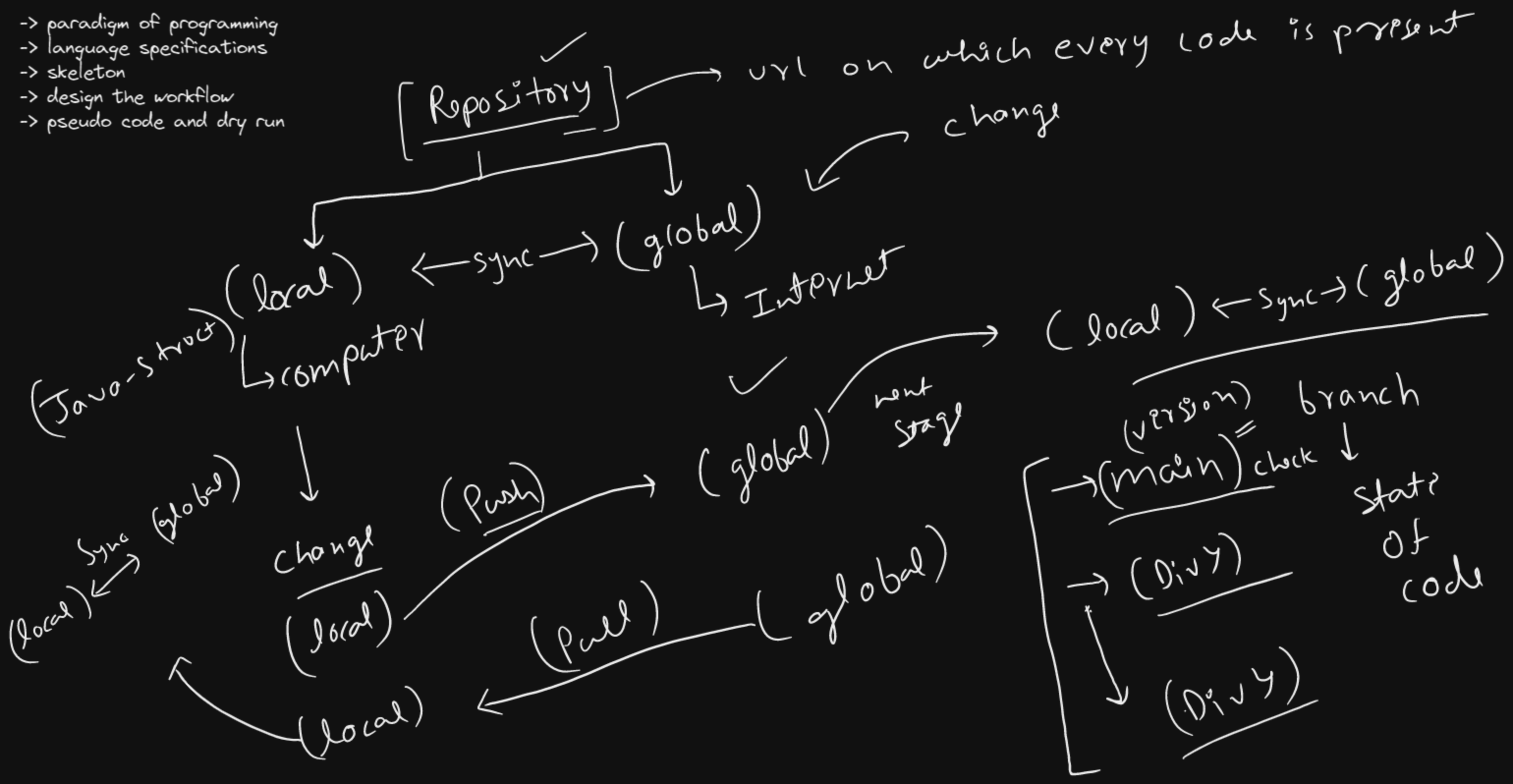
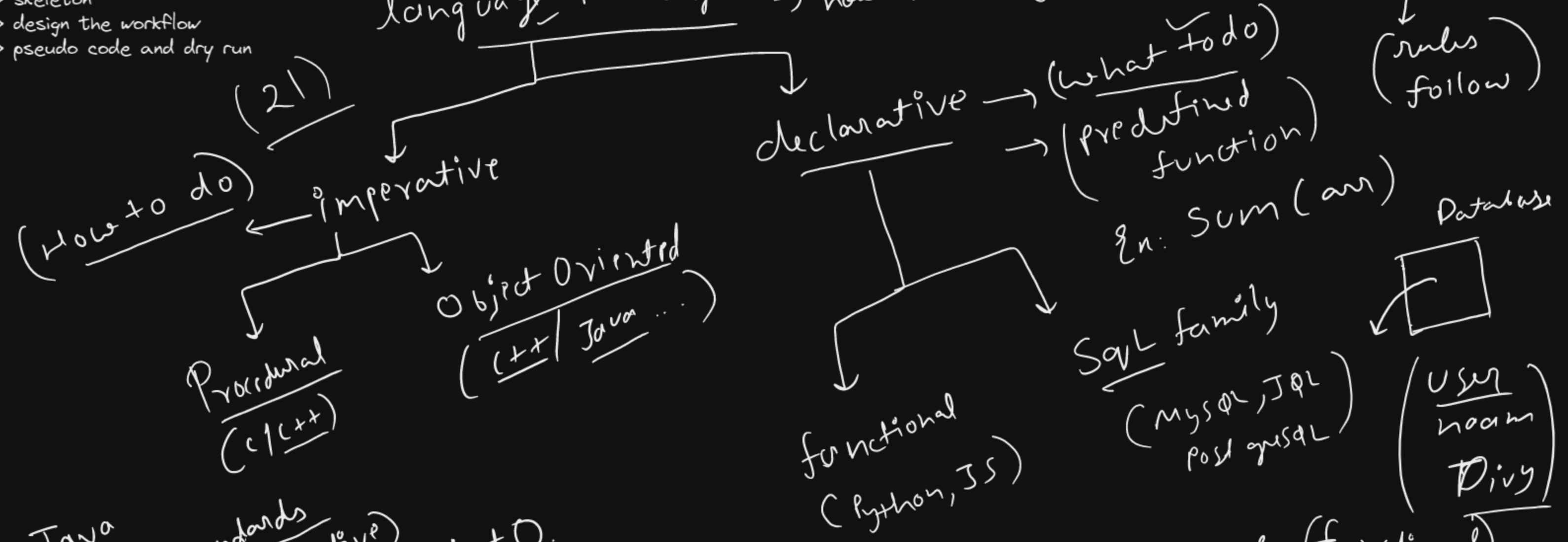


