# Use of Languages in Android Applications

***Asan Urazimanov***

*27 November 2018*

*Software Composition Seminar*
*University of Bern*

# RELEVANCE

Android smartphone market share > 85%

Used for sensitive applications
*(e-banking, healthcare, etc.)*

Languages influence app development

# MOTIVATION

## LEVERAGING INFORMATION

Not much information is available about the use of languages

"Missing generalization” of languages in Android development

## IMPROVING CODE QUALITY

Problems may arise due to the use of complex languages, or their improper integration

Support of practical use

# ASSESSMENT TOOL

## cloc

github.com/AlDanial/cloc v 1.76  T=0.10 s (667.6 files/s, 57257.8 lines/s)

| Language | files | blank | comment | code |
|---|---|---|---|---|
| Java | 23 | 442 | 478 | 2262 |
| XML | 40 | 221 | 564 | 1736 |
| Groovy | 3 | 10 | 49 | 47 |
| Markdown | 1 | 24 | 0 | 42 |
| YAML | 1 | 5 | 7 | 14 |
| ProGuard | 1 | 2 | 15 | 0 |
| SUM: | 69 | 704 | 1113 | 4101 |

# Structure

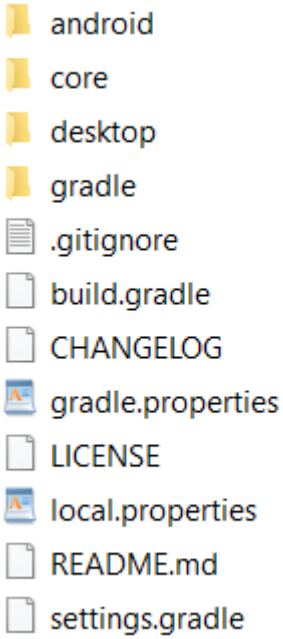
File structures used in projects

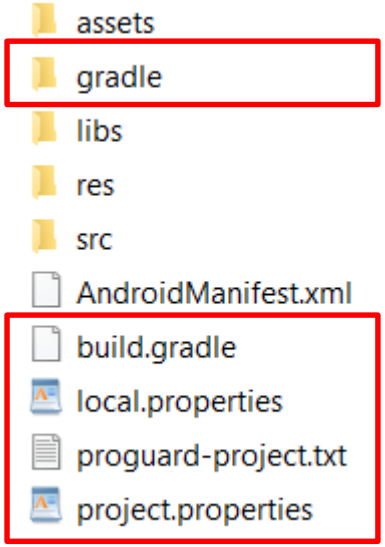
# STRUCTURE

- android
- core
- desktop
- gradle
- .gitignore
- build.gradle
- CHANGELOG
- gradle.properties
- LICENSE
- local.properties
- README.md
- settings.gradle

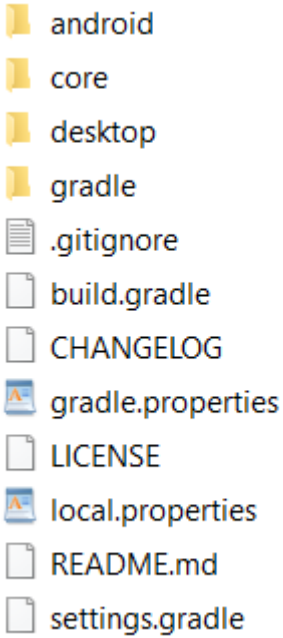
- assets
- gradle
- libs
- res
- src
- AndroidManifest.xml
- build.gradle
- local.properties
- proguard-project.txt
- project.properties

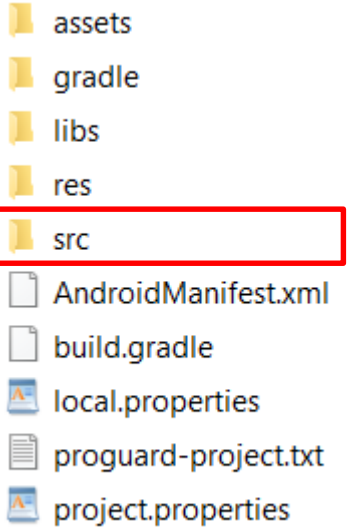
# STRUCTURE


android
core
desktop
gradle
.gitignore
build.gradle
CHANGELOG
gradle.properties
LICENSE
local.properties
README.md
settings.gradle

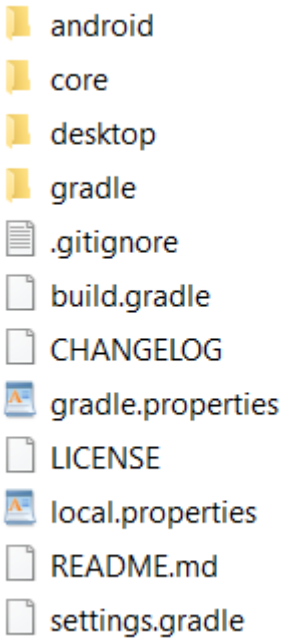

assets
gradle
libs
res
src
AndroidManifest.xml
build.gradle
local.properties
proguard-project.txt
project.properties

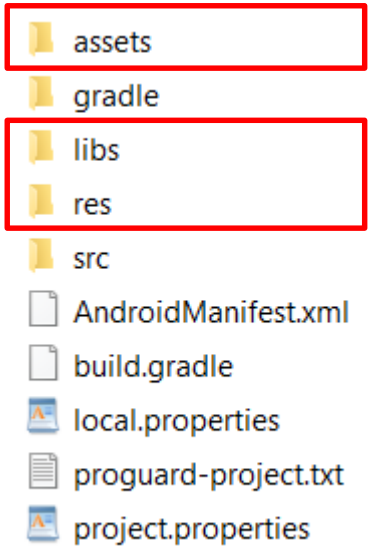
# STRUCTURE


android
core
desktop
gradle
.gitignore
build.gradle
CHANGELOG
gradle.properties
LICENSE
local.properties
README.md
settings.gradle

