

# 约定

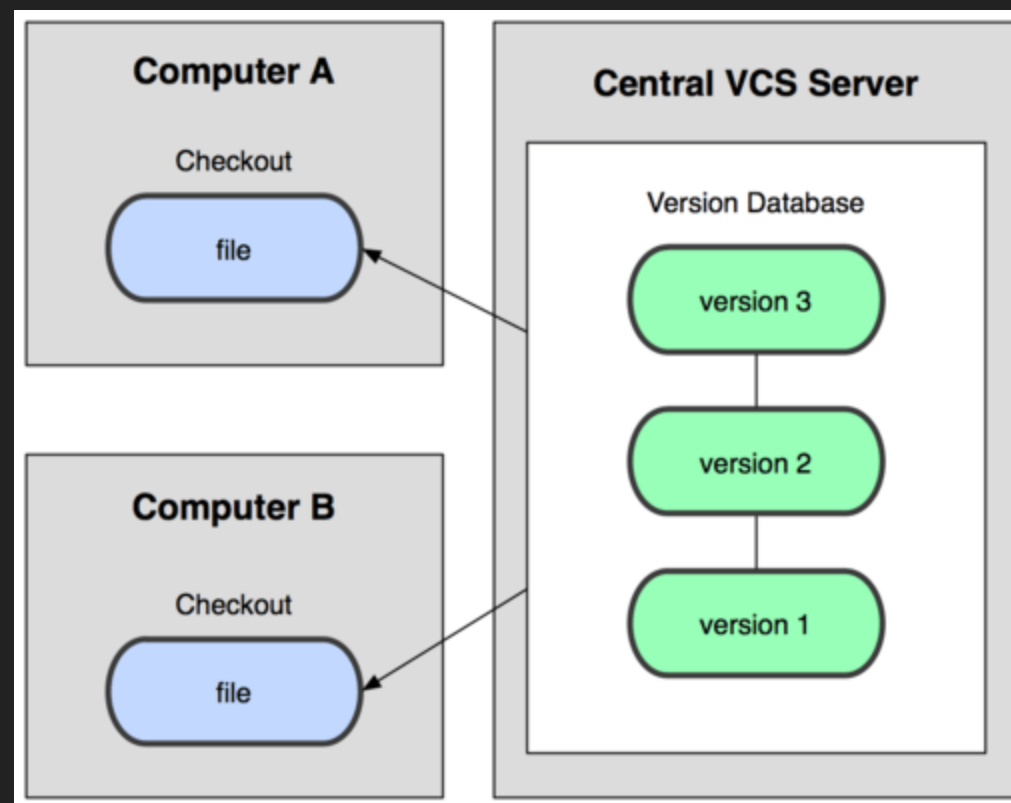
- 只有在练习时可以使用电脑，其他时间不需要
- 不需要记笔记，资料随后会分享出来
- 手机静音
- 请随时提问
- 语速太快、慢、大、小，请随时提醒
- 请大家遵守课堂纪律

