

---

# 상황중심의 git

문민식

---



---

## INDEX

1. 왜 상황중심?
  2. 로컬
  3. 원격
-

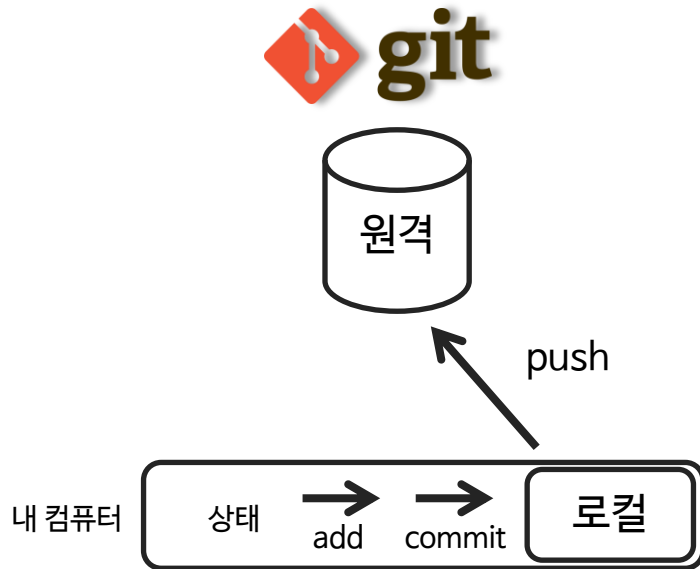
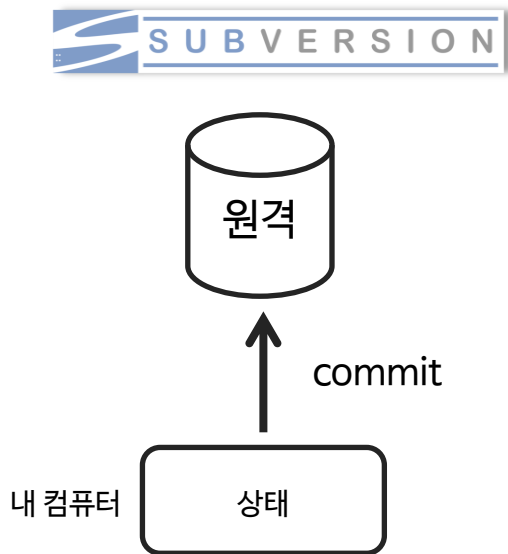
SVN 첫 사용 느낌

---

“편하네”

---

## SVN 첫 사용 느낌 - 왜?



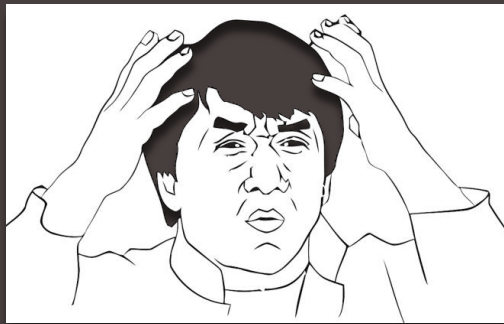
같은 행위에 명령어 단계가 줄어들어 편하다고 느낌.

# 상황

git이 필요해.

## 상황1 - 요구사항이 자주 변한다.

---



이관 직전 버전을 예측하기 힘들다.

---

## 상황2 - 이관 당 많은 요구사항&소스변경

---



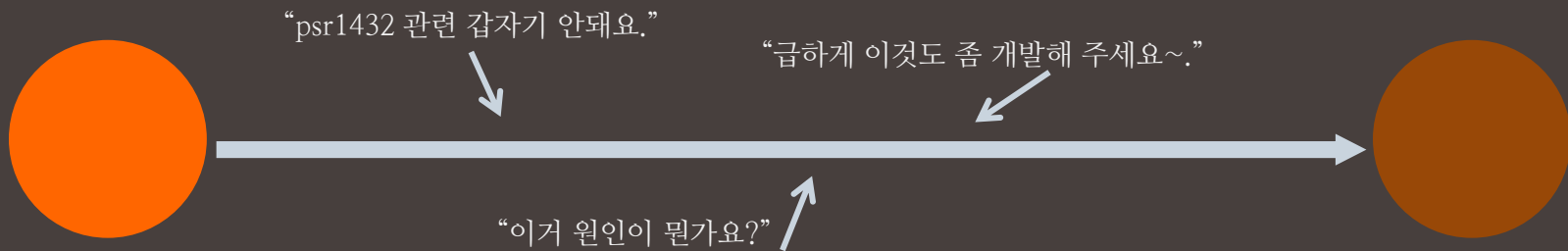
기능11번 관련해서 어떤  
파일들이 수정된거지?

