

Branch (การแตกกิ่ง)

Ref:

- ❑ <https://blog.pjjop.org/version-control-and-git-from-zero-to-hero-part-6/>
- ❑ <https://medium.com/amiearth/มาทำความรู้จัก-git-branch-560c23e67eb6>
- ❑ <https://github.com/opendream/progit/blob/master/03-git-branching/01-chapter3.markdown>
- ❑ <https://www.somkiat.cc/git-merge/>

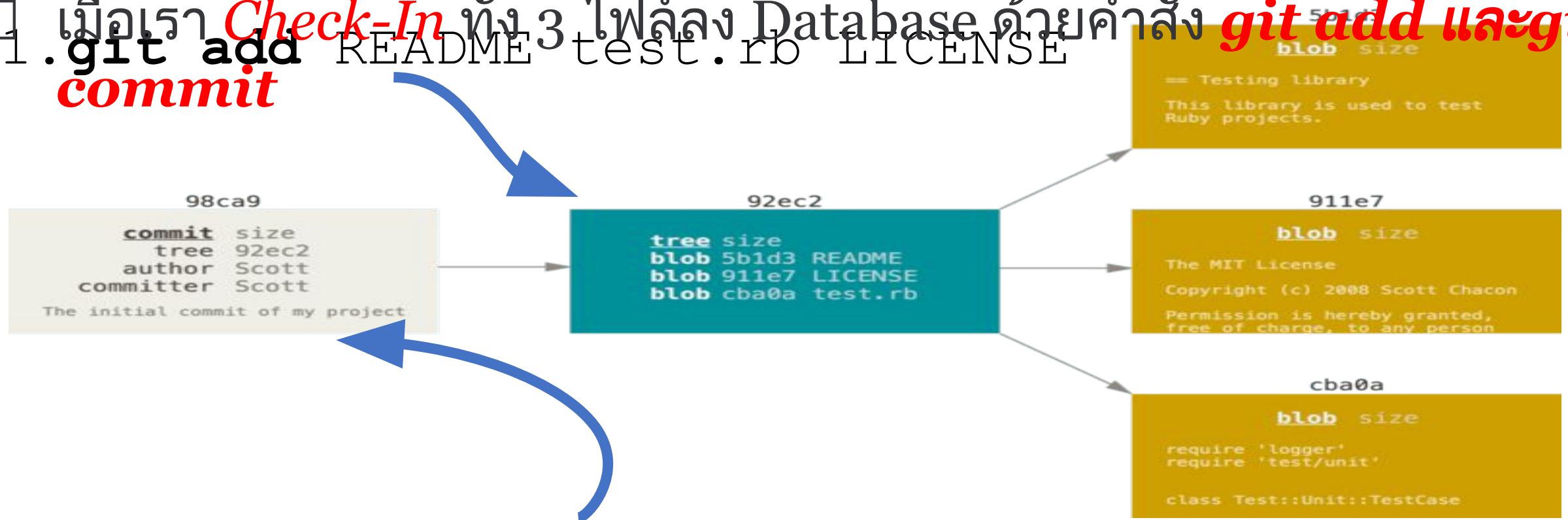
Version Control System

- Version Control System เกือบทุกตัวจะมี **Branch Feature** ให้ใช้งานในรูปแบบใด
- การแตก Branch ใน Version Control System เหล่านั้นค่อนข้าง**มีค่าใช้จ่ายที่สูง** เหมือนกับการทำสำเนา Source Code ทั้ง Folder ยิง Project มีขนาดใหญ่ยิ่งใช้เวลาในการสร้าง Branch หลายวินาที หรือเป็นนาที
- แต่ Branch ใน Git เป็นอะไรที่แตกต่างกัน เพราะ**ทำงานได้รวดเร็วและใช้ทรัพยากรน้อยกว่า**

Check-In (เก็บไฟล์ใน Local repository)