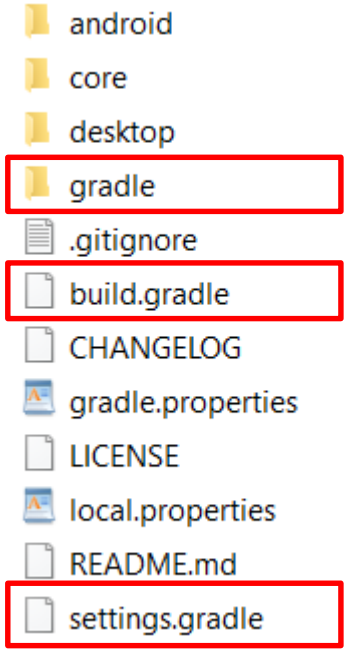

assets
gradle
libs
res
src
AndroidManifest.xml
build.gradle
local.properties
proguard-project.txt
project.properties

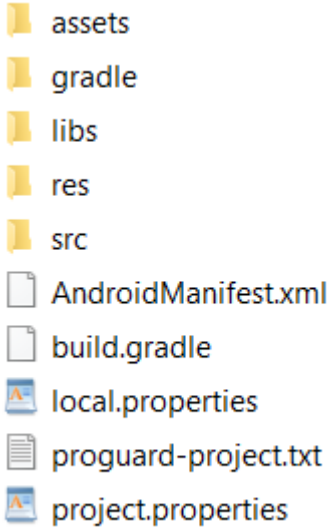
# STRUCTURE

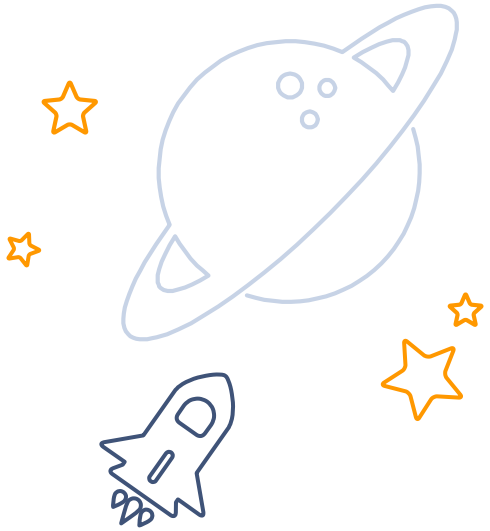

android
core
desktop
gradle
.gitignore
build.gradle
CHANGELOG
gradle.properties
LICENSE
local.properties
README.md
settings.gradle


assets
gradle
libs
res
src
AndroidManifest.xml
build.gradle
local.properties
proguard-project.txt
project.properties

# STRUCTURE


android
core
desktop
gradle
.gitignore
build.gradle
CHANGELOG
gradle.properties
LICENSE
local.properties
README.md
settings.gradle

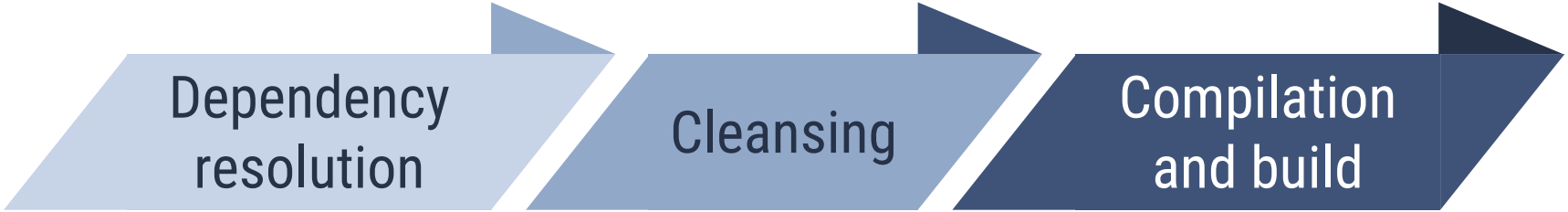

assets
gradle
libs
res
src
AndroidManifest.xml
build.gradle
local.properties
proguard-project.txt
project.properties



# BUILD TOOLS

Features and differences

# BUILD TASKS




**Dependencies: JUnit,  Hamcrest**

# ANT


| | |
|---|---|
| `ivy.xml` | Ivy dependencies specified in `ivy.xml` |
| `build.xml` | build script |