# BASIC TOOLS

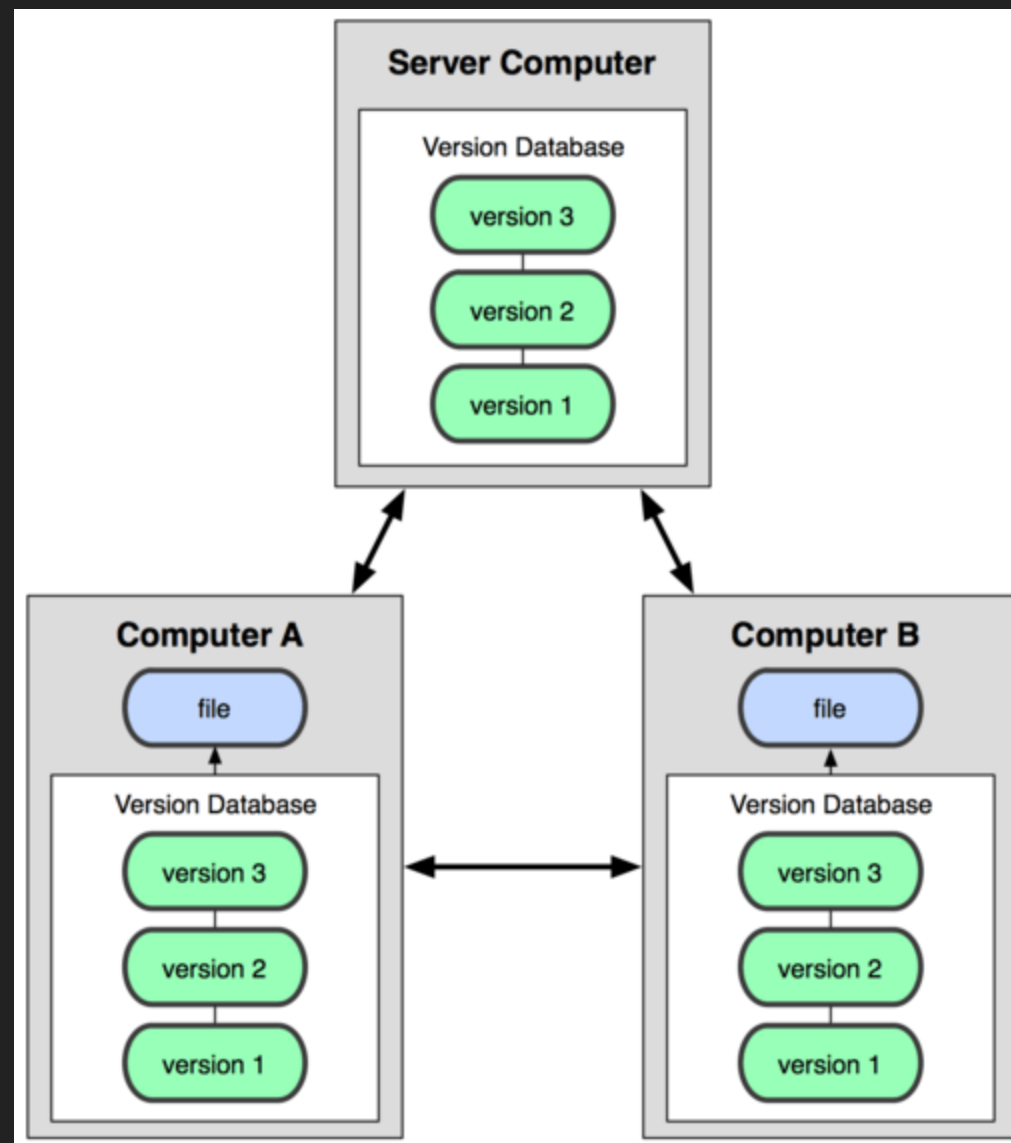
# GIT

- 一个分布式版本控制系统
- 最初由Linus Torvalds在2005年设计开发

- Centralized Version Control Systems



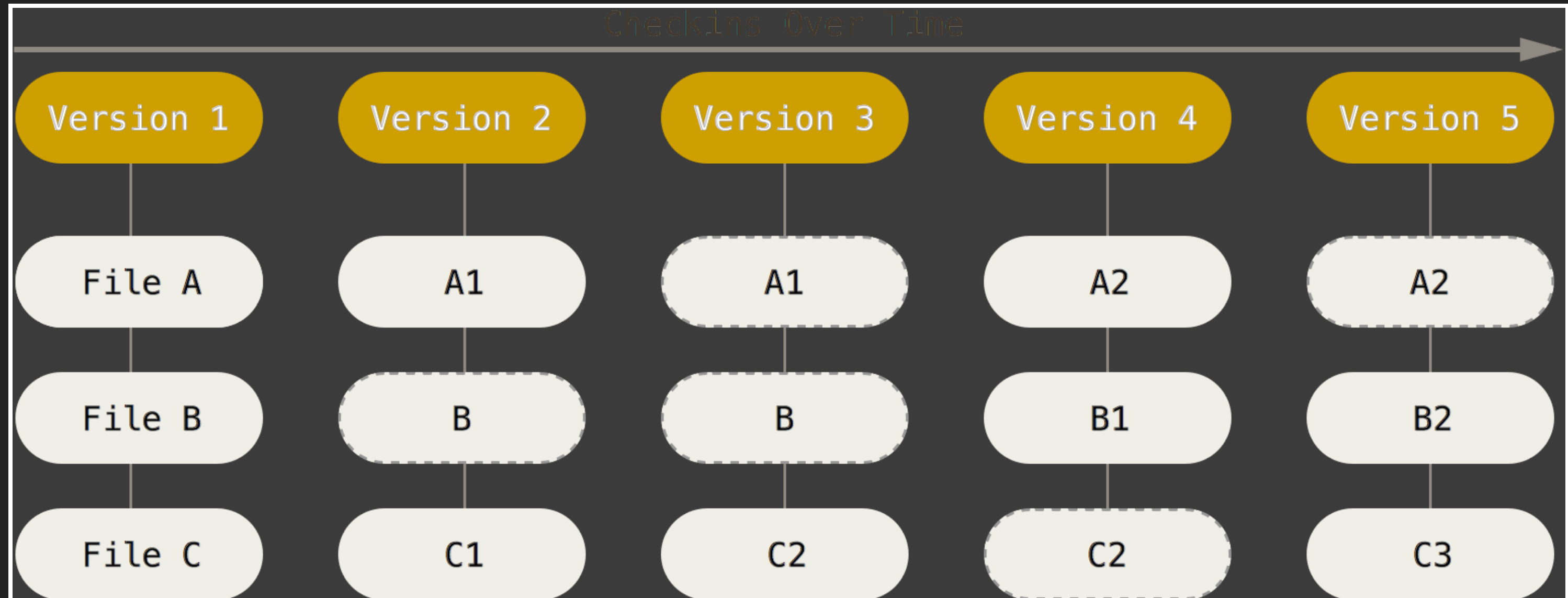
- Distributed Version Control Systems