□ สมมติเรามีไฟล์อยู่ 3 ไฟล์ คือ README, test.rb และ LICENSE ใน Git Folder

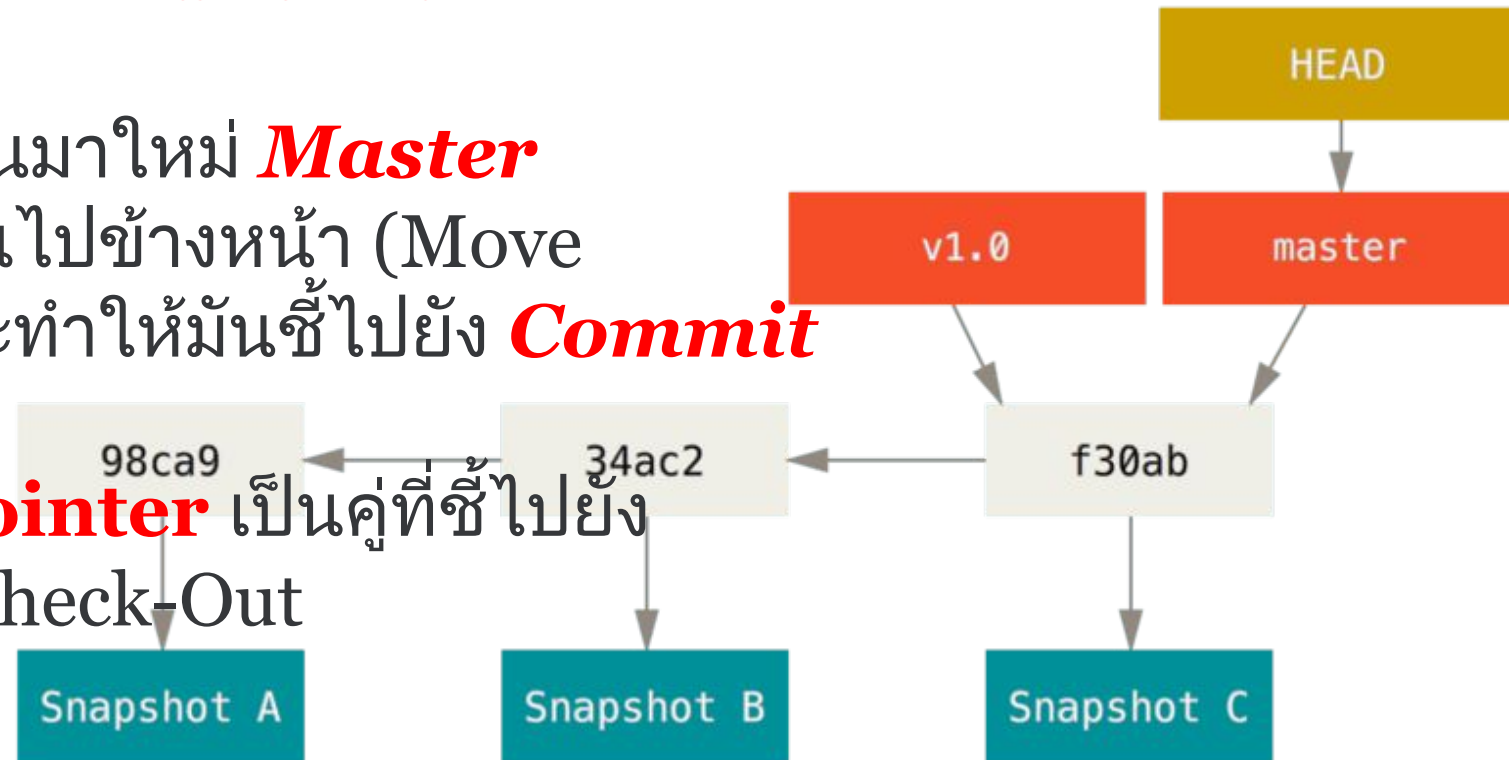
□ เมื่อเรา *Check-In* ทั้ง 3 ไฟล์ลง Database ด้วยคำสั่ง *git add* และ *git commit*