# MAVEN

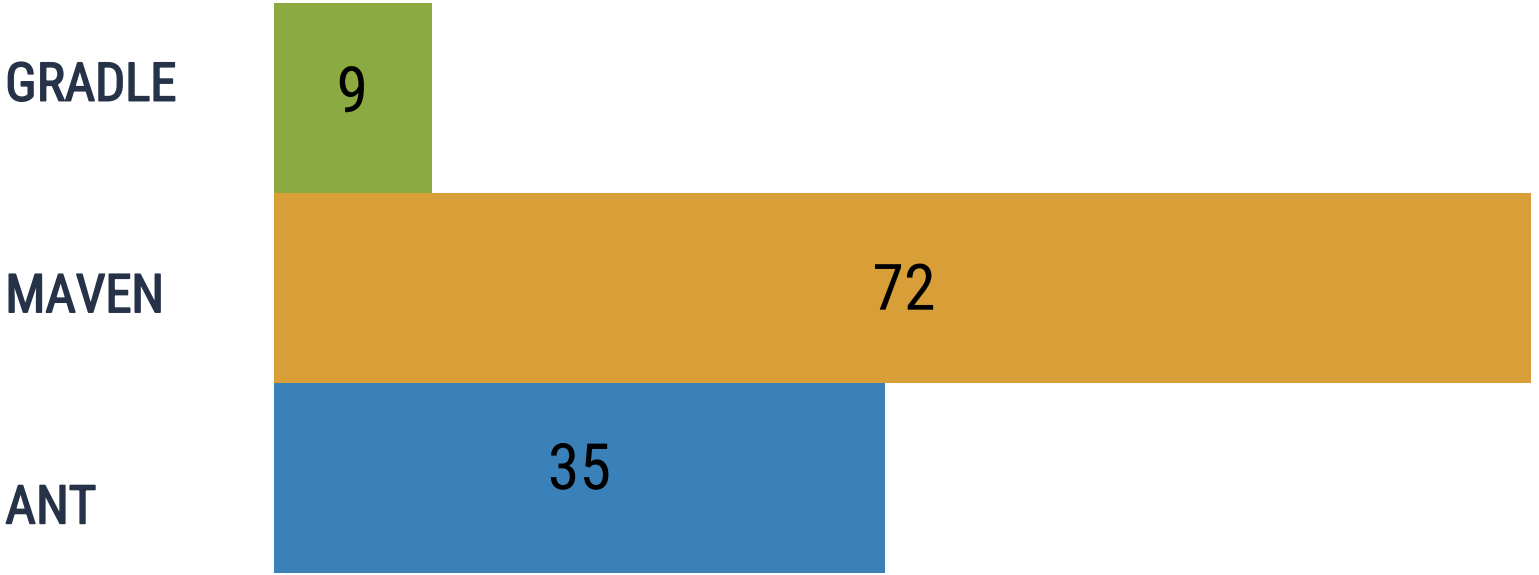

| | |
|---|---|
| `pom.xml` (main config) | Compiles project by using parameters |
| `pom.xml` (supplemental config) | Plugins |

# GRADLE


`build.gradle`

Custom DSL based on Groovy

# LINES OF CODE



# BUILD TOOLS OVERVIEW

## Ant

Full control
over the build process.

Build scripts use XML.

## Maven

Flexibility.

Strictly structured,
highly standardized.
Complicated.

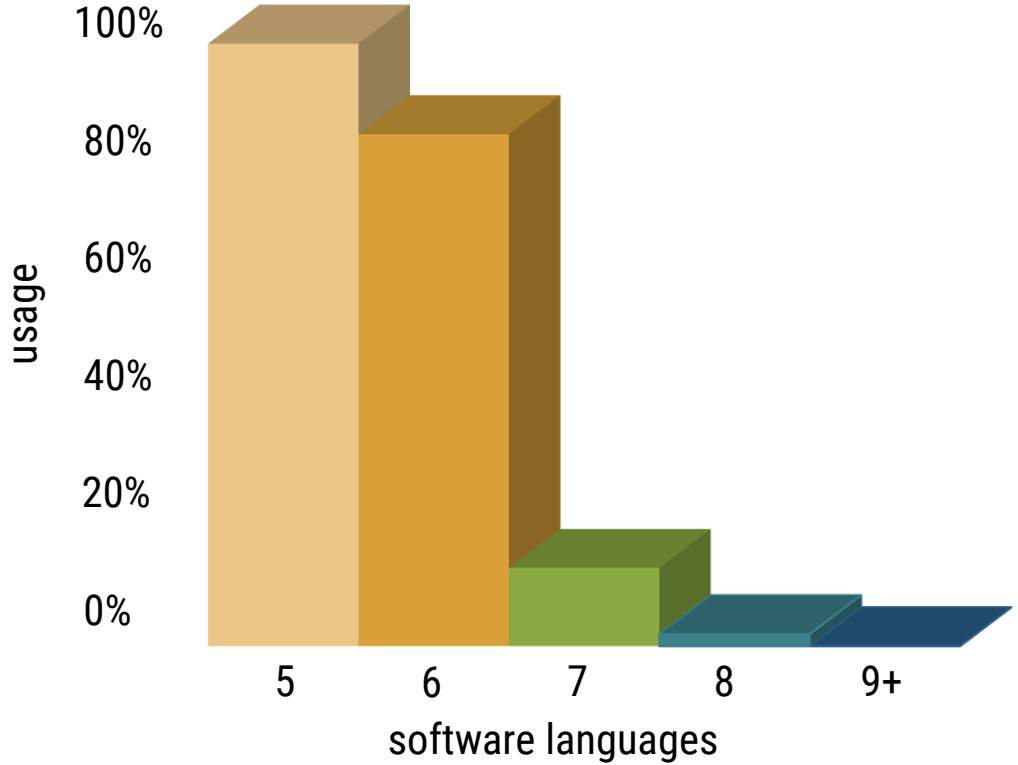
## Gradle

Provides all features.

More intuitive and shorter
build scripts.