## WHY GIT

- 几乎所有操作都在本地
- 快速
- 小步提交的绝配
- 备份

- 直接记录快照，而非差异比较



- 轻量级的分支

# INSTALL

<http://msysgit.github.io/>



## BASICS

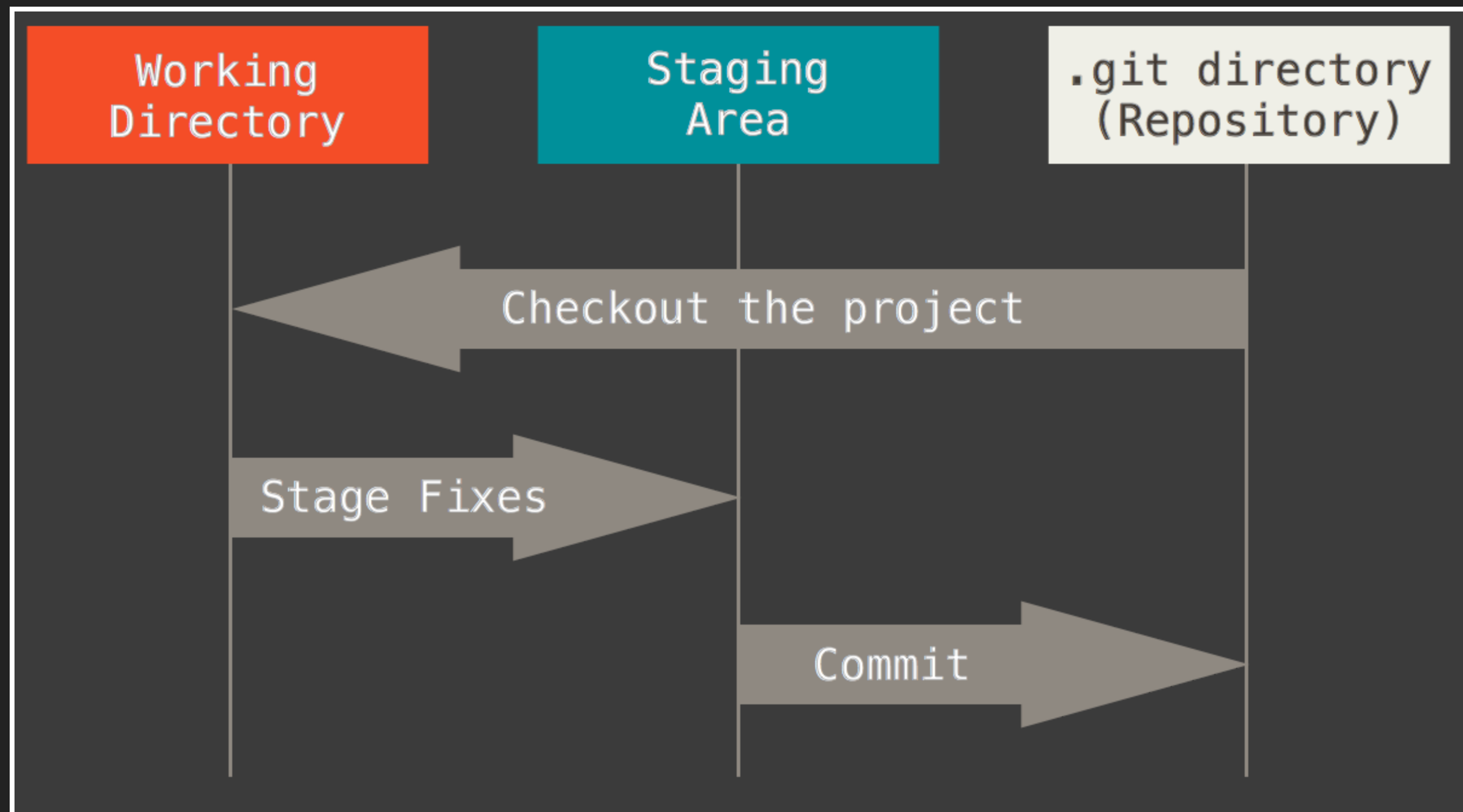
- 获取 Git 仓库
  - `$ git init`
  - `$ git clone`