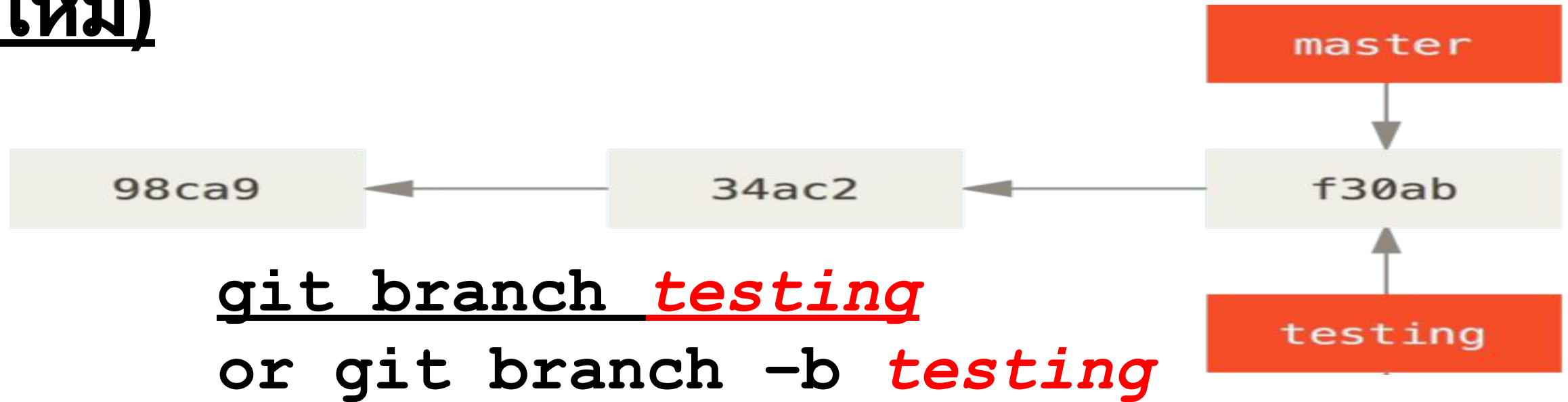
/2. `git commit -m 'The initial commit of my project'`

Companion (คู่หู Head+Master)

1. Branch คือ Pointer ที่สามารถเคลื่อนที่ไปยัง Commit บนสุดของ Timeline โดย Git จะสร้าง Branch ชื่อ **Master เป็น Branch Default เมื่อมีการ Commit ในครั้งแรก**
2. ทุกครั้งที่เราสร้าง Commit ขึ้นมาใหม่ **Master Branch Pointer** จะเคลื่อนไปข้างหน้า (Move Forward) โดยอัตโนมัติ ซึ่งจะทำให้มันชี้ไปยัง **Commit ล่าสุดของ Timeline** เสมอ
3. นอกจากนี้มันยังมี **HEAD Pointer** เป็นคู่หูที่ชี้ไปยัง Master Branch ที่เรากำลัง Check-Out



Creating a New Branch (การสร้างกิ่งใหม่)



สร้าง Branch ใหม่ชื่อ Testing

- ✓ Git จะสร้าง ***Testing Pointer*** ที่ชี้ไปยัง ***Commit***
- ✓ เดิมที Master Pointer ที่ชี้ไปยัง Commit

