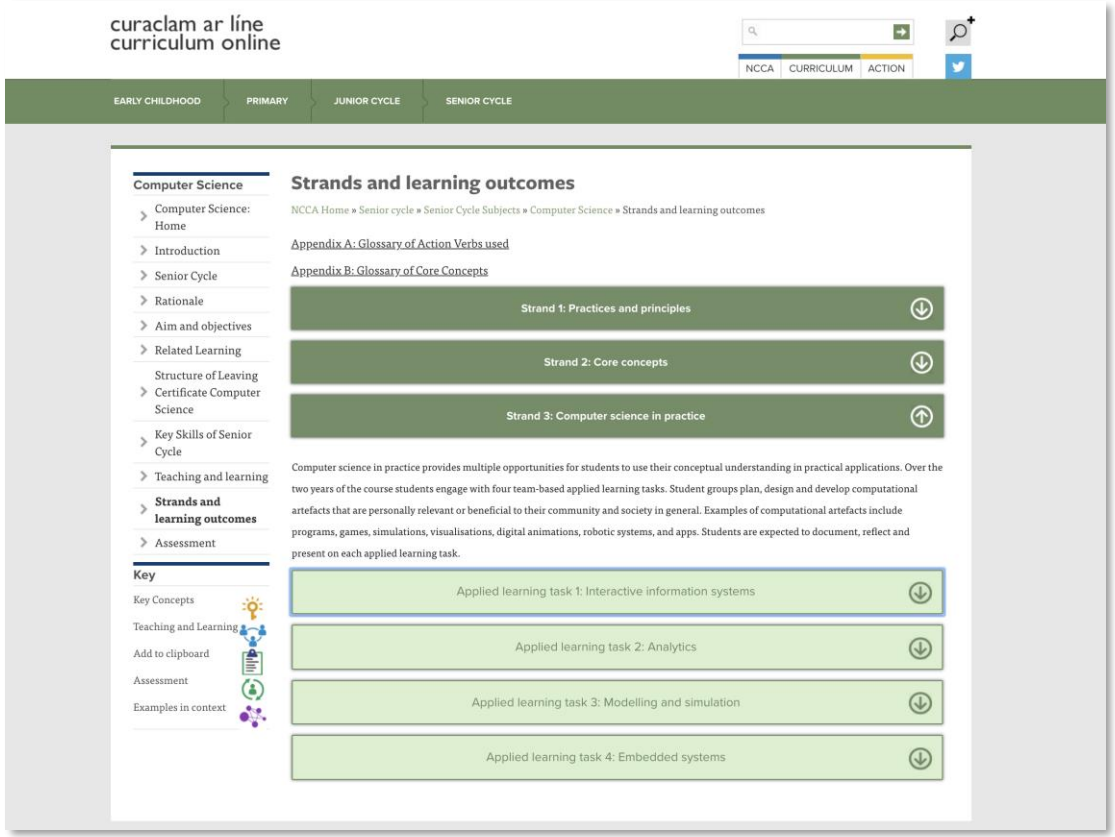
School	Sector	Main Theme
Abbey Vocational School, Donegal	ETB	Student Voice (Real world applications)
Coláiste Bríde, Dublin	Voluntary Secondary	How CS changes your thinking
Coláiste Na Ríochta, Kerry	ETB	Making CS work for your school
Mount Temple Comprehensive, Dublin	Community and Comprehensive	How CS changes your thinking
Presentation Secondary School, Tipperary	Voluntary Secondary	Computer Science - beyond coding
St. Eunan's College, Donegal	Voluntary Secondary	Classroom Collaboration

https://ncca.ie/en/senior-cycle/curriculum-developments/computer-science

https://www.ncca.ie/media/3261/lccs-information-brochure.pdf


LCCS Launch Video
02:38





https://www.curriculumonline.ie

Data Sets

kaggle

amazon

yelp

Census AtSchool

airbnb

unicef

worldometer

DATA.GOV.IE

An Phríomh-Oifig Staidrimh Central Statistics Office

INTERNATIONAL MONETARY FUND

Data Science Communities

IBM Data Science Community - https://community.ibm.com/community/user/datascience/home

