Default structure of a dart console :

**Bin folder 🡪** contains the source code.

**Lib folder🡪** contains the library dependencies imported into the project.

**Test folder 🡪** default test to check your code.

In Linux, a dependency known as **FSWatch** can be used that is a monitoring tool, that monitors the code changes, and when we save the changes using **ctrl+s** then it will automatically run the code and reflect output In the terminal, we don’t need to run the code each time.

**Const🡪** the data that you specify while writing your code and it will be held inside memory like heap or stack or where the dart decides to place it, and you won't be able to modify it after creation. It does not allow the assignment to a new value and does not allow modifications to the value itself. For eg in the list.

Two types of mutability are there:

* reAssignment to a new value.
* Modifications to the value itself after it has been assigned.

The variable is anything that you assign a value to.

**Final Variable 🡪 Final keyword**

It doesn’t allow the re-assignment of the variable but modifications can be done to the initial value of the variable.

**Mutating Variables🡪 var keyword**

These variables allow both types of mutabilities, that it can be assigned to a new value possible plus modifications to the value can be made after it has been assigned.

Imp:

When we use the ‘var’ keyword then its value is assigned at the compile time, you assign the variable to a particular value, and based on the value it recognizes the type of variable.

Whereas when we use the ‘String’ keyword, it locks in the data type and specifies the dart that this variable has to be a string.

Constant values can be assigned to final variables but final values cannot be assigned to constant variables. Because the final variable is partially immutable.

**Late Variable 🡪** These variables are initialized only when they are used. When we assign the late variable to a function call. Then this variable will keep this function call in mind and when we use that variable in a print statement, then actually at that time this function call will be made.

**Type Promotion🡪**

Promoting the data type from less precise value to more precise value. Specifying the value of a variable by a more precise data type. That using double instead of int for age by specifying double as the data type is the type promotion of that data type.

Maps 🡪 contains a key-value pair, passing the key to the map, one can access the value.

Sets🡪 Sets are like lists but they do not allow duplicate values.

Dynamics and symbols data types.