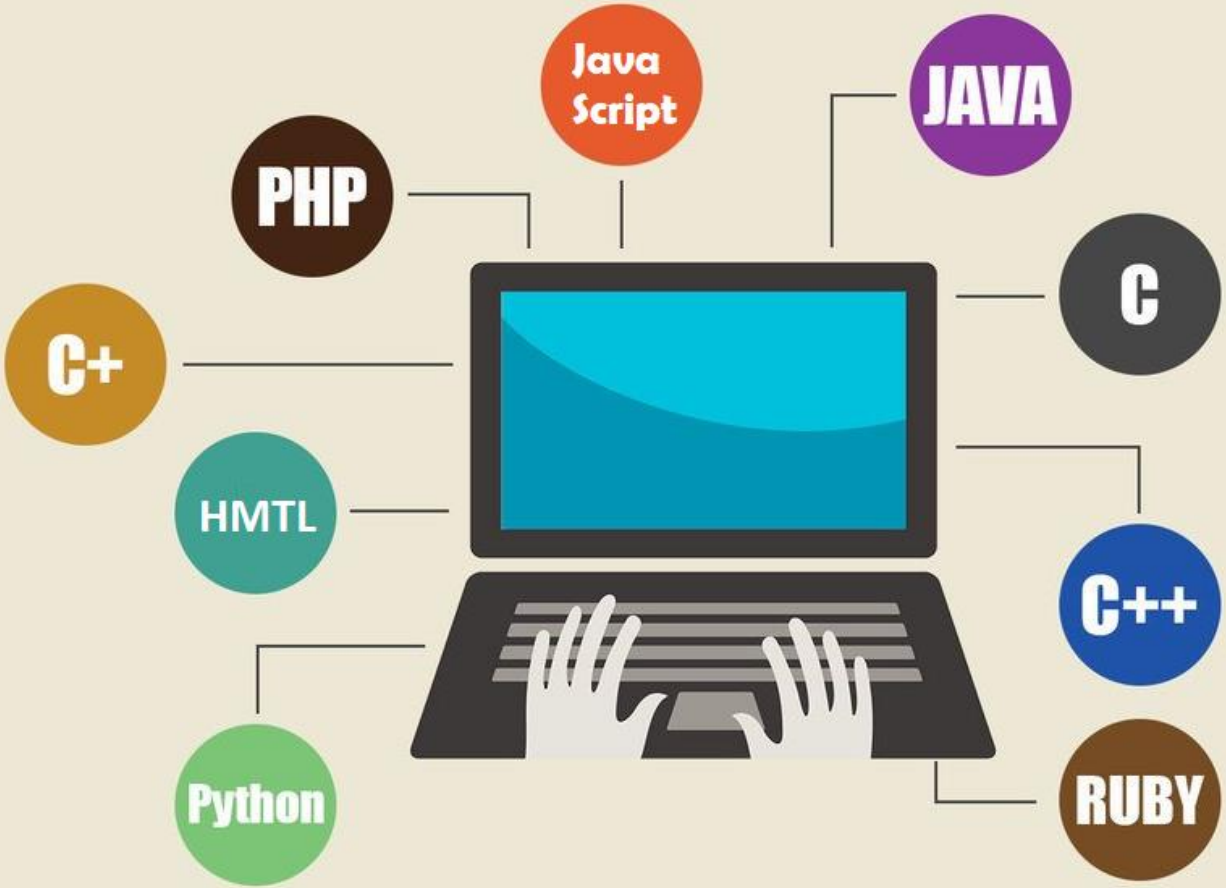
# PROGUARD

# PROGUARD

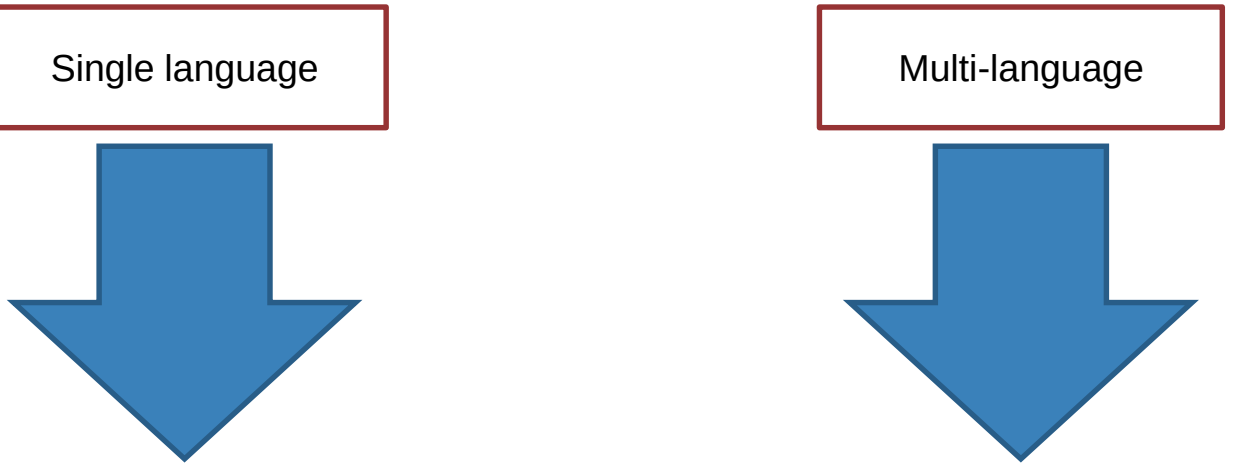


# LANGUAGES

Features and differences


Java Script
JAVA
PHP
C
C+
HMTL
C++
Python
RUBY

# XML



## JAVA

well-established language