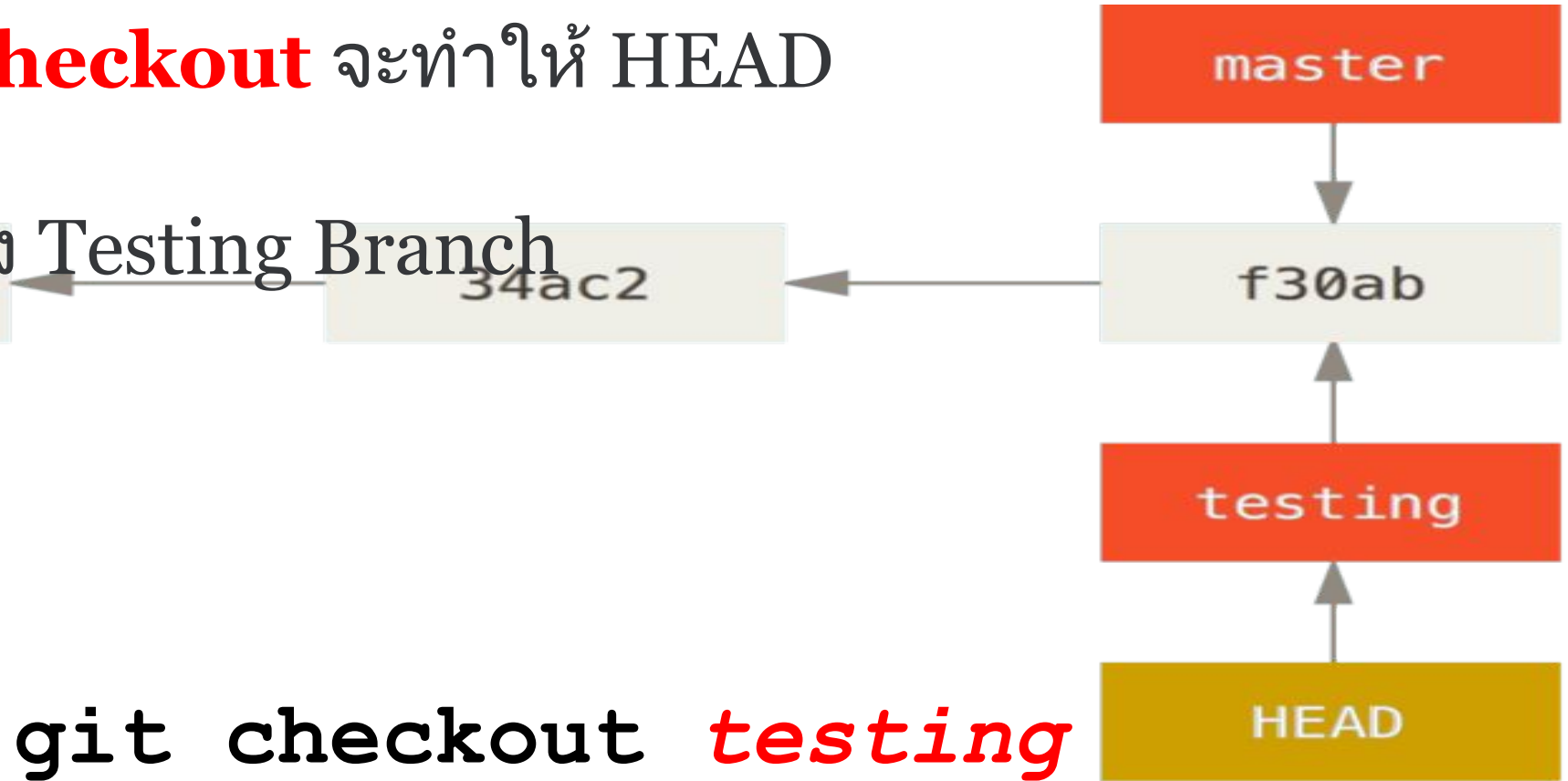
Head Pointer (ตัวชี้ตำแหน่งหัวของกิ่ง)

❖ **Branch ปัจจุบัน** ยังคงเป็น Master Branch
เพราะ **HEAD Pointer** ยังคงชี้ที่ **Master**



Switching Branches (เปลี่ยนตำแหน่งหัวของกิ่ง)

- ❖ คำสั่ง **git checkout** จะทำให้ HEAD pointer เคลื่อนตัวไปยัง Testing Branch



Edited Code (แก้ไขคำสั่ง)

```
vi test.rb # แก้ไขคำสั่งในไฟล์นี้  
git commit -a -m 'made a change'
```

