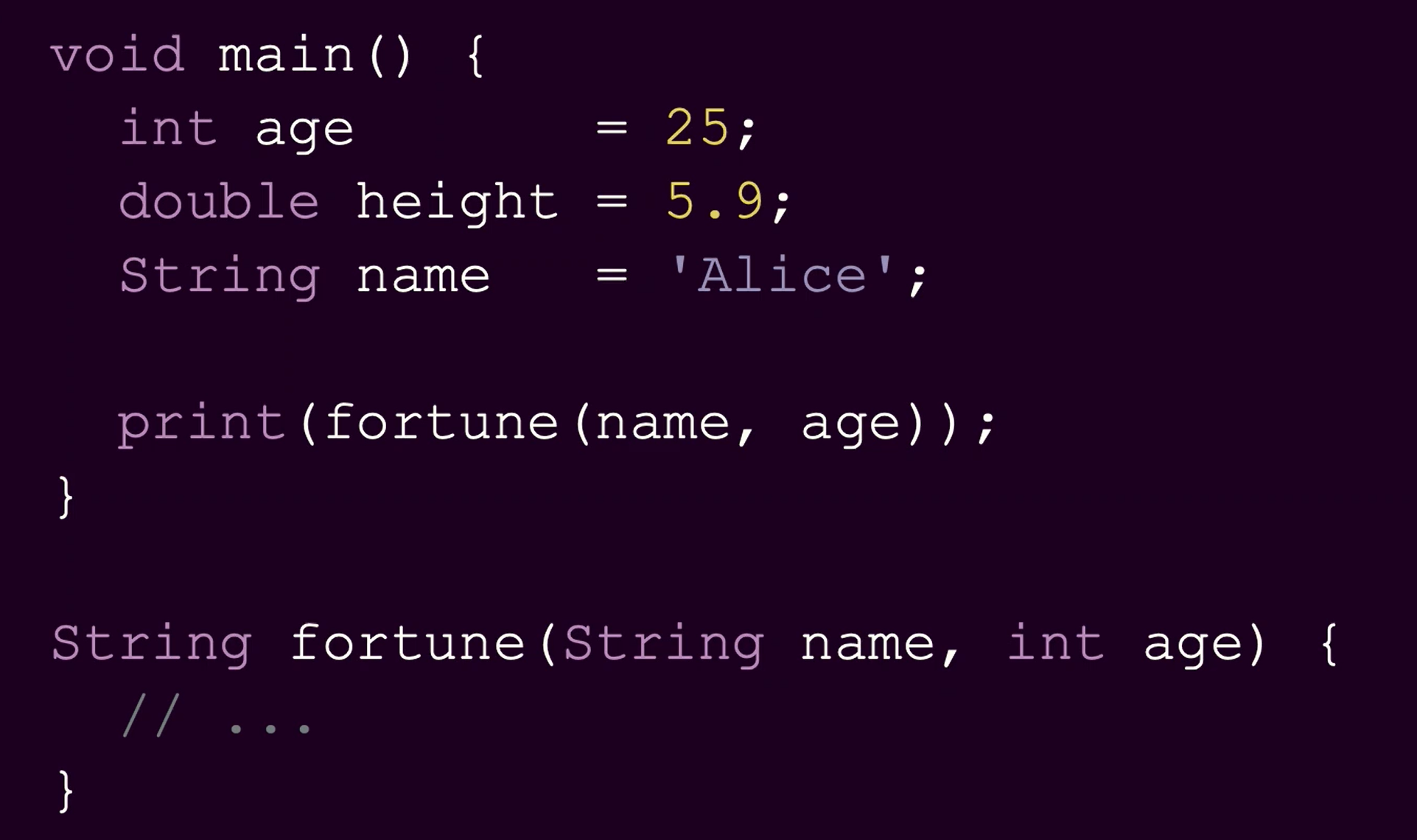
CS442 - Mobile app dev

1/12/2024

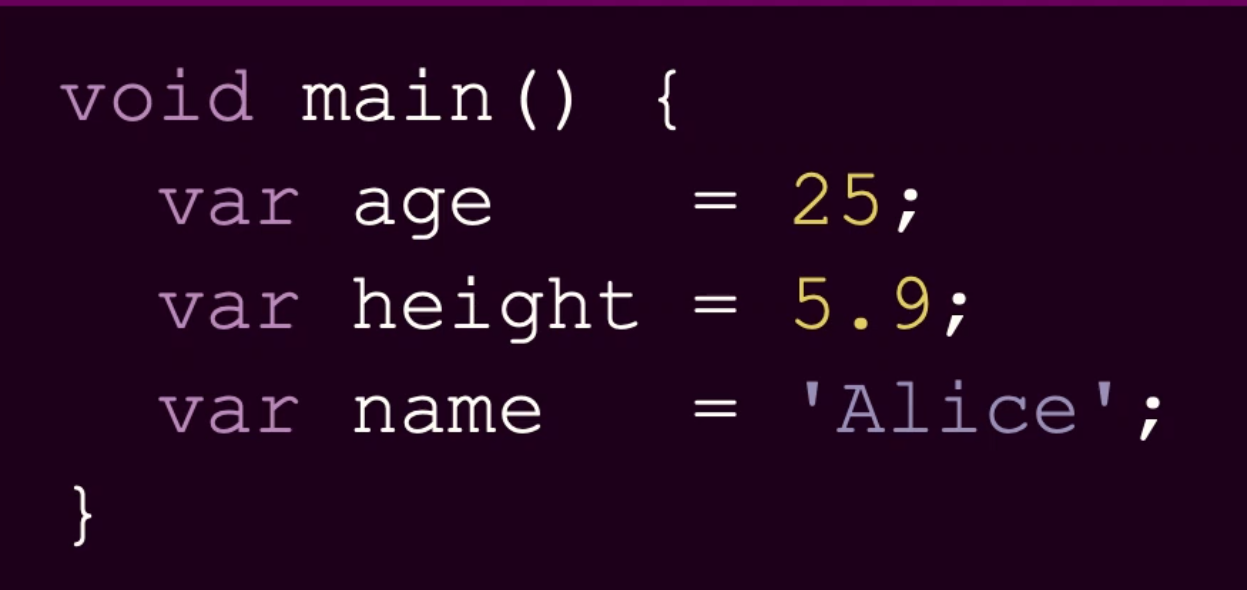
* Flutter is a framework that uses Dart to develop apps
* Language - DART - developed by Google 2013 as a replacement for Javascript
* Reason - certain specific needs of mobile apps are not natively available in Javascript and there are lot of changing versions in JS
* Mobile app mostly needs constant display of information whether or not used and it needs a native language for asynchronous programming
* Dart - help in easy debugging--->when we find an error while testing, -->Dart: we can update that part of the code and then rerun the testing.
* The data collected until the point of failure is still stored and can be stored and resumed.
* --->Other frameworks: - build the entire thing new with the update in code and do the testing again.
* Dart is transpiler friendly --->Transpliler-->converting a code in one higher order language to another higher order language. (Compliler-->high to machine(low))
* flutter libraries used in the app dev are also written in other languages and stored as libraries which are used by transpilers.
* So, even if a particular framework needs any other language other than Dart, we can still write code in Dart and then transpile into another lang(JS) and run the code in S
* Dart is very much suitable for cross platform dev for this reason. it can be used to run in Window/andriod/ios/Linus/browser extension.
* When a language is compiled during run time, it gets translated to the native language on which the engine is built upon -->
* flutter engine ,the base code is written in C++ (fast language)-->therefore, although we write in Dart, the code executes faster because it runs in C++ engine.
* Statically typed - every variable is assigned datatype - it cannot be changed.
* Type safe- if there is a mismatch of types with variable and values - compiler throws Type error.