## KOTLIN

new programming language

- less lines of code
- data classes
- null safety

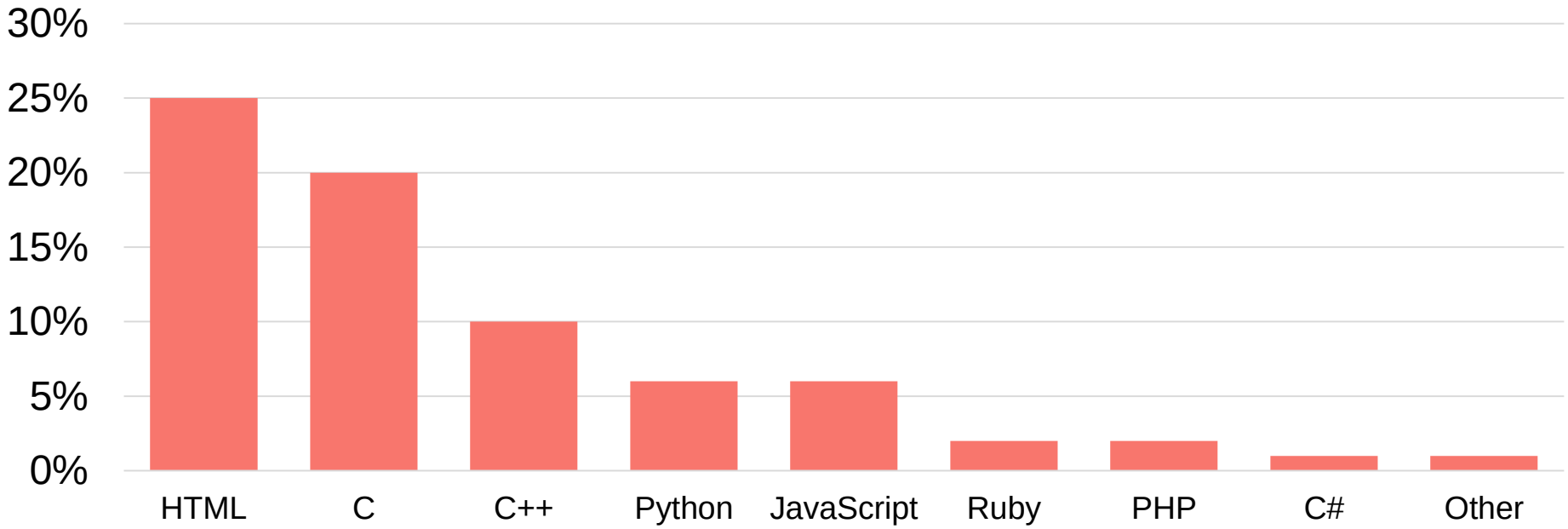
# BUILD TIME DIFFERENCE

| [s] | Without Gradle | With Gradle |
|---|---|---|
| JAVA | 10.82 | 9.17 |
| KOTLIN | 13.12 | 11.44 |