기능1, 2, 3 ... 20 개발에 수정된 파일목록

---

### 상황3 - 예상에 없던 변경들이 생긴다.

---



“분명 뭔가 섞였는데...”

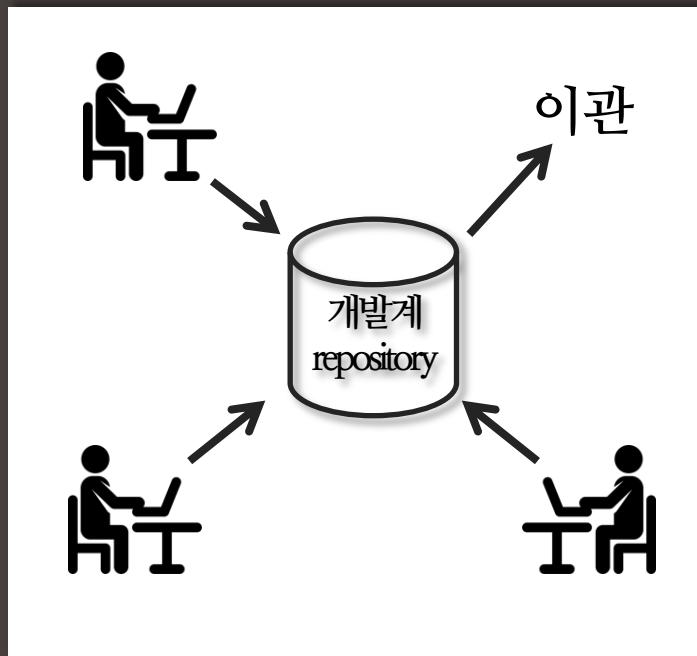
“뭐 고치고 있었더라?”

---



## 상황4 - 커밋을 자유롭게 할 수 없음.

---



상황5 - 하나의 기능도 여러 방법으로 개발가능.

---



상황 -> git으로 해결 -> 경험.

---



=



=



# 명령어

로컬

# 명령어 사용빈도

| command  | google results | %            |
|----------|----------------|--------------|
| commit   | 528,000        | 7.981980075  |
| push     | 523,000        | 7.906393143  |
| pull     | 506,000        | 7.649397572  |
| clone    | 489,000        | 7.392402002  |
| checkout | 470,000        | 7.105171658  |
| add      | 446,000        | 6.742354382  |
| branch   | 439,000        | 6.636532676  |
| log      | 388,000        | 5.865545964  |
| diff     | 369,000        | 5.578315621  |
| fetch    | 355,000        | 5.36667221   |
| merge    | 354,000        | 5.351554823  |
| init     | 343,000        | 5.185263572  |
| status   | 286,000        | 4.323572541  |
| reset    | 267,000        | 4.036342197  |
| tag      | 246,000        | 3.718877081  |
| rebase   | 203,000        | 3.068829461  |
| rm       | 142,000        | 2.146668884  |
| show     | 104,000        | 1.572208197  |
| bisect   | 62,800         | 0.9493718726 |
| grep     | 49,400         | 0.7467988934 |

## 준비물 : just-alert 시스템 (JAS)

---

1. alert 기능 밖에 없는 시스템
2. git 으로 요구사항에 대응해 나가자.

### JUST ALERT SYSTEM

start

이 페이지 내용:

master

확인

# git clone

---

“리모트에서 프로젝트 가져가세요.”

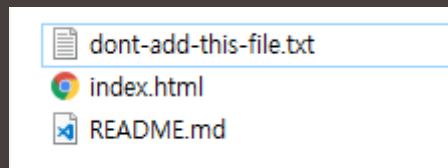
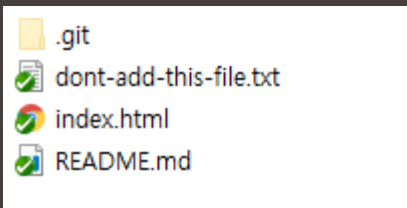
<https://github.com/MinSikMoon/just-alert.git>

```
public@minsik MINGW64 /d/GIT  
$ git clone https://github.com/MinSikMoon/just-alert.git|
```

# .git 지워보기

---

“이 프로젝트의 git관련된 기록들 다 지우고 싶다.”





