

# ICT – Year 8 – API Medium Term Plan/SOW

**The Academy of St Francis of Assisi**

	<b>Title : Programming in Python</b>		Number of lessons in sequence	<b>11 (including revision, assessment &amp; improvements)</b>
<b>UNIT 1</b>				
<p><b>Overarching Curricular Goals (Aims)</b> <small>(What do you intend students know about and be able to do by the end of the topic, or scheme of learning. Critical knowledge needed to inform later learning and wider contexts.)</small></p>	<p><b>By the end of this unit students will:</b> Pupils will revisit and recap their knowledge from the Year 7 unit of work which was based on a simple text- based programming language and extend their understanding of the three main programming concepts; Sequence, Selection &amp; Iteration using a second text-based programming language, Python. Pupils will be able to identify the similarities and differences the two programming languages have and be able to start writing their own code using Python. They will learn how to embed the programming concepts into Python code and create a program based of a set of scenarios. Throughout the unit, pupils will be to predict the behaviour of the Python code and how to ‘debug’ their program if it doesn’t work based on a set of rules. They will learn how keywords such as variable, data type, if statement, while and for loops are implemented into Python code and how they link with the main programming concepts.</p>		<p><b>Links to National Curriculum</b></p> <p><b>Links to &amp; building upon prior learning Including KS2 if Yr7</b></p>	<p>Use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems</p> <p>Pupils completed a unit of work in Year 7 which focussed on programming however this was with a different, simpler programming language which only had 15 key words to use. The same concepts will be revisited by this time, using a programming language that is used in industry, Python. This is still a text based programming language but gives pupil’s the opportunity to be more independent without the use of predictive commands like in Small Basic.</p> <p><b>KS2 Link</b> - Use sequence, selection &amp; iteration/ work with variables/ use logical reason to solve simple algorithms</p>
<p>Outcomes/ Success Criteria</p>	<p>Knowledge Learners will:</p> <ul style="list-style-type: none"> <li>• Be able to identify key ‘code ingredients’ such as variables, data types, IF statements and loops within Python programs</li> <li>• Be able to compare two different programming languages and describe the similarities and differences</li> <li>• Be able to describe different types of programming errors and be able to explain why the errors occurred</li> </ul> <p>Skills: Learners will:</p> <ul style="list-style-type: none"> <li>• Be able to embed the three main programming concepts into Python programming</li> <li>• Be able to predict the behaviour of Python code</li> <li>• Be able to identify different types of errors in their code and attempt to debug</li> </ul>			
<b>2/3 tier vocabulary.</b>	<b>Differentiation/Scaffolding/Support.</b>	<b>Stretch and challenge opportunities in class, enrichment and home learning.</b>	<b>Opportunities for wider reading/Listening/watching.</b>	
<p>See Knowledge Organiser</p> <p><b>Oracy:</b></p> <p>Also Add hyperlink to KO</p>	<p><b>Knowledge Support:</b> Using the knowledge organiser to enable pupils to be able to link programming knowledge to the skills they are learning. Pupils to be encouraged to use KO when choosing the appropriate programming concept or data type when coding their own program.</p> <p><b>Key Concept Support</b> – visualisation of key terms that are frequently used in programming. E.g. Variables – including activities where pupils have the opportunity to demonstrate their knowledge through the use of annotated diagrams.</p> <p><b>Reading support:</b> Encouraging pupils to become confident reading a chunk of code and being able to explain what will happen when it is run.</p> <p><b>Skills support:</b> Flash cards with adaptive programming code for pupils to use if they are struggling with the programming aspect of the unit. Include teacher demonstrations within lessons to model how to program to certain types of scenarios.</p> <p>The most common areas of misconceptions during this unit is being able to recognise the basic programming constructs. Pupils will be given keyword tasks at the start of the unit which will be revisited throughout the unit in lessons. Once it is clear they understand the definitions of these, they will be shown how they used them in the programming unit last year and how to do it in a different programming language. This</p>		<p>More able pupils will be able to access more advanced programming skills such as nested IF statements and while loops. They will be able to create a program based on a more complex scenarios and amalgamate the different concepts into one program.</p> <p>Less able pupils will be having sentence starter style programming starters with suggested.</p> <p><b>Scholarship:</b></p> <p><b>Why is Python a good programming language for kids?</b> <a href="https://codakid.com/5-reasons-python-programming-is-perfect-for-kids/">https://codakid.com/5-reasons-python-programming-is-perfect-for-kids/</a></p> <p><b>Careers – how Python can help children prepare for jobs other than computing.</b> <a href="https://www.codecamp.com.au/blog/five-careers-where-coding-skills-will-help-your-kids-get-ahead">https://www.codecamp.com.au/blog/five-careers-where-coding-skills-will-help-your-kids-get-ahead</a></p> <p><b>Python Programming for the Absolute Beginner</b> By Michael Dawson</p> <p><b>Hello World!</b> Programming Magazine subscription</p>	<ul style="list-style-type: none"> <li>- <b>Imitation Game</b> – Alan Turing breaking the WW2 codes using Enigma – links History of Computing unit from last year</li> <li>- <b>Python Programming for the Absolute Beginner</b> By Michael Dawson</li> <li>- <b>Podcast: Fetch Decode Explain</b></li> <li>- <b>Why is Python a good programming language for kids?</b> <a href="https://codakid.com/5-reasons-python-programming-is-perfect-for-kids/">https://codakid.com/5-reasons-python-programming-is-perfect-for-kids/</a></li> <li>- <b>Careers – how Python can help children prepare for jobs other than computing.</b> <a href="https://www.codecamp.com.au/blog/five-careers-where-coding-skills-will-help-your-kids-get-ahead">https://www.codecamp.com.au/blog/five-careers-where-coding-skills-will-help-your-kids-get-ahead</a></li> <li>- <b>Python Programming for the Absolute Beginner</b> By Michael Dawson</li> <li>- <b>Hello World!</b> Programming Magazine subscription</li> </ul>