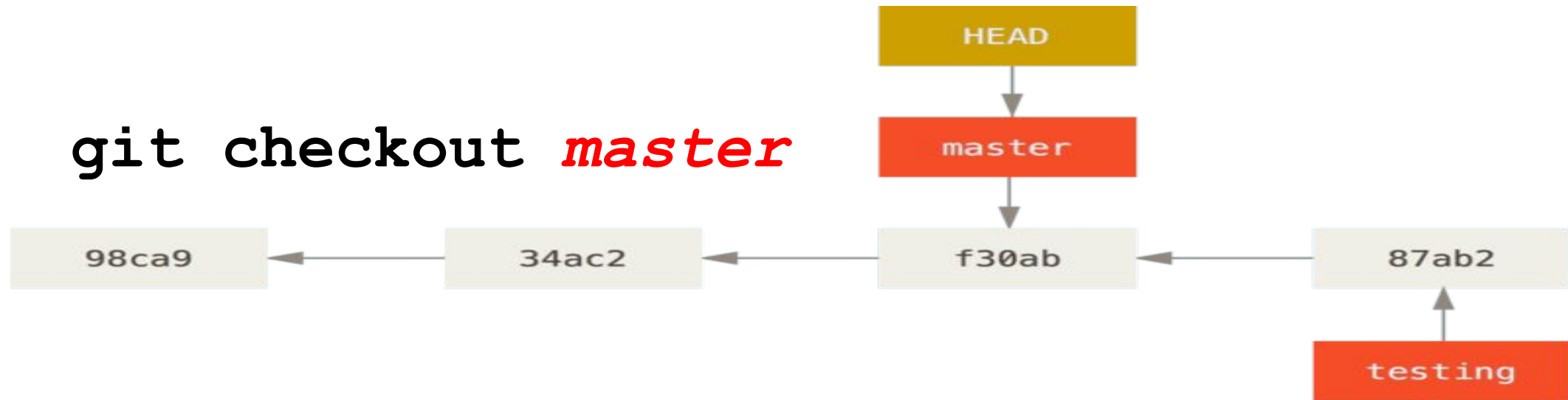


แก้ไข Code ในไฟล์ test.rb แล้ว **Check-In**

- ❖ สิ่งที่เกิดขึ้นก็คือ Testing Pointer และ **HEAD Pointer** จะเคลื่อนตัวไปข้างหน้า Commit ใหม่ (**87ab2**)
- ❖ ขณะที่ **Master Pointer** ยังอยู่ที่เดิม (**f30ab**)

Switching to Master Branch

`git checkout master`



การสลับไปมาระหว่าง Branch มายัง Master Branch

- ❖ โดยใช้คำสั่ง `git checkout` ซึ่งจะทำให้ **HEAD Pointer** เคลื่อนตัวกลับมายัง **Master Pointer**

New Timeline (สายงานใหม่)

แก้ไข Code ในไฟล์ test.rb แล้ว Check-In ที่ Master Branch

- ❖ **HEAD Pointer** และ **Master Pointer** จะเคลื่อนตัวไปข้างหน้ายัง **Commit** ล่าสุด (**c2b9e**)
- ❖ ซึ่งจะทำให้เกิดการแยกตัวออกเป็น 2 Timeline

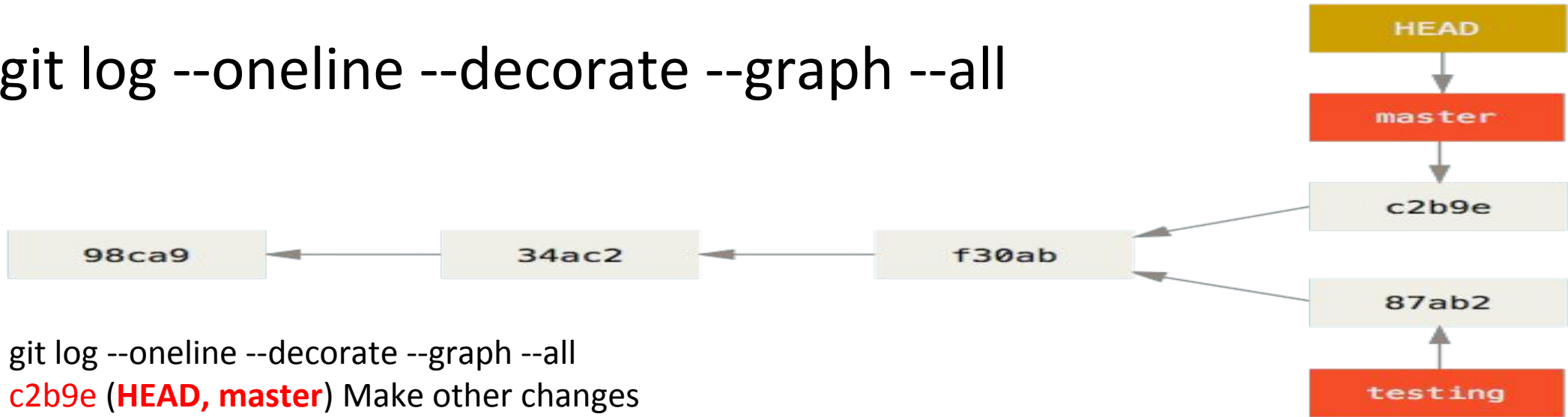


Source Code มี 2 Timeline การ Merge Branch

- ❖ สิ่งสำคัญเมื่อมีการทำงานร่วมกันเป็นทีมเพื่อส่ง Source Code ที่แก้ไขไปยัง Branch หลัก

Graph (แสดงข้อมูลกิ่ง)

git log --oneline --decorate --graph --all



```
$ git log --oneline --decorate --graph --all
```

```
* c2b9e (HEAD, master) Make other changes
```

```
| * 87ab2 (testing) Make a change
```

```
|/
```

```
* f30ab Add feature #32 - ability to add new formats to the  
central interface
```

```
* 34ac2 Fix bug #1328 - stack overflow under certain conditions
```

```
* 98ca9 Initial commit of my project
```