**Introduction to cryptography**

**Lesson 3: Ciphers and text-based programming**

**Introduction**

In this lesson students introduced to text-based programming. They explore and analyse JavaScript code before using their developing knowledge of selection, variables, functions and text-based programming, alongside their own algorithms to create a Caesar cipher in Python (if you have micro:bits. If you do not, an alternative final activity is given).

**Time:** approx. 60 minutes

**Learning objectives**

* To develop practical understanding of text-based programming
* To understand and use selection, variables and functions
* To analyse JavaScript code
* To program a Caesar cipher with Python (if you have micro:bits)

**Materials needed:** Lesson plan, lesson guide, Caesar cipher algorithms from previous lesson, Caesar cipher example files, [micro:bit Python editor](https://python.microbit.org/) (optional), micro:bits (optional), USB cables, battery cables.

**Lesson summary**

1. Recapping algorithms (5 minutes)
2. Introducing text programming (5 minutes)
3. JavaScript Caesar cipher (10 minutes)
4. Python Caesar cipher (30 minutes)
5. Review & wrap up (10 minutes)

**Introduction: Recapping algorithms (5 minutes)**

* Use **slide 2** to recap students’ learning from last lesson and introduce the learning objectives on **slide 3** if you wish.

**Introducing text programming (10 minutes)**

* Show **slide 4** and ask students what they can see in the image (a very simple micro:bit program showing the code in both graphical (MakeCode blocks editor) and text (JavaScript) form.
* Invite students to think/pair/share the question on **slide 5** about their current level of experience/understanding of text based programming.
* If this is their first introduction, give them 5 minutes to complete the task on the slide in pairs to experiment with going between the blocks and JavaScript editors on MakeCode.

**JavaScript Caesar cipher (10 minutes)**

* Visit the [web link](https://codepen.io/MattCK/pen/OyQyxE) given on **slide 6** and ask students what they think it is (a JavaScript Caesar cipher).
* Invite students to use the cipher and lead a discussion around the purpose of the 3 different columns.
* Explain you would like students to experiment with it in pairs, trying to ‘decipher the cipher’ and work out what each part of the JavaScript code is doing (they can ignore the other 2 columns, unless you want them to explore CSS and HTML too).
* Highlight that it is very unlikely they will be able to work it all out, but they should recognise some familiar elements from previous programming and/or their algorithms and explain that all programmers have to work with different languages, so it’s a good skill to learn.
* This is tricky, however give students 5 minutes to work in paris, then lead a discussion around what they have found out, highlighting the use of selection, variables, functions and anything else of interest (**slides 7 to 9** may help with the task and/or discussion).

**Python Caesar ciphers (30 minutes):** If you **do have** physical micro:bits

* Explain that they are now going to use their algorithms and their knowledge from looking at the JavaScript cipher to program their own Caesar cipher to decipher messages using Python.
* Show **slide 10** and ask students to think/pair/share what they can see in the code.
  + Can they spot the variables and functions?
  + Can they talk through the code?
  + How does it differ from the JavaScript and their algorithm in how it creates the cipher?

Lead a brief discussion to support understanding, highlighting the purpose of the different parts of the code:

* + The function that gives each letter of the alphabet a number, so it can shift it along a set amount to a different number
  + The function that goes through each letter of the encoded work and applies the shift.

Show slide 11 and do the same as above, ensuring all students have at least a basic grasp of the different parts of the code:

* + Checking if button A is pressed
  + The value of the shift key
  + The amount the cipher has shifted
  + How the display is cleared and the message displayed.
* Get students to work in pairs to visit MicroPython and code and test their own Caesar cipher. Depending on their confidence, you may wish to give them some/all of the code to start off with, perhaps printing it off so they at least have to type it in (and learn that they need to be very exact). Alternatively, the supplied .HEX files can be dragged and dropped into the online Python editor <https://python.microbit.org> or they can be opened in **micro:bit classroom**.
* Depending on confidence, you could use the simplified Python Caesar program (**slide 12**) which does not use functions but which produces similar results.
* Encourage them to make comments to explain their code as they go (and print it off if you wish to assess it).

**Python Caesar ciphers (30 minutes):** If you **don’t have** physical micro:bits

* Print out the code of the hex file and cut it up into sections.
* Give small groups the code sections, A3 paper and glue.
* Ask students to work in small groups to consider what each section does and sequence it in the right order to make a working program, annotating on their A3 paper notes to say what they think each section does (use the prompts in the Python section above to help)
* Encourage groups to ‘step-through’ the program to test and debug it before pairing up with another team to send and receive messages with their ‘program’.
* Discuss their learning as a class.

**Review and wrap up (10 minutes)**

* Review the learning objectives on **slide 13** if you wish and use the questions on slide 14 to review students’ learning in this lesson and Cryptography as a whole.

**Differentiation**

**Support:**

* Students may benefit from being given specific sections of the JavaScript / Python code to look at, selected to suit their areas of confidence. Then they can be given additional sections of code to develop their understanding.
* If students are unable to access the text-based programming of the Caesar cipher, they can stay in the Micro:bit editor, looking between the blocks and JavaScript editor to build up their understanding slowly with help (e.g using a simple tutorial such as the flashing heart and looking at how it is represented in both editors).

**Stretch & challenge:**

* Students can be challenged to write more of their own code, adapting or totally changing the Python code to create their own Caesar cipher. Explaining to the class how they did so could help to develop others.
* Students could join the Cyber challenge <https://go.joincyberdiscovery.com/>

**Opportunities for assessment:**

* Informal assessment of students’ responses during the activities.
* More formal assessment of students’ ciphers and responses to the review (you could ask them to give their responses to the questions individually on paper/on a computer if you preferred).