# Introduction to Python

## What is Python?

- Python is a powerful programming language, which is very friendly for beginners.
- It is widely used in science and education, and by global organisations including NASA, Google and Walt Disney.
- It is written in text, making it fast to write short programs.
- Python was created by Guideo van Rossum in 1989, and he named it after the Monty Python's Flying Circus, a television comedy show in the 1970s.
- Rapid Router is written in Python 3, which is the latest version of the language.

## Why are we using Python in Rapid Router?

- Python is easy to use and easy to debug. Beginners can write clear and simple lines of code fairly quickly.
- It is also the programming language of choice in many secondary schools, and a large number of resources are being created for teaching and learning this particular language.

## What concepts will children learn when programming using Python in Rapid Router?

They will learn to be accurate and consistent in writing code. Clean syntax is critical as a basis for programming in all languages.

## Sequencing

They will direct the van using movement instructions, e.g.

```
v.move_forwards()
```

## Repetition

They will use loops to repeat the same code.

- The **for** statement, which repeats for the number of times you specify, e.g.

```
for num in range(4):
```

Where num is a name you have chosen, and 4 is the number of times you want the loop repeated

```
    v.turn_left()

    v.turn_right()
```

These three indented lines of code (indented 4 spaces) Will be executed each time the loop is passed through.

```
    v.move_forwards()
```

These lines of code tell the Rapid router van which way to move.

- the **while** statement, which runs a block of code as long as its condition is true, and stops when it becomes false, e.g.

```
while not v.at_destination():
```

While the van is not at its destination run the following indented code

```
    v.move_forwards()
```

Move the van forwards one space (the computer cycles through this code until the first line is false, ie the van has reached its destination)

## Selection

They will use the **if** statement, which conditionally executes a block of code, along with **else** and **elif** (a contraction of **else-if**); e.g.

```
if v.is_road(‘FORWARD'):
```

If the road exists forward run the indented code after the colon

```
    v.move_forwards()
```

Move the van forwards one space

```
elif v.is_road(‘LEFT'):
```

Otherwise if the road goes left, run the indented code after the colon

```
    v.turn_left()
```

Move the van to the left one space

```
else:
```

Otherwise if none of the above conditions apply, run the indented code after the colon

```
    v.turn_right()
```

Move the van to the right one space

## Defining new procedures or functions (which is another word for procedure).

They will use the **def** statement, which defines a **procedure**, e.g.

```
def bend():
```

Tells Python that you are creating a new procedure called bend which consists of the indented code after the colon

```
    v.turn_right()
```

This code will be the code the computer runs, when the procedure bend is called

```
    v.turn_left()
```

- Then they will **call** their new **procedure** or **function** in a program, among other lines of code:

```
bend()
```

Tells Python to run the procedure bend, which will turn the van right then left

```
v.move_forwards()
```

Move the van forwards

```
for count in range(2):
```

Repeat 2 times what comes after the colon (in this case bend)

```
    bend()
```

Run the procedure bend

```
v.move_forwards()
```

Move the van forwards (note this is outside the repeat loop)

```
bend()
```

## Creating variables

Children will learn to create a **variable** by giving it a name and assigning a value to this name. A **variable** is a slot in the computer's memory where you can store a numeric or text-based value. You can use almost any name except those that Python uses already. See the end of the Blockly-Python Phrasebook for these reserved words.

E.g.

```
number = 5
```

This tells the computer that you are naming a new variable number and giving it an initial value of 5.

## Using new functions which are already part of Python, e.g.

- The **print()** function
- This function prints anything that is inside the brackets. Those things must be separated by commas and any bits of text (strings) need to be inside inverted commas, e.g.

**print('Hello your groceries have been delivered!')**

## Incrementing variables

Some children will progress as far as changing the value of a **variable** within a program to address a particular task, e.g.

| | |
|---|---|
| `number = 1` | Defines a variable called number and gives it a starter value of 1 |
| `while number < 11:` | The colon tells the computer to run all the indented code |
| `    print(number)` | Says print the current value of the variable number |
| `    number = number + 1` | Tells the computer to increment the value of the variable number by 1 |

(Note: professional programmers would use **number+=1** to increment a **variable**, but we are using the above to make it clearer to the children what is happening. It will also work clearly with the other mathematical operators: - , * and /).

## Debugging syntax errors

- When you are writing up your code, it is common to make simple mistakes in the way you use colons, brackets, inverted commas and in the spelling of Python commands. The computer can't cope with this and will tell you it's stuck by producing an error message. This message will tell you in which line of code the mistake is located making it easy to check and debug.
- ParseError:bad input on line 5

## Debugging your program solution

You will also write some programs where the code is error free, but the program is not doing what you intend. You will then have to debug your algorithms and the code.

## What next?

- After completing the Upper Key Stage 2 levels in Rapid Router, children should have an initial understanding of programming/writing code in Python.

- Computing at School has a great number of Python resources http://www.computingatschool.org.uk/

- Phil Bagge's Python resources are aimed at Key Stage 2 - http://www.pythoncode.co.uk/

- The children may wish to explore further and learn more coding using an online course such as:

  Codeacademy http://www.codecademy.com/

- Or books such as:

  Python Basics, level 1 and 2 – Chris Roffey, Cambridge University Press,
  ISBN 978-1-107-65855-4