Control Flow Statements:

You can control the flow of your Dart code using any of the following:

- if and else
- for loops
- while and do-while loops
- break and continue
- switch and case
- assert

Dart If Statement

The syntax of if statement is given below.

Syntax -

```
If (condition) {
    //statement(s)
}
```

The given condition is if statement will evaluate either TRUE or FALSE, if it evaluates true then statement inside if body is executed, if it evaluates false then statement outside if block is executed.

Dart if-else Statement

In Dart, if-block is executed when the given condition is true. If the given condition is false, else-block is executed. The else block is associated with the if-block.

Syntax:

```
    if(condition) {
    // statement(s);
} else {
```

```
    // statement(s);
}
```

Here, if -else statement is used for either types of result TRUE or False. If the given condition evaluates true, then if body is executed and if the given condition evaluates false; then, the else body is executed.

Dart if else-if Statement

Dart if else-if statement provides the facility to check a set of test expressions and execute the different statements. It is used when we have to make a decision from more than two possibilities.

Syntax

```
if (condition1) {
  // statement(s)
}
else if(condition2) {
  // statement(s)
}
else if (conditionN) {
  // statement(s)
}
else {
  // statement(s)
}
```

Here, this type of structure is also known as the else….if ladder. The condition is evaluated from top to bottom. Whenever it found the true condition, statement associated with that condition is executed. When all the given condition evaluates false, then the else block is executed.

Dart Switch Case Statement

Dart Switch case statement is used to avoid the long chain of the if-else statement. It is the simplified form of nested if-else statement. The value of the variable compares with the multiple cases, and if a match is found, then it executes a block of statement associated with that particular case.

The assigned value is compared with each case until the match is found. Once the match found, it identifies the block of code to be executed.

The syntax is given below.

Syntax:

```
 switch( expression )
 {
case value-1:{

      //statement(s)
      Block-1;
            }
                        break;
case value-2:{
                              //statement(s)
      Block-2;
                        }
                        break;
case value-N:{
                              //statement(s)
      Block-N;
                        }
                        break;
default:   {
      //statement(s);
                  }
}
```

```dart
import 'dart:io';

Run | Debug
void main() {

  print('enter the text');
  var txt = stdin.readLineSync();

  switch (txt) {
    case 'a':
      print('this is case 1');
      break;
    case 'b':
      print('this is case 2');
      break;
    default:
      print('This is default');
  }
}
```