

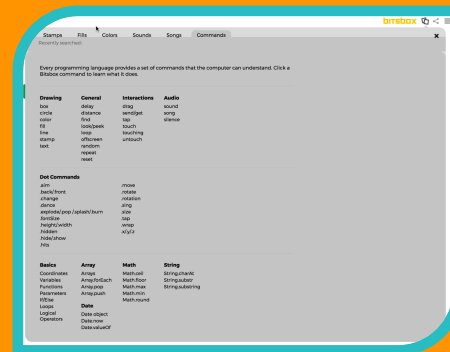
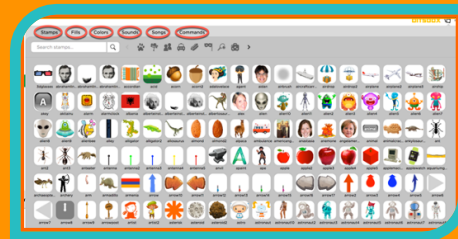
Learning

- To explore event-driven programming using a text-based programming language
- To understand the importance of decomposition (breaking a problem into smaller parts and solve one part at a time)
- To understand that variables contain values
- To use algorithms to develop a solution to a problem
- To translate algorithms into code
- To use abstraction and functions in programs
- To understand that apps are computer programs that are developed according to a plan
- To develop an app according to a plan
- To develop strategies for testing and debugging computer programs

Key Vocabulary

text based programming	Computer programs written in text (letters,numbers,punctuation) rather than visual blocks
decomposition	Splitting a task into smaller parts to make them easier to solve
variable	Names given to something we want a computer to store
algorithm	Steps to follow to achieve a task
program	Instructions written in a language (code) computers can understand
abstraction	Taking the detail out of something to make it easier to understand/solve
function	A set of grouped commands that can be used more than once
test	Execute a program to find out whether it works
debug	Finding and correcting problems with algorithms and programs

Examples



```
function changeColour() {
  color(look(x,y)) /* this 'looks' at the colour you've just tapped and changes all line colours */
  circle(100,990,20)
}
```

Key Questions

Where do you change the behaviour of an asset?	Script editor
Where do you interact with your app?	Emulator
Which events does your app use?	E.g. tap/shake/drag
What does this function do?	E.g. updates the score
What is this variable for?	E.g. stores the current colour of the paintbrush
What are comments in code useful for?	Explaining what sections of code do
Why is planning a program important?	To help design algorithms that will work before coding them
What strategies did you use for testing/debugging?	E.g. working systematically, testing parts of the app and correcting mistakes as you work