- 跟踪新文件
  - `$ git add`

- 检查当前文件状态
  - `$ git status`

- 暂存已修改文件
  - `$ git add`

- 三种状态
  - 已提交 (committed)
  - 已修改 (modified)
  - 已暂存 (staged)



- 提交更新
  - `$ git commit`

- 查看提交历史
  - `$ git log`
  - `$ git log --oneline --decorate --color --graph`

## EXERCISE GIT BASICS

- Fork <http://code.huawei.com/qixi/prs-capability.git>
- `$ git clone`
- `$ git add`
- `$ git status`
- `$ git commit`
- `$ git log`



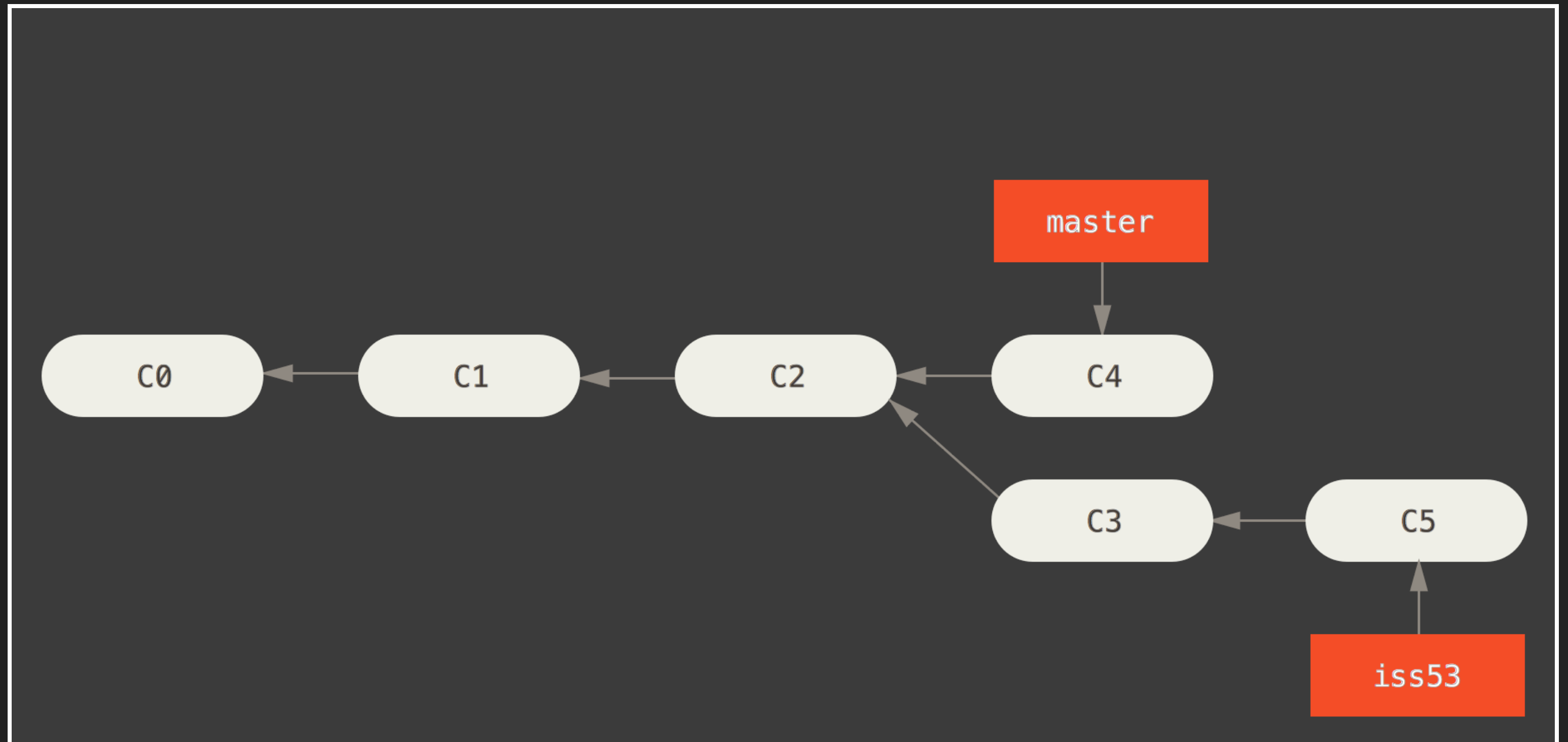
## CONFIG

- `$ git config --global user.name "Qi Xi"`
- `$ git config --global user.email xqi@thoughtworks.com`
- `$ git config --list`

## REMOTE

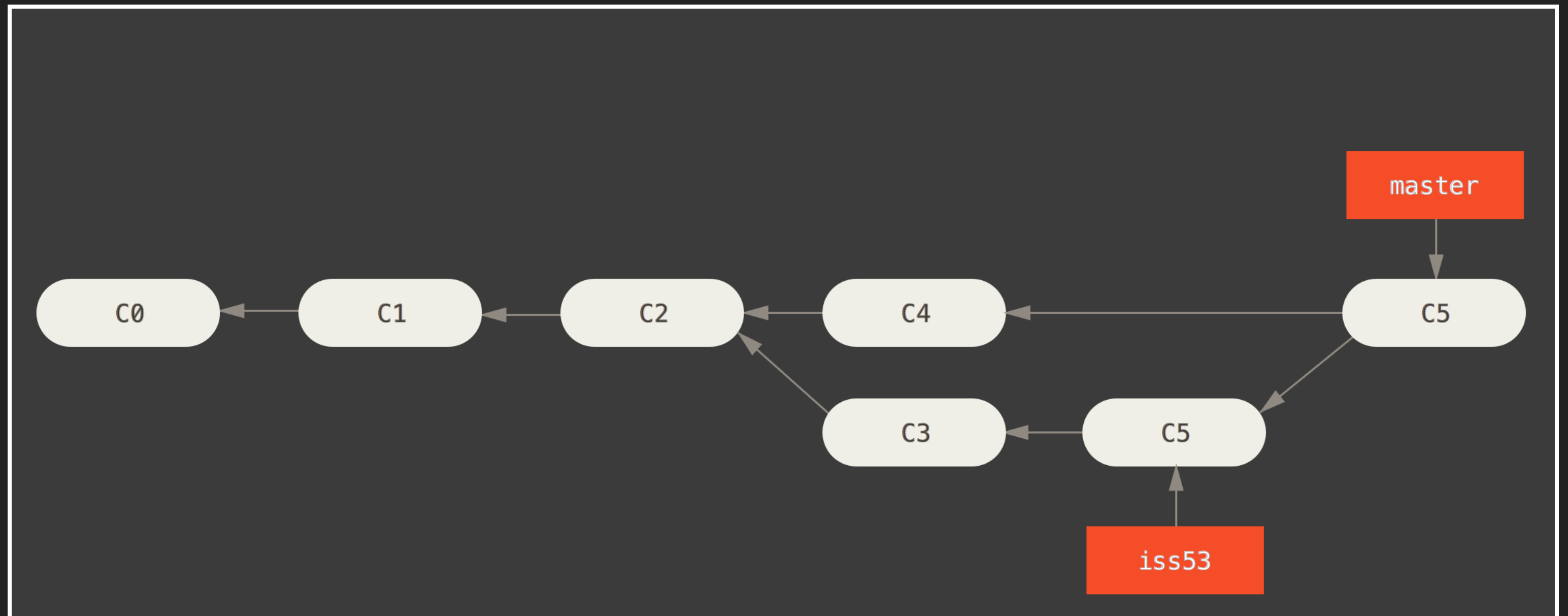
- `$ git remote -v`
- `$ git fetch`
- `$ git pull`
- `$ git push`