way, they are able to link the knowledge of the concept to the skills of using the programming language.

**Stretch and Challenge** – Some pupils are going to understand Python programming more than others and will code solutions to problems quickly. These pupils will be given GCSE level programs to try and solve in Python code. If this seems to high level for the pupils, jumbled up code will be given as support. The pupil's task will then be to unjumble it and write it in Python in the correct order. Pupils will then be asked to comment the code using key terminology and identifying key concepts

Unit Title	Sequence of learning Lesson title, theme, big question.	Key concepts/outcomes/knowledge and skills. (Variables, Selection, Sequence, Iteration, Data types, Pseudo Code, Flowcharts, )	Assessment/ including specific content/ knowledge/skills tested. Green=assess/Blue=improve	HWK. Add Hyperlink To be in books clearly marked	Furthering Cultural Capital. & Opportunities for reading	Recall of prior or future topics –	Lesson resources including or hyperlink to supporting websites/resources/books/texts & individual lessons. 5xT+L essentials to be included in individual lessons,
1	Recap of keywords and basic programming knowledge	<p><b>Key Concepts/Knowledge/Skills</b></p> <ul style="list-style-type: none"> <li>- Pupils to recap the key programming concepts of sequence, selection and iteration; understanding what they mean and how they fit into the programming language they used in Year 7</li> <li>- Pupils to recap how variables fit into programming. They should be able to identify where they are in a piece of code and what they would store within a program</li> <li>- At this stage, pupils won't be writing any of their own code but being able to recap, review and analyse</li> </ul> <p><b>Outcomes</b></p> <ul style="list-style-type: none"> <li>- Pupils will be able to identify where the main programming concepts have been used within a piece of code</li> <li>- Pupils will be able to identify and describe where variables are used in a piece of code and what they will store</li> </ul>		Learn keywords from KO	<p>Research – what sort of industries still use Python programming?</p> <p>Twitter Article – How Python is used in industry.</p>	Recall of keywords from Year 7 Programming SOW specifically Sequence, Selection & Iteration	<p><b>Teacher PowerPoint</b></p> <p><b>Student Resource</b></p>
2	Introduction to Python/ Compare two different programming languages	<p><b>Key Concepts/Knowledge/Skills</b></p> <ul style="list-style-type: none"> <li>- Pupils to learn basic Python programming and understanding how to get the program to print simple statements, how to create variables and store data in variables and start to look at how to debug a program if there are errors identified</li> <li>- Pupils will be able to understand the syntax rules Python follows so their code can run successfully</li> </ul>	<p>Knowledge quiz based on keywords and definitions of key terminology pupils will need to know for the unit of work.</p> <p>Pupils will improve their work based on feedback and close any gaps in their knowledge.</p>			Recall of Microsoft Small Basic programming language pupils learnt last year and compare to Python code. Pupils are to	<p><b>Teacher PowerPoint</b></p> <p><b>Student Resource</b></p>

