

# Git 是什么 (Basic)

## 1. Git 是什么？为什么需要它？

Git 是一个分布式版本控制系统 (Distributed Version Control System, DVCS)。

“版本控制” = 记录文件的变化历史，以便在需要时恢复到之前的状态。

## 2. Git 的工作流程



本地工作目录 (Working Directory)	暂存区 / 索引 (Staging/Index)	本地仓库 (Repository/.git)
存放正在开发的文件	存放准备提交的文件	存放提交后的文件
通过 <code>git add</code> 将文件放入暂存区	通过 <code>git add</code> 将文件放入暂存区	通过 <code>git commit</code> 将暂存区的文件提交到仓库
通过 <code>git checkout</code> 从仓库恢复文件到工作目录		

## 3. Git 的底层数据结构

Git 的底层数据结构由 4 部分组成：

对象类型	描述	示例
Blob	存储文件内容 (binary large object)	hello.java
Tree	存储目录结构 (指向 Blob 对象的集合)	项目根目录
Commit	存储提交信息 (指向 Tree 对象)	"v1.0.0"
Tag	存储版本标签 (指向 Commit 对象)	v1.0.0

tree 160000, tree blob 100644 .

```
[commit]
|
v
[tree]
├─ [blob] hello.txt
└─ [blob] app.js
```

## 4. HEAD??

HEAD → 160000 main or 100644

- HEAD → main
- git checkout 160000 HEAD 160000
- git reset HEAD 160000

## 5. ?????

160000 160000 160000 160000 .

```
A---B---C ← main (160000)
      ↑
    HEAD
```

- 160000 160000 → 160000 160000
- 160000 → 160000 160000 160000 160000 160000 160000

## 6. Reflog??

git reflog

→ HEAD 160000 160000 160000

- reset checkout 160000 160000
- 160000 Git HEAD 160000 160000

# 7. .git ????? ?? ??

```
.git/
├─ objects/      ← 00, 00, 00, 00 00 000
├─ refs/         ← 0000, 00 00 000
├─ HEAD          ← 00 0000 000
├─ index         ← 0000 00
├─ logs/         ← reflog 00 00
└─ config        ← Git 00
```

## ? ??

??	??
Commit	00 000 + 00 00 00
Blob	00 00 00
Tree	00 00
Branch	000 0000 000
HEAD	00 000 or 00
Staging Area	00 000 0000 00
Reflog	HEAD 00 00 00 00 (000 00 )