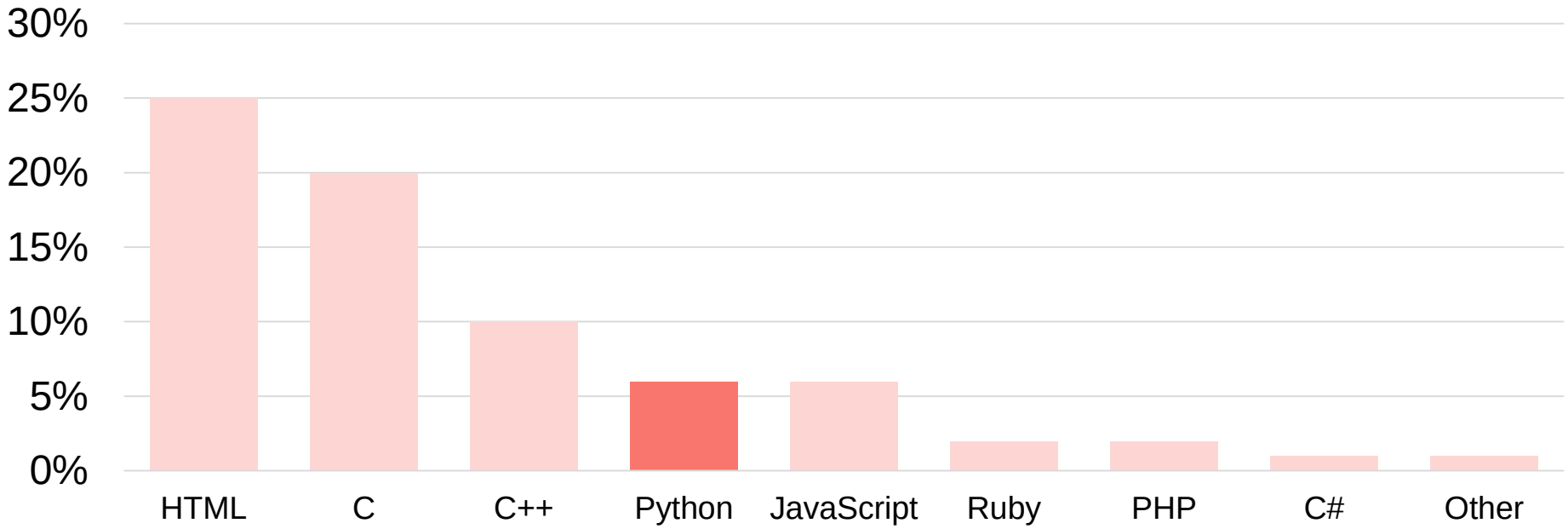
22%

13%

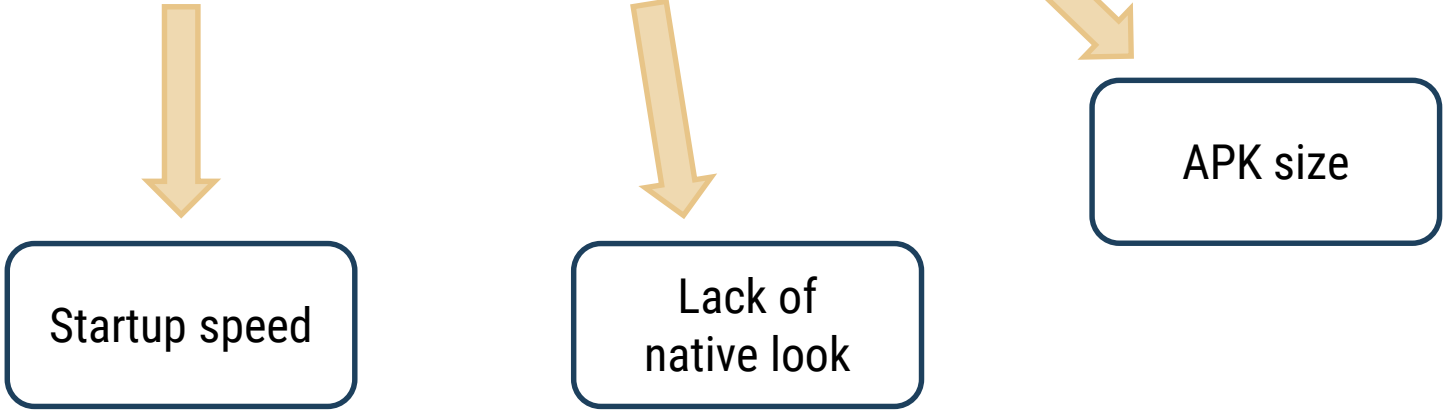
# LANGUAGE USAGE



# LANGUAGE USAGE

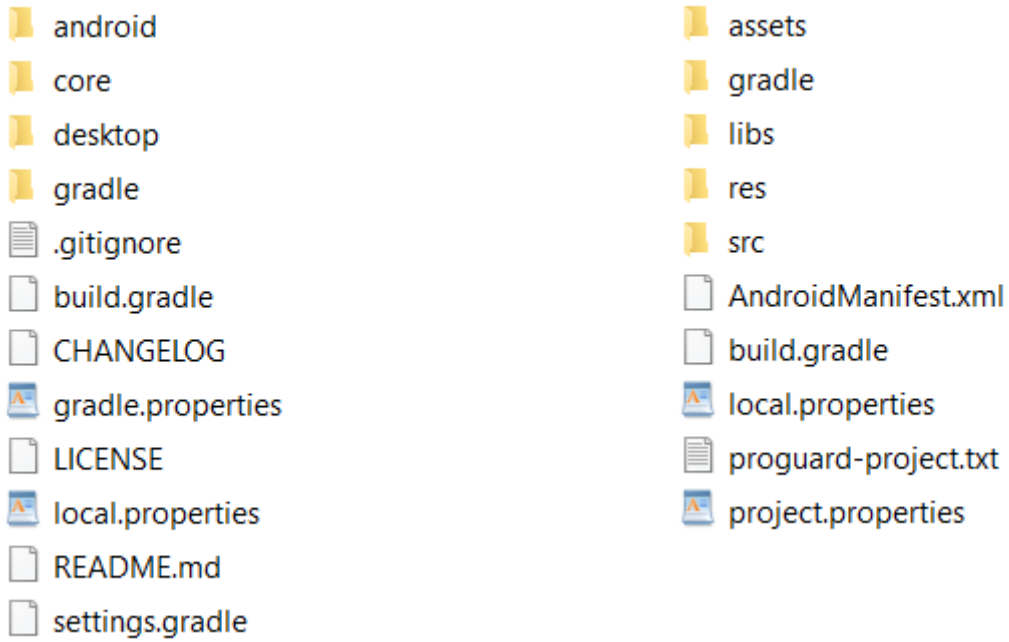


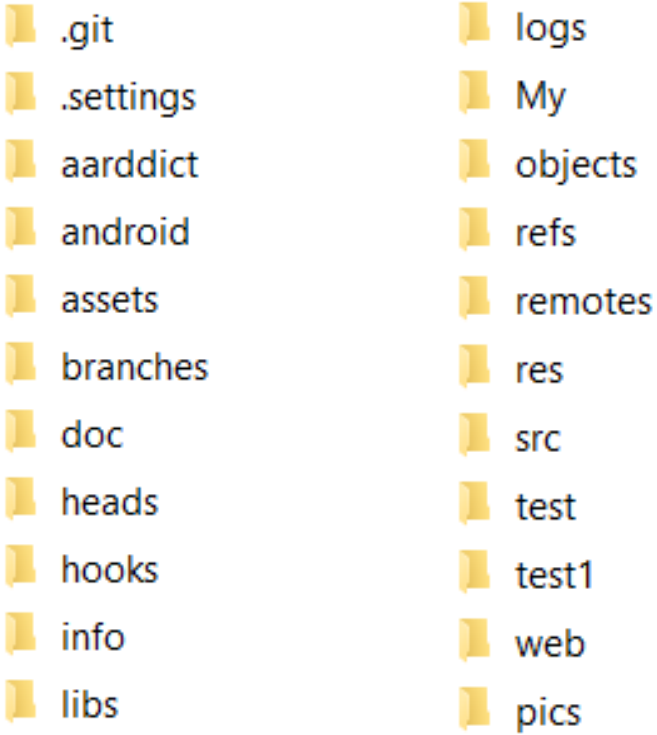
# Why Python?