- > paradigm of programming
- > language specifications
- > skeleton
- > design the workflow
- > pseudo code and dry run



- > paradigm of programming
- > language specifications
- > skeleton
- > design the workflow
- > pseudo code and dry run

language Paradigm → how should you write code (Standards)



Java  
 ↳ 2 standards  
 (Imperative)  
 ✓ Procedural, Object O.

Exception

JS → Procedural, OOP, (functional)  
 Pyth → " " "  
 Java\* → " " "  
 " " "

- > paradigm of programming ✓
- > language specifications ✓
- > skeleton
- > design the workflow
- > pseudo code and dry run

Java use cases  
 fault tolerance → web (enterprise)  
 → Big Data  
 → Concurrent programming

(Spring Boot)  
 web framework

Statically ✓

→ 0, 1

↑ datatypes

→ int, double, ...

Dynamic  
 ↳ Runtime  
 ↳ slow

Static  
 ↳ compile time  
 ↳ fast

(language typed)

① Statically

↳ Datatype

↳ Strict type

② Dynamically ✓

↳ No Datatype needed  
 ↳ loose type

Java  

```
int x = 10
String y = "hello"
```

JS  

```
var x = 10
var y = "hello"
```

 ↳ number  
 ↳ string

- > paradigm of programming ✓
- > language specifications ✓
- > skeleton ✓
- > design the workflow
- > pseudo code and dry run

Extension

↳ .java

(Javac) ✓

.class

(JVM)

Java

run

↳ Java is a statically typed interpreted or imperative paradigm

hybrid language which  
↳ [compiler]

Structure of Java Program

- ↳ ① .java file should have a class
- ↳ ② only 1 public is there in .java
- ↳ ③ main function is the entry point

classes

↳ public

↓  
function (main)

[.java → [class → method]]

- > paradigm of programming ✓
- > language specifications ✓
- > skeleton ✓
- > design the workflow
- > pseudo code and dry run

Extension

↳ .java

(Javac) ✓

.class

(JVM)

.Java

run

↳ Java is a statically typed interpreted or imperative paradigm

hybrid language which  
↳ [compiler]

Structure of Java Program

- ↳ ① .java file should have a class
- ↳ ② only 1 public is there in .java
- ↳ ③ main function is the entry point

classes

↳ public

↓  
function (main)

[.java → [class → method]]