# git init & git status

---

“git 버전관리 시작해보기.”

```
public@minsik MINGW64 /d/GIT/just-alert
$ git init
Initialized empty Git repository in D:/GIT/just-alert/.git/

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        README.md
        dont-add-this-file.txt
        index.html
```

# git add & commit

---

“don't-add-this-file.txt 빼고 커밋해주세요.”

```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git add index.html README.md
```

```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git status
On branch master
```

No commits yet

Changes to be committed:  
(use "git rm --cached <file>..." to unstage)

```
new file:   README.md
new file:   index.html
```

Untracked files:  
(use "git add <file>..." to include in what will be committed)

```
dont-add-this-file.txt
```

```
public@minsik MINGW64 /d/GIT/just-alert (master)
```

```
$ git commit -m "첫 커밋입니다."
[master (root-commit) 25be102] 첫 커밋입니다.
2 files changed, 43 insertions(+)
create mode 100644 README.md
create mode 100644 index.html
```

```
public@minsik MINGW64 /d/GIT/just-alert (master)
```

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
```

```
dont-add-this-file.txt
```

nothing added to commit but untracked files present (use "git add" to track)

# git rm --cached

---

“저 파일은 add하면 안되는구나. 빼야지”

```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git add dont-add-this-file.txt

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   dont-add-this-file.txt

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git rm --cached dont-add-this-file.txt
rm 'dont-add-this-file.txt'

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        dont-add-this-file.txt
```

---

# .gitignore

---

“don't-add-this-file.txt 귀찮아.  
신경도 안쓰고 싶은데...”

```
❖ .gitignore x
1 dont-add-this-file.txt
2 .gitignore|
```

```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git status
On branch master
nothing to commit, working tree clean
```

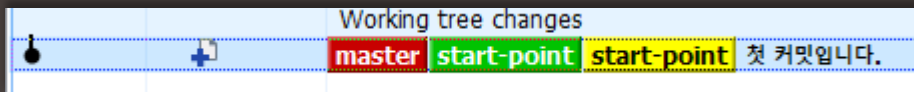
# git branch & tag

---

“언제든지 돌아갈 수 있는 시작점 만들자.”

```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git branch start-point

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git tag start-point
```



# 현업의 요구사항 발생

---

## 요구사항 정의서 1

- 기능 1 : “1번 알람” alert 생성
  - 기능 2 : “2번 알람” alert 생성
  - 기능 3 : “3번 알람” alert 생성
  
  - 위 3개 alert는 순서대로 나오게
-

# git branch

“요구사항 별로 branch 만들어보자”



```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git branch start-point

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git tag start-point

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git branch psr1/dev1

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git branch psr1/dev2

public@minsik MINGW64 /d/GIT/just-alert (master)
$ git branch psr1/dev3
```

| Working tree changes  |   |        |           |           |           |                                  |
|---|---|--------|-----------|-----------|-----------|----------------------------------|
|  |  | master | psr1/dev1 | psr1/dev2 | psr1/dev3 | start-point start-point 첫 커밋입니다. |

# git checkout

---

“첫번째 요구사항 개발하러 psr1/dev1 브랜치로 이동”

```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git checkout psr1/dev1
Switched to branch 'psr1/dev1'

public@minsik MINGW64 /d/GIT/just-alert (psr1/dev1)
$ █
```



# 각 브랜치에서 기능개발하고 commit하자.

```
<!-- Script 파트 -->
<script>
  function mainAlert() {
    func1();
  }

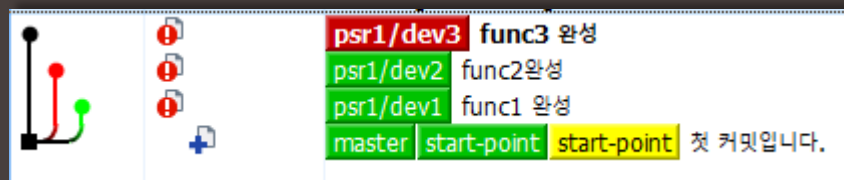
  function func1() {
    alert("1번 알람");
  }
</script>
```

```
<!-- Script 파트 -->
<script>
  function mainAlert() {
    func2();
  }

  function func2() {
    alert("2번 알람");
  }
</script>
```

```
<!-- Script 파트 -->
<script>
  function mainAlert() {
    func3();
  }

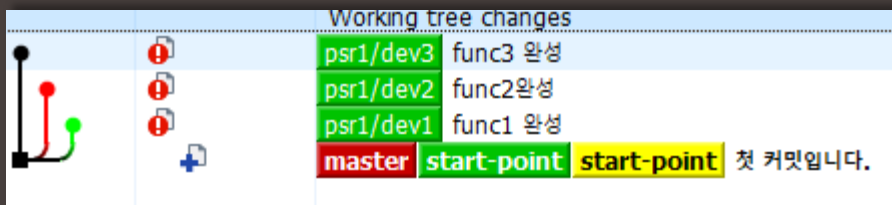
  function func3() {
    alert("3번 알람");
  }
</script>
```



## git checkout : 다시 처음으로 돌아오기

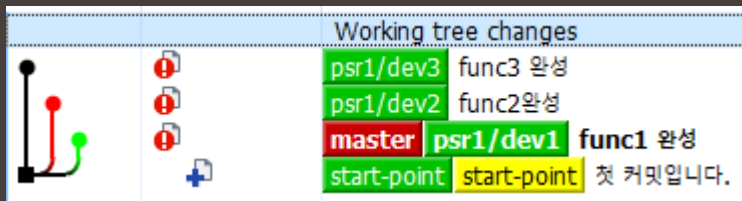
---

“이제 슬슬 이관 준비해볼까.”



## git merge: 1과 합치기 (fast-forward)