# BRANCH



- `$ git branch iss53`
- `$ git checkout iss53`
- `$ git branch`

# MERGE



- `$ git checkout master`
- `$ git merge iss53`

- `$ git branch -d iss53`

## GIT-SVN


- `$ git svn clone http://svn.example.com/project/trunk`
- `$ git svn rebase`
- `$ git svn dcommit`










# GITLAB


一个利用 Ruby on Rails 开发的开源应用程序，实现一个自托管的Git项目仓库，可通过Web界面进行访问公开的或者私人项目。

# CODE CLUB


 CodeClub / GitLab


Search in this project









You won't be able to pull or push project code via SSH until you [add an SSH key](#) to your profile. [Don't show again](#) | [Remind later](#)


 Project management and code hosting application. <http://code.huawei.com/> – [★ Star](#) 18 [Fork](#) 10


SSH HTTP `git@code.huawei.com:codeclub/gitlab.git` public 


Activity Readme Changelog Contribution guide License


17,260 commits 30 branches 194 tags 123.48 MB

 Push events  Merge events  Comments  Team

 duquansong 00119395 commented on issue #176 at CodeClub / GitLab less than a minute ago  
突然觉得这个貌似啤酒杯 咖啡杯有漂亮的没有？

 caochun KF74316 commented on issue #176 at CodeClub / GitLab 9 minutes ago

 Forked from: huzhiyong 00131288 / GitLab

 Compare code

## MERGE/PULL REQUESTS

- fork
- clone
- commit
- push
- merge request

**EXERCISE MERGE REQUESTS**

# IDE

## INTELLIJ IDEA

- <https://www.jetbrains.com/idea/>
- `bin/idea.exe.vmoptions -Xmx1024m`

# MAVEN

- 2002年发布 (Ant in 2000)
- 约定 (Ant write all the commands)
- XML

- pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.huawei</groupId>
  <artifactId>basic-tools</artifactId>
  <version>1.0.0</version>
  <packaging>war</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>3.2.4.RELEASE</version>
```

# GRADLE

- 2007年发布
- DSL Groovy
- Android



- build.gradle

```
apply plugin: 'java'

compileJava.options.encoding = 'UTF-8'
compileTestJava.options.encoding = 'UTF-8'
```

- build.gradle

```
apply plugin: 'jetty'
apply plugin: 'findbugs'
apply plugin: 'jacoco'

compileJava.options.encoding = 'UTF-8'
compileTestJava.options.encoding = 'UTF-8'

dependencies {
    compile "org.springframework:spring-web:3.2.4.RELEASE"
}

findbugs {
    ignoreFailures = true
    excludeFilter = file("myExcludeFilter.xml")
}
```

# SBT

- Simple Build Tool
- 增量编译
- 拥有交互shell
- Scala
- DSL

## build.sbt

```
lazy val root = (project in file(".")).
  settings(
    name := "hello",
    version := "1.0",
    scalaVersion := "2.11.4"
  )
```

## 目录结构

```
src/  
  main/  
    resources/  
      <files to include in main jar here>  
    scala/  
      <main Scala sources>  
    java/  
      <main Java sources>  
  test/  
    resources  
      <files to include in test jar here>  
    scala/  
      <test Scala sources>  
    java/  
      <test Java sources>
```

src/main/scala/Hi.scala

```
object Hi {  
  def main(args: Array[String]) = println("Hi!")  
}
```

```
$ sbt run
```

## EXERCISE SBT AND INTELLIJ IDEA

- build.sbt
- Hi.scala
- IntelliJ IDEA import



# 课后练习

<https://github.com/numbbbbb/progit-zh-pdf-epub-mobi>

阅读前3章，下次培训时为大家讲解rebase操作。