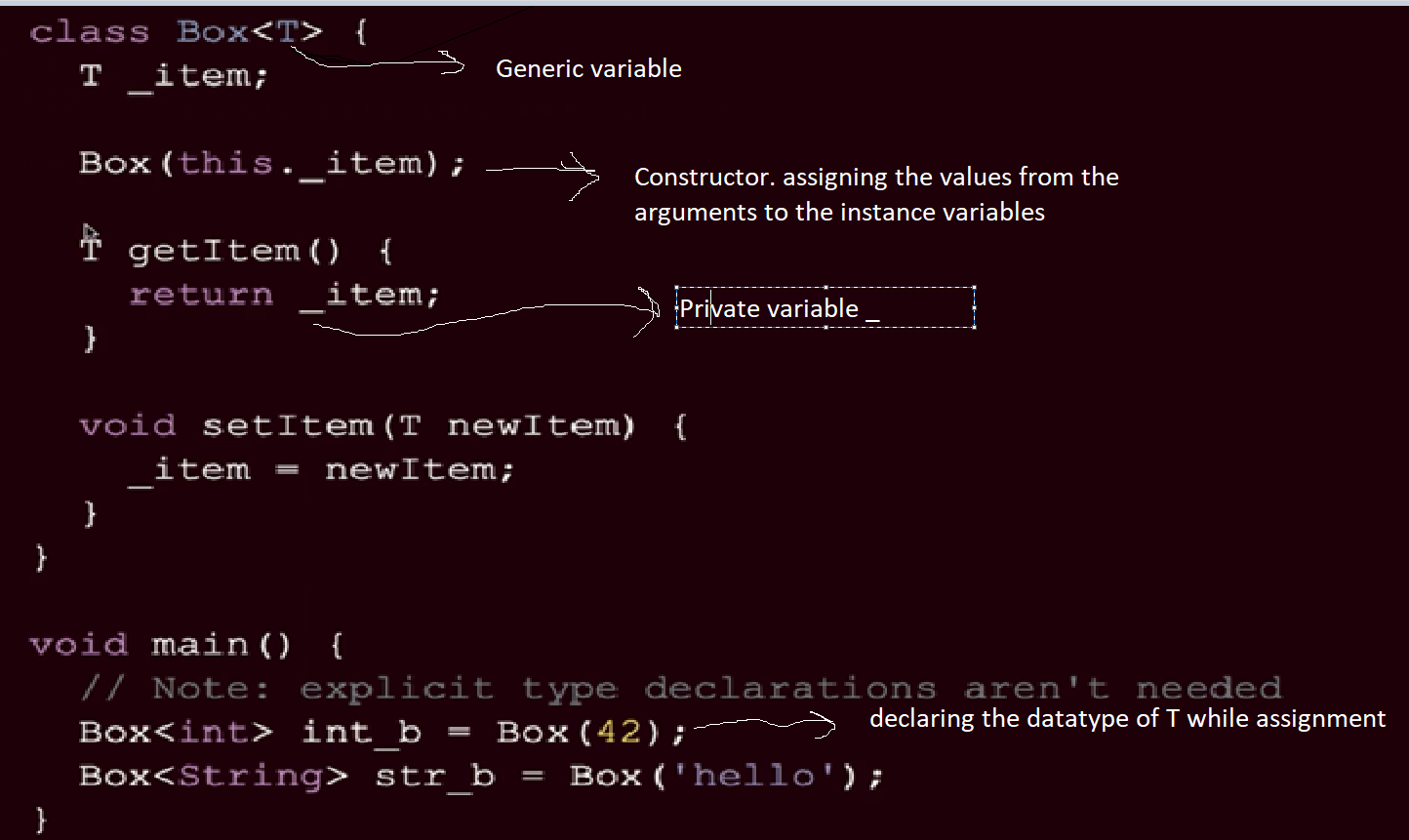
* Even the value type returned by the function is also predeclared. Statically typed and type safe is helpful to show any possible errors in the compilation itself
* Null pointer exception – This is the most prevelant error In all programming languages – because a string datatype can take a string or null generally.
* Here in Dart, String does not accept null. We have to give a value when we access a string variable /function with such parameters.
* Thus, Dart is NULL safety (A subset of Type safety)

Type inference:

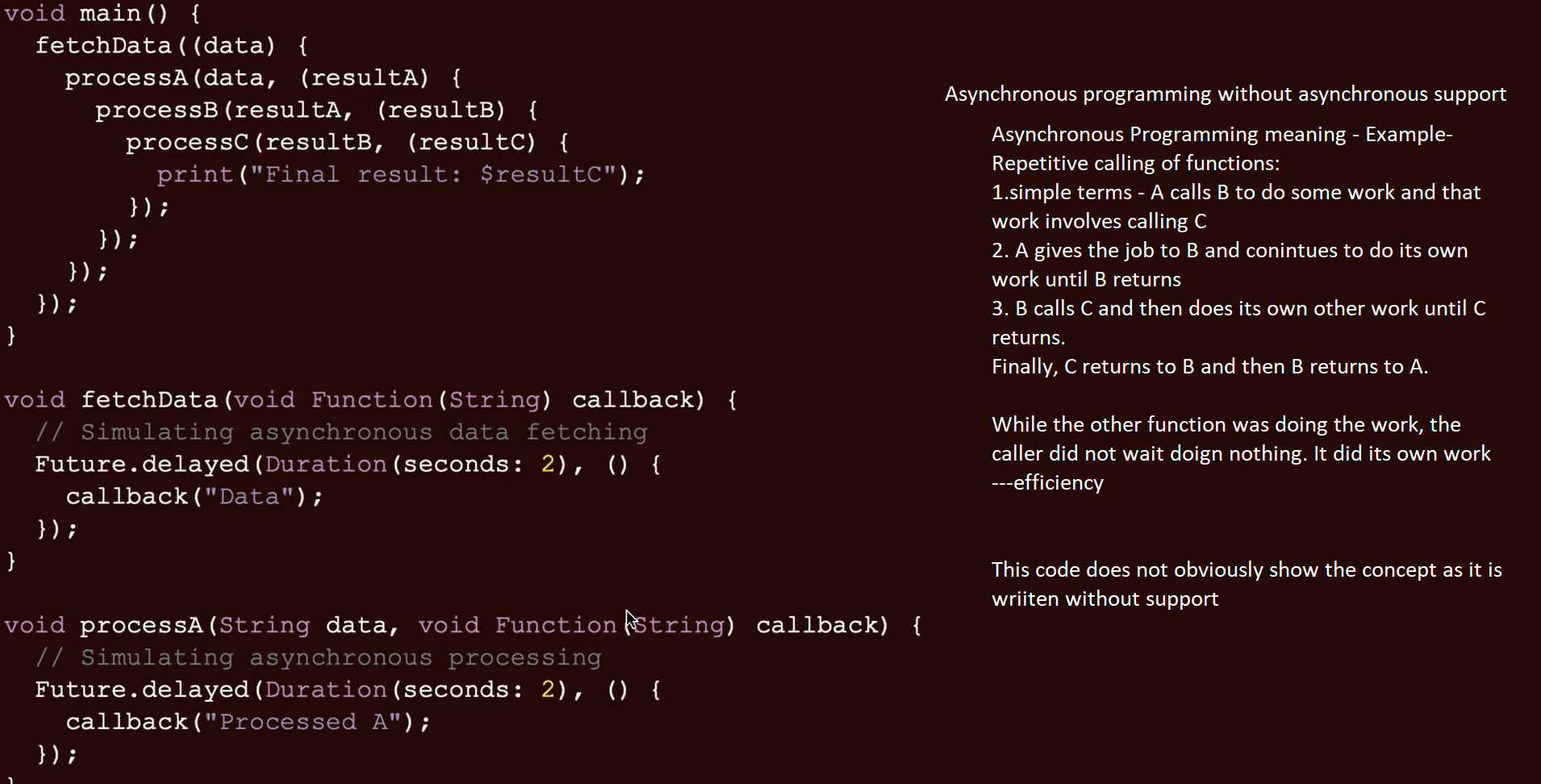
* The compiler detects the type based on the value assigned and declares the datatype (sounds like Python..but Dart still doesn’t allow equating/assigning different datatype values). We can just mention var while declaring.
* However, once a type assigned , we cannot assigned a different type value to this variable. It is just that we don’t assigned but the compiler does and it cannot be changed.
* In below example, we cannot equate or assign name to age , because they are of diff types. Dart is statically typed
* Note - we can make a Dart variable as dynamically typed but not recommended unless necessary (this would turn of type safety at compiler level for this variable)



