

Knowledge Organiser

Unit 5 – Python Programming



Summary

Programming is writing computer code to create a program, in order to solve a problem. Programs consist of a series of instructions to tell a computer exactly what to do and how to do it. An algorithm is a set of instructions that describes how to get something done. It is crucial that the steps in an algorithm are sequenced and performed in the right order - otherwise the algorithm will not work correctly.

Python is a general purpose and high-level programming language. You can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high-level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks

Key Words

Shell	This is the window where you type in the commands
Loop	A concept called loop, which helps in executing one or more statements up to a desired number of times.
Function	A function is a block of organized, reusable code that is used to perform a single, related action.
Variable	Variables are used to store information to be referenced and manipulated in a computer program. They also provide a way of labelling data with a descriptive name, so our programs can be understood more clearly by the reader and ourselves. It is helpful to think of variables as containers that hold information.

Key Words continued

Import turtle	<code>import turtle</code>
<code>wn=turtle.Screen</code>	<code>wn=turtle.Screen()</code>
<code>Bob=Turtle()</code>	<code>Bob=turtle()</code>
<code>wn.bgcolour("lightgray")</code>	<code>wn.bgcolor("lightgray")</code>
<code>Bob.color(red)</code>	<code>Bob.color("red")</code>
<code>bob.pensize(2)</code>	<code>Bob.pensize(2)</code>
<code>Bobforward(200)</code>	<code>Bob.forward(200)</code>
<code>Bob.left(90)</code>	<code>Bob.left(90)</code>
<code>Bob.forwards(200)</code>	<code>Bob.forwards(200)</code>

Improve the code

<pre> Import turtle wn=turtle.Screen() turtle.forward(200) turtle.right(90) turtle.forward(200) turtle.right(90) turtle.forward(200) turtle.right(90) turtle.forward(200) turtle.right(90) turtle.reset() </pre>	<pre> import turtle wn=turtle.Screen() def line(): turtle.forward(200) def angle(): turtle.right(90) for i in range(4): line() angle() </pre>
--	---



Python Turtle Cheat Sheets

Use these statements at the start of every program

```
import turtle
turtle.mode('logo')
turtle.speed(integer)

turtle.shape('turtle')
```

Getting ready to draw

Make all the turtle commands available to your program

Set the mode

Set the animation speed of the turtle. 1 = slowest, 10 = fastest. 0 turns off animation completely

Set the shape. You can also choose from: *arrow, square, circle, triangle* and *classic*

Movement

<code>turtle.forward(<i>distance</i>)</code>	Go forwards by amount <i>distance</i>
<code>turtle.backward(<i>distance</i>)</code>	Go backwards by amount <i>distance</i>
<code>turtle.right(<i>angle</i>)</code>	Turn right by <i>angle</i> degrees
<code>turtle.left(<i>angle</i>)</code>	Turn left by <i>angle</i> degrees
<code>turtle.home()</code>	Go home (0, 0) and face north
<code>turtle.goto(<i>x</i>, <i>y</i>)</code>	Go to position <i>x</i> , <i>y</i>
<code>turtle.setheading(<i>degrees</i>)</code>	Point in compass direction <i>degrees</i> . 0 is north, 90 is east, 180 is south, 270 is west

An example:

```
import turtle

turtle.mode('logo')
turtle.speed(10)
turtle.shape('turtle')

# Draw a square
turtle.forward(100)
turtle.right(90)
turtle.forward(100)
turtle.right(90)
turtle.forward(100)
turtle.right(90)
turtle.forward(100)
```

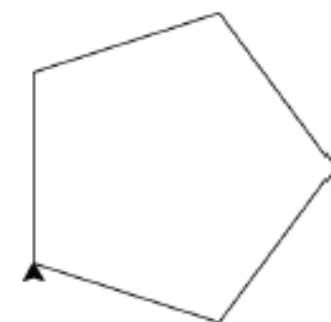
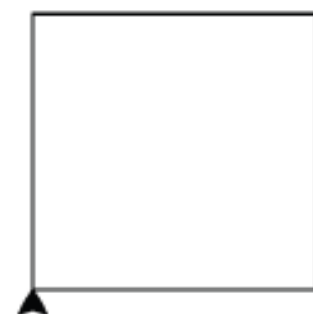
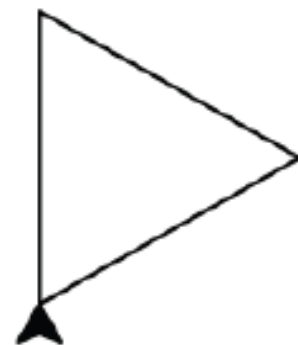
Iteration examples:

```
for i in range(3):  
    turtle.forward(100)  
    turtle.right(120)
```

```
for i in range(4):  
    turtle.forward(100)  
    turtle.right(90)
```

```
for i in range(5):  
    turtle.forward(100)  
    turtle.right(72)
```

Can you see a pattern between the numbers on the left, and the shapes on the right?



How could you use this knowledge to draw an octagon?

