**Switch-Case**

In previous classes we have learned about if and if-else conditions. These conditions are used to change the flow of the code such as, based on one condition, the program will behave in one way and another way when different conditions come. The if-else is not the only way to do so. We have one more method that can perform the same thing and make our code more readable. It is called the switch case statement.

In our home we have multiple switches for different appliances and when you press one switch a specific equipment runs. In the same way, in switch case statements we have a switching statement which is input value. Now, that input value can have many cases or types. Based on the case/ type of switching statement we execute a piece of the code.

We will define the switch(value) which means we’ll be switching the response based on the value. Then we need to define the possible case and response. Which is defined with case keyword

So in the parentheses of the switch statement we will write our cases. The case will be like a condition with which we will compare the user input.

Break statement is used to terminate a running code statement.

we have a default statement which gives a default response when the user input does not match with any of the cases. To write a default statement we simply write the default keyword and then colon. In the next line will write what happens when the user enters any other

**createInput()**

we will create an input dialog box to get the user input. We declare two variables—input and heading. Then using the **createInput**() function we will create a text input box on the canvas. Then we will specify the position of the input box using the **input.position()** function.

We will create a text heading for the text input box using **createElement()** function. In the createElement function we need to specify the heading size such as ‘h4’ and the text for the heading. Using the position function we position the heading on the canvas. We will also set the size and the alignment of the text.

Users can enter the value here; to read the value entered by the user, we will use **input.value()** function and store this in a variable.

**Lifetime:**

Time = Distance/Speed