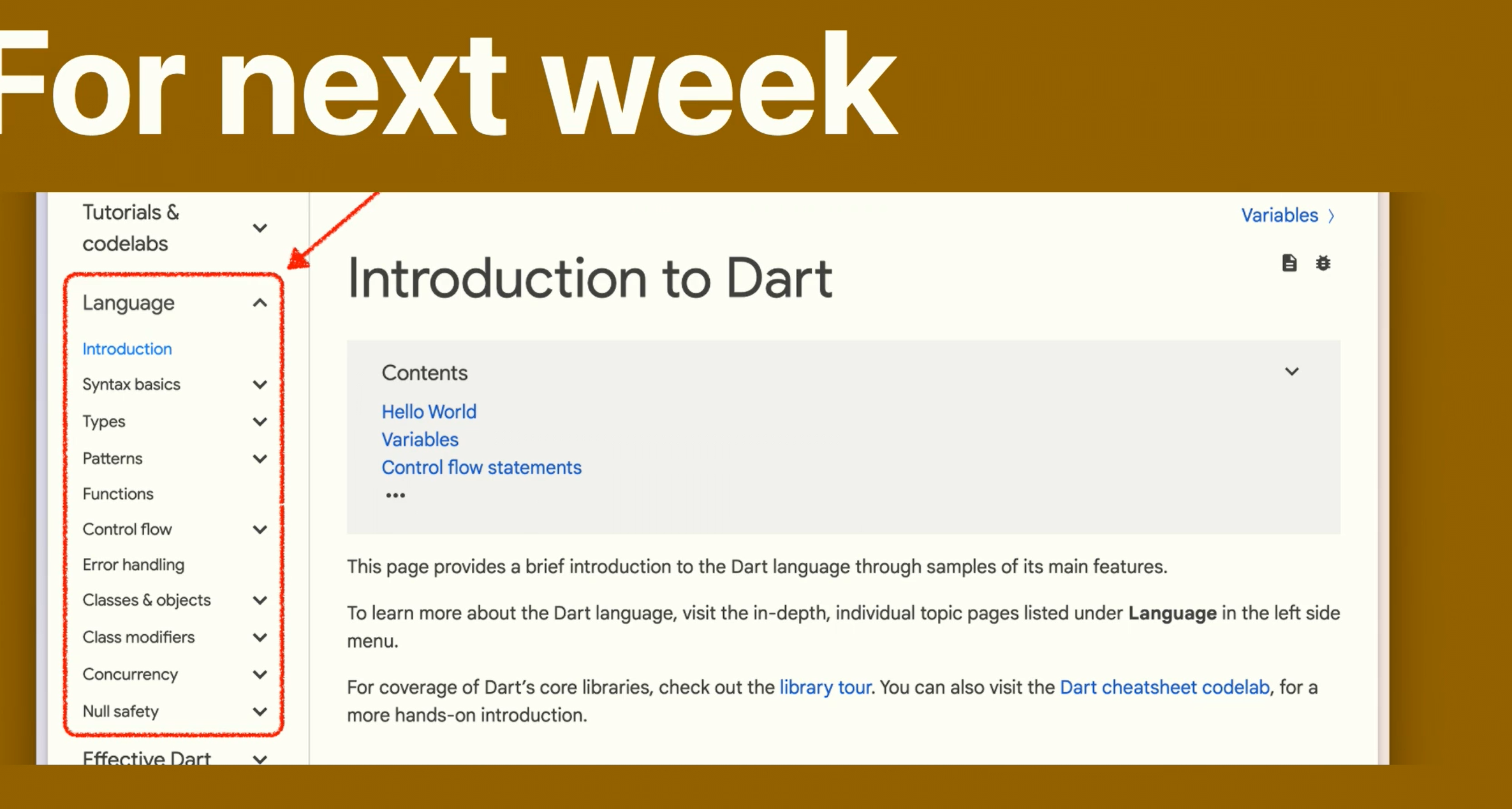
* Dart is **fully object oriented** – like Python – int,float and all datatypes can be referenced and has objects. (unlike Java – which has primitive datatypes (int,float) and reference types(strings..))
* Single inheritance 🡪A class can inherit only from another one class. It doent allow multiple inheritance to avoid the confusion of 2 super classes having same methods and which one would the child feature.
* Without this confusion, we can use multiple inheritance by using the DART feature – **mixins - >** which allows calling a class code into a cass without inheriting the parent class.
* GENERICS: - This is an alternative of Polymorphism (where we use function override, parameter override). This can hold a list of objects irrespective of the datatype
* In below example, T is the generic variable.
* \_ in Dart means it is a private variable (Encapsulation)
* Advantage of not using Polymorphism but a Generics instead - attaching the datatype to the data structure/container/object instances even though a variety of datatypes are assigned for the instances of the same class