		<ul style="list-style-type: none"> <li>- Pupils to compare Microsoft Small Basic code to Python code and describe/explain how they have similarities and differences</li> <li>- Pupils will be able to draw an annotated diagram to show the process of how variables store values in a program and how they can be recalled</li> </ul> <p><b>Outcomes</b></p> <ul style="list-style-type: none"> <li>- Pupils will be able to start writing their own code in Python using statements such as print and understanding the syntax that Python follows</li> <li>- Pupils will be able to explain the similarities and differences in two different programming languages</li> </ul>				compare both programming languages to identify what is similar and different about the two examples they are shown.	
3	Predicting the behaviour of code	<p><b>Key Concepts/Knowledge/Skills</b></p> <ul style="list-style-type: none"> <li>- Pupils to be able to analyse a piece of Python code and predict what will happen when it is run on the computer</li> <li>- Pupils to be able to code the Python program and see how their prediction and end result is either similar or different</li> <li>- Pupils to continue to follow Python syntax rules and debug their code when errors are identified</li> </ul> <p><b>Outcomes</b></p> <ul style="list-style-type: none"> <li>- Pupils will be able to start identifying different programming concepts whilst predicting the behaviour of the piece of code</li> <li>- Pupils will be able to start linking key words to concepts and how they look in a second programming language and how this is interpreted in Python code</li> </ul>	<p><b>Self-Assessment</b> – pupils will assess their own work to see if they have correctly predicted how a piece of code will behave when it is run. <a href="#">Pupils will improve their work based on answers given in class.</a></p>	Error spotting/debugging within Python code	Clips from Imitation Game which gives pupils an insight into Alan Turing breaking the codes in WW2	A similar task was used in Microsoft Small Basic unit but only as a starter so pupils are used to the structure of the activity	<p><b>Teacher PowerPoint</b></p> <p><b>Student Resource</b></p>
4 & 5	Sequence and Selection in Programming	<p><b>Key Concepts/Knowledge/Skills</b></p> <ul style="list-style-type: none"> <li>- Pupils to be able to recall their knowledge of sequence and selection and how they are used in programming.</li> <li>- Pupils will be able to embed their knowledge of sequence and selection and link it to their programming knowledge</li> <li>- Pupils will be able to implement sequence and selection into their Python program to make their code more dynamic and they should be able to narrate how the program functions</li> </ul> <p><b>Outcomes</b></p> <ul style="list-style-type: none"> <li>- Pupils will be able to create their own program based on a set of scenarios that will include sequence and selection.</li> <li>- Pupils will be able to independently be able to debug their program based on syntax rules</li> </ul>	<p>Progress Check – can pupils identify where sequence and selection has been used in a program? Class activity</p> <p>Homework Task – Flowcharts</p>			Pupils covered sequence and selection and learnt how to embed the concepts into Microsoft Small Basic programming last year	<p><b>Teacher PowerPoint/Python Code</b></p> <p><b>Student Resource</b></p>
6 & 7	Iteration in Programming	<p><b>Key Concepts/Knowledge/Skills</b></p> <ul style="list-style-type: none"> <li>- Pupils to be able to recall their knowledge of iteration and how it is used in programming.</li> <li>- Pupils will be able to embed their knowledge of iteration focussing on two different types of loops and link it to their programming knowledge</li> <li>- Pupils will be able to identify and explain the difference between FOR and WHILE loops</li> </ul>	Progress Check – can pupils identify where iteration has been used in a program? Can they tell the difference between the two types of loop? Class activity	Identifying programming concepts in Python Code	Python programmer in industry – Guest Speaker	Pupils covered what iteration means last year as part of Year 7 unit but didn't program how to do it so they will be learning a new skill which builds	<p><b>Teacher PowerPoint/Python Code</b></p> <p><b>Student Resource</b></p>

		<ul style="list-style-type: none"> <li>- Pupils will be able to implement iteration through the use of FOR and WHILE loops into their Python program to make their code more dynamic and they should be able to narrate how the program functions</li> </ul> <p><b>Outcomes</b></p> <ul style="list-style-type: none"> <li>- Pupils will be able to create their own program based on a set of scenarios that will include sequence and selection.</li> <li>- Pupils will be able to independently be able to debug their program based on syntax rules</li> </ul>				on prior knowledge of the programming concept	
7	Consolidation of Programming Skills – Independent Programming	<p><b>Key Concepts/Knowledge/Skills</b></p> <ul style="list-style-type: none"> <li>- Pupils will consolidate their knowledge of sequence, selection and iteration and have to use their own judgement to decide which programming concept will need to be used based on a given scenario</li> <li>- Pupils will be able to independently be able to comment their code and give detailed annotations explaining how their code will execute</li> <li>- Pupils will be able to debug their program independently</li> </ul> <p><b>Outcomes</b></p> <ul style="list-style-type: none"> <li>- Pupils will produce a program that incorporates appropriately used programming concepts</li> <li>- Pupils will be able to narrate how their program will execute</li> <li>- Pupils will be able to describe and justify any bugs that are identified and explain how they will rectify the error</li> </ul>	<p>Peer Assessment – Error Spotting/Debugging Homework</p> <p>Independent Python programming project combining the skills learnt over lessons into one project which is teacher assessed</p>				<p><b>Teacher PowerPoint</b></p> <p><b>Student Resource</b></p>