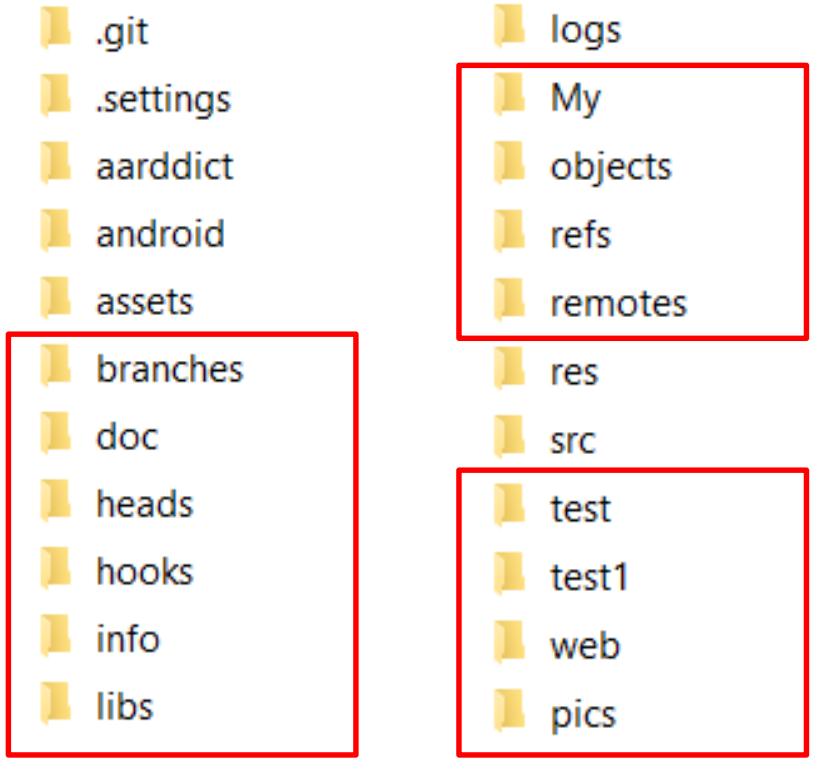


# AMAZING DISCOVERIES

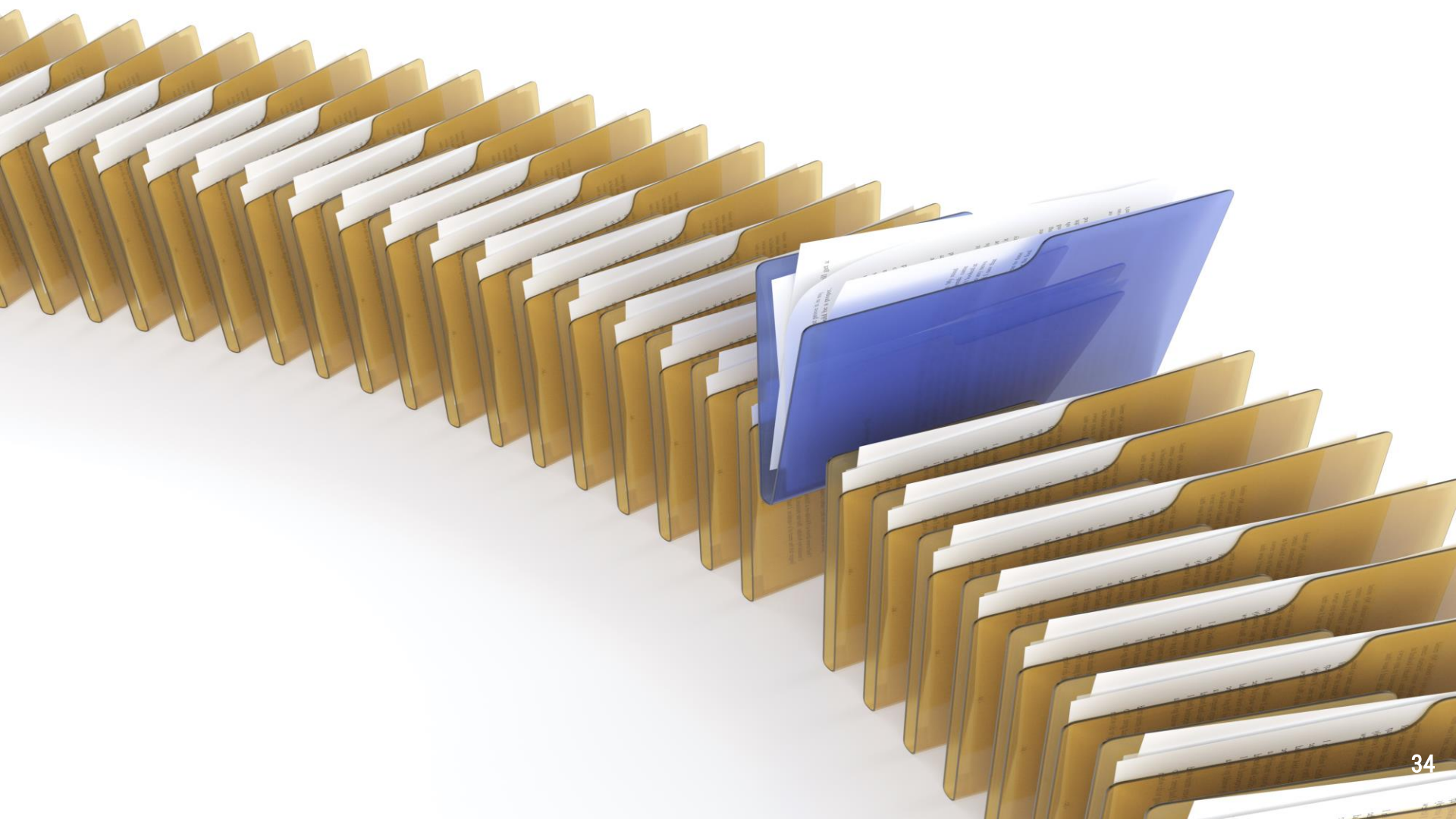
# 1


android
core
desktop
gradle
.gitignore
build.gradle
CHANGELOG
gradle.properties
LICENSE
local.properties
README.md
settings.gradle
assets
gradle
libs
res
src
AndroidManifest.xml
build.gradle
local.properties
proguard-project.txt
project.properties

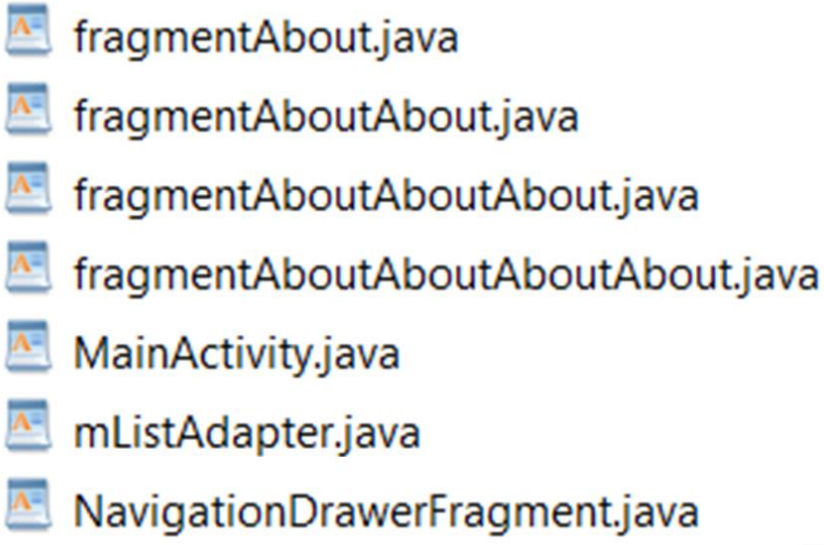

.git
.settings
aarddict
android
assets
branches
doc
heads
hooks
info
libs
logs
My
objects
refs
remotes
res
src
test
test1
web
pics

