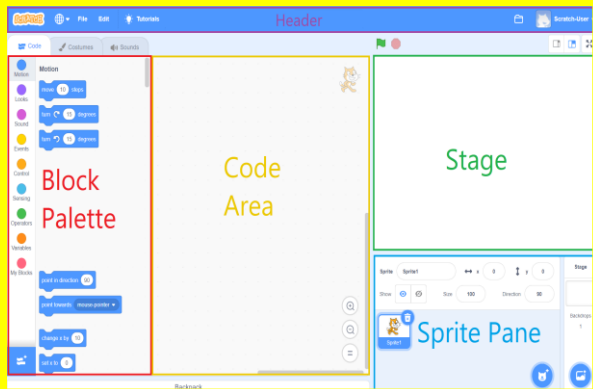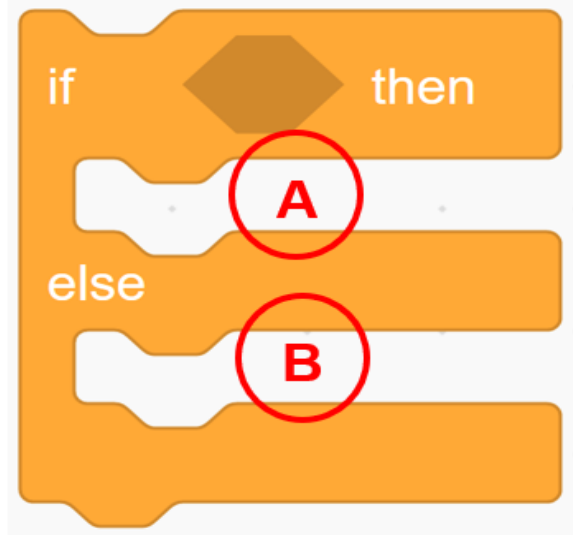# 7.3 Programming 1

The aim of this unit and the following unit ('programming 2') is to build your confidence and knowledge of the key programming constructs. The main programming concepts covered in this unit are sequencing, variables, selection, and count-controlled iteration.



| Code Category | Example Block | Description |
|---|---|---|
| Motion | move 10 steps | Code blocks that affect the position of a sprite on the stage |
| Looks | switch costume to costume2 | Code blocks that change the appearance of a sprite or stage |
| Sound | play sound Meow until done | Code blocks that make a sound |
| Events | when clicked | Code blocks that affect when a script begins to run |
| Control | repeat 10 | Code blocks that control selection and iteration in a script |
| Sensing | touching color ? | Code blocks that run when a specific action occurs |
| Operators | pick random 1 to 10 | Code blocks to run mathematical operations |
| Variables | set my variable to 0 | Code blocks that store a value to be used in a script |

## Conditions

- Basic conditions of operators > < =
- Logic operators (and, or, not)

| Term | Definition |
|---|---|
| Sequence | One of the three basic programming constructs. Instructions that are carried one after the other in order. |
| Selection | One of the three basic programming constructs. Instructions that can evaluate a Boolean expression and branch off to one or more alternative paths. |
| Iteration | One of the three basic programming constructs. A selection of code that can be repeated either a set number of times (count-controlled) or a variable number of times based on the evaluation of a Boolean expression (condition-controlled). |
| Variable | A value that can change depending on conditions or information passed to the program. |
| Boolean expression | An algebraic expression which has a Boolean value |
| Comparison operator | Used to compare two expressions |
| Computer bug | Code that causes your computer to behave in an unexpected way |
| Resilience | The capacity to recover quickly from difficulties |
| Subroutine | A block of code within a program that is given a unique, identifiable name. Supports code reuse and good programming technique. |
| Decomposition | Breaking down a problem into smaller, more manageable parts in order to make the problem easier to solve |
| List | A data structure that allows multiple pieces of data under a single name |
| Data structure | A way or organising and managing data in a programming language that ideally enables efficient access and modification of the data |



Runs the code in the block if the condition is true



Runs the code in gap A if the condition is true. Runs the code in gap B if the condition is not true (false)