```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git merge psr1/dev1
Updating c44fd93..5838303
Fast-forward
 index.html | 6 +++++-
 1 file changed, 5 insertions(+), 1 deletion(-)
```



# git merge: 2와 합치기 (충돌해결 : 3way-merge)

```
14 </div>
15
16
17 <!-- Script 파트 -->
18 <script>
19     function mainAlert() {
20         <----- HEAD (Current Change)
21         func1();
22     }
23
24     function func1() {
25         alert("1번 알람");
26     }
27
28     function func2() {
29
30     }
31     function func2() {
32         alert("2번 알람");
33     }
34 }
35 </script>
```

Accept Current Change | Accept Incoming Change | Accept Both Changes | Cancel

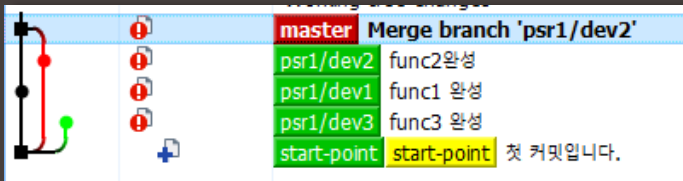
public@minsik MINGW64 /d/GIT/just-alert (master|MERGING)  
\$ git merge psr1/dev2  
Auto-merging index.html  
CONFLICT (content): Merge conflict in index.  
Automatic merge failed; fix conflicts and then commit the result.

```
function func1() {
    alert("1번 알람");
}

function func2() {
    alert("2번 알람");
}
```

```
public@minsik MINGW64 /d/GIT/just-alert (master|MERGING)
$ git add index.html

public@minsik MINGW64 /d/GIT/just-alert (master|MERGING)
$ git commit
[master 1affe1e] Merge branch 'psr1/dev2' func2와 머지
```

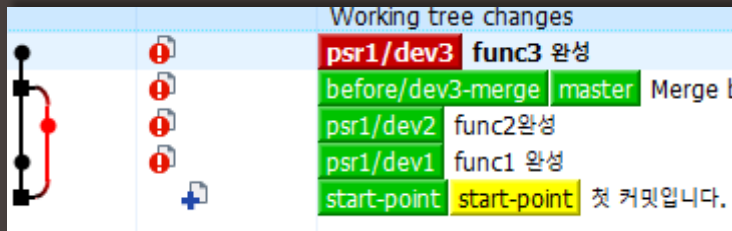


# git rebase & rebase --continue : 3과 합치기

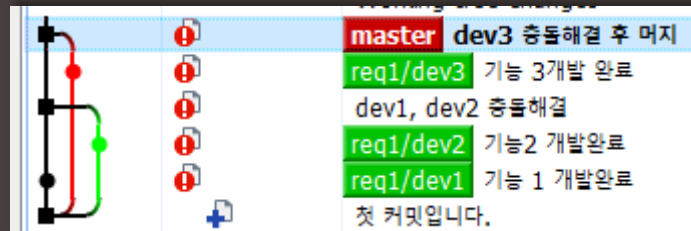
```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git checkout psr1/dev3
Switched to branch 'psr1/dev3'

public@minsik MINGW64 /d/GIT/just-alert (psr1/dev3)
$ git rebase master
```

rebase



merge



## 현업의 요구사항 변경 발생

---

“기능 2 빼주세요.”

---

# git cherry-pick

“1,3 골라서 붙이면 되네.”

```
.gitignore x index.html x
16
17 <!-- Script 파트 -->
18 <script>
19   function mainAlert() {
20     func1();
21   }
22   function func1() {
23     alert("func1!");
24   }
25   function func3() {
26     alert("func3!");
27   }
28   function func3() {
29     alert("func3!");
30   }
31   >>>>>> 01e81da... f3 (1)

Accept Current Change | Accept
16
17 <!-- Script 파트 -->
18 <script>
19   function mainAlert() {
20     func1();
21     func3();
22   }
23   function func1() {
24     alert("func1!");
25   }
26   function func3() {
27     alert("func3!");
28   }
29   function func3() {
30     alert("func3!");
31   }
32 </script>
33 </body>
```

```
public@minsik MINGW64 /d/GIT/just-alert (master)
$ git cherry-pick dev1 dev3
[master 8aef5b7] f1
Date: Thu Nov 22 23:11:56 2018 +0900
1 file changed, 5 insertions(+), 1 deletion(-)
error: could not apply 01e81da... f3
hint: after resolving the conflicts, mark the corrected paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'
```