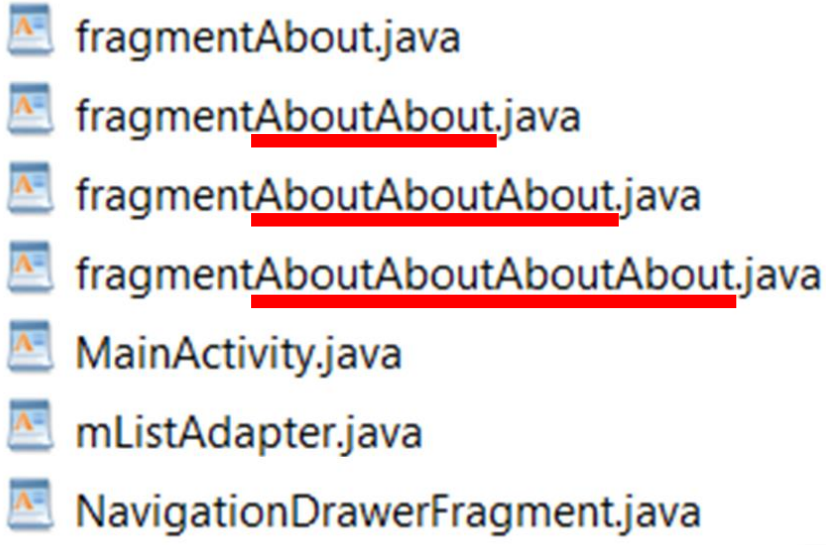

.git
.settings
aarddict
android
assets
branches
doc
heads
hooks
info
libs
logs
My
objects
refs
remotes
res
src
test
test1
web
pics

# 2



# 3


fragmentAbout.java
fragmentAboutAbout.java
fragmentAboutAboutAbout.java
fragmentAboutAboutAboutAbout.java
MainActivity.java
mListAdapter.java
NavigationDrawerFragment.java

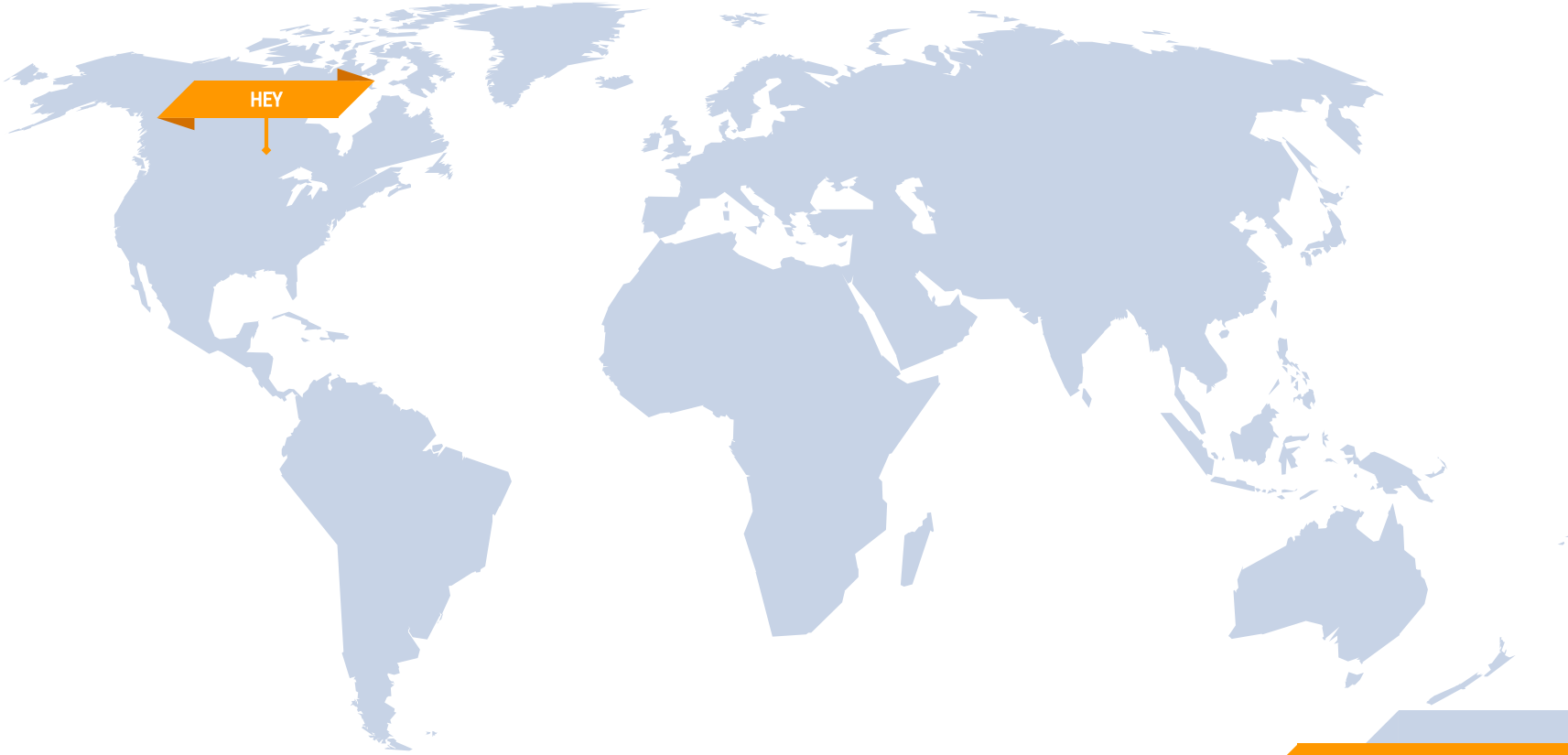

fragmentAbout.java
fragmentAboutAbout.java
fragmentAboutAboutAbout.java
fragmentAboutAboutAboutAbout.java
MainActivity.java
mListAdapter.java
NavigationDrawerFragment.java

# 4

> *I've wanted to be able to run **Smalltalk** on Android and iOS for many years. I finally decided to do something about it.*

# 5

# FORTRAN



# OUTLOOK

Security Investigation

Automatization

# SUMMARY

## Structure

no consistent implementation of best practices

## Build Tools

many projects lack proper use of build tools

## Languages

diverse use of languages complicates application maintenance