Open Data Science - https://ods.ai/

Data Science Central - https://www.datasciencecentral.com/

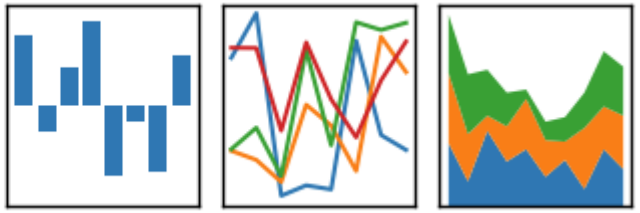
Driven Data - https://www.drivendata.org/


COMPSCI.IE
All of these resources can be found on Compsci.

Analytics Toolkit

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



plotly

NumPy

matplotlib

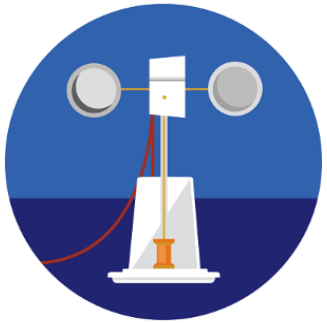
Version 3.0.2

Useful Tutorials

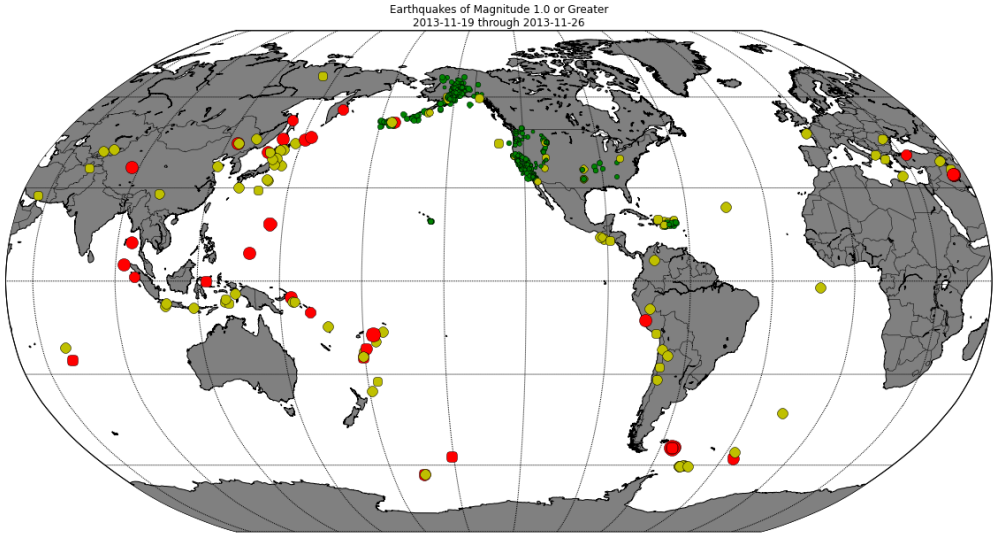
Comparing Speeds

Microsoft

Hacking STEM



Anemometer



http://introtopython.org/visualization_earthquakes.html

https://realpython.com/tutorials/data-science/

Pythonic Data Cleaning With NumPy and Pandas

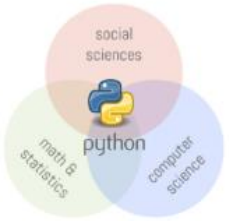
Mar 26, 2018 data-science intermediate

The Ultimate Guide To Speech Recognition With Python

Mar 21, 2018 advanced data-science machine-learning

Python Plotting With Matplotlib (Guide)

Feb 28, 2018 basics data-science



Python for Social Scientists

data-science python

Using Pandas to Read Large Excel Files in Python

data-science

Analyzing Obesity in England With Python

basics data-science

An Roinn Oideachais
Department of Education