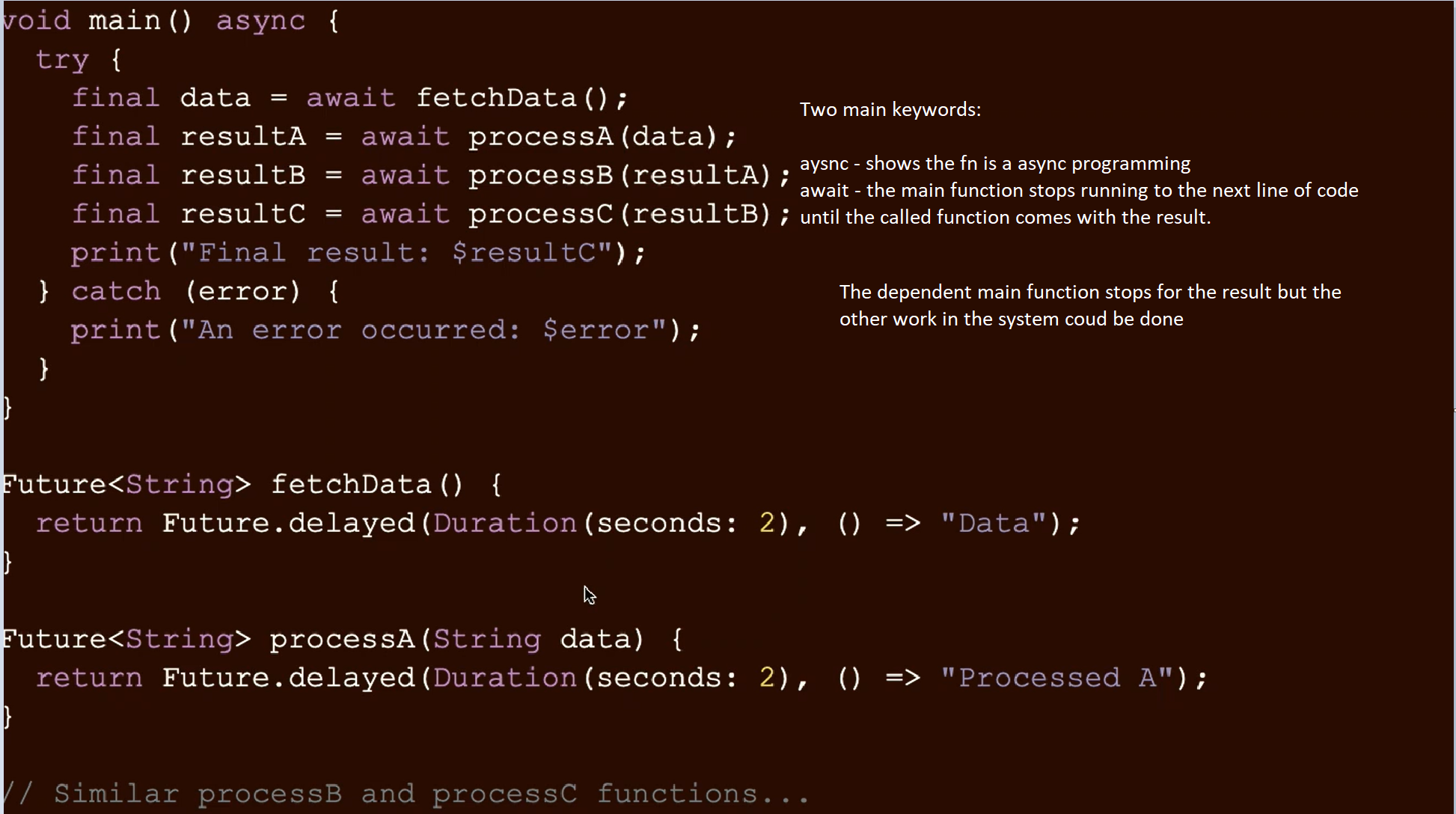


* Garbage collected - A GC-enabled programming language includes one or more garbage collectors (GC engines) that automatically free up memory space that has been allocated to objects no longer needed by the program.
* ASYNCHRONOUS PROGRAMMING –



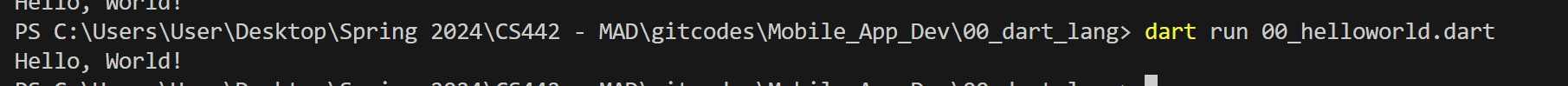
Note – to above screenshot - But the dependent function will wait for the result expected





1/17/2024 Dart Basics

* Cloned github, connected to VScode
* Trying out various commands
* Converting the dart file into js/any executables🡪Transpile 🡪commad is “dart compile js <filename> “🡪this now creates a new js file out.js with all necessary functions and can be executed like any other js file to get the same output
* Running from terminal “dart run <filename>”



Follow program files