

Git diff

Version Control and
Git & Github basics

강환수 교수



AI Experts
Who Lead
The Future

01

파일이나 커밋 차이 git diff



- SA(staging area)에서 WD(Working directory) 사이의 차이 보임
 - Staging 된 파일상태와 현재 수정 중(작업 디렉토리)인 상태 비교
 - 특정 파일을 지정하면 파일에 대한 차이를 알려 줌
 - Working Directory와 Staging Area간의 비교도 가능하고
 - commit간의 비교, branch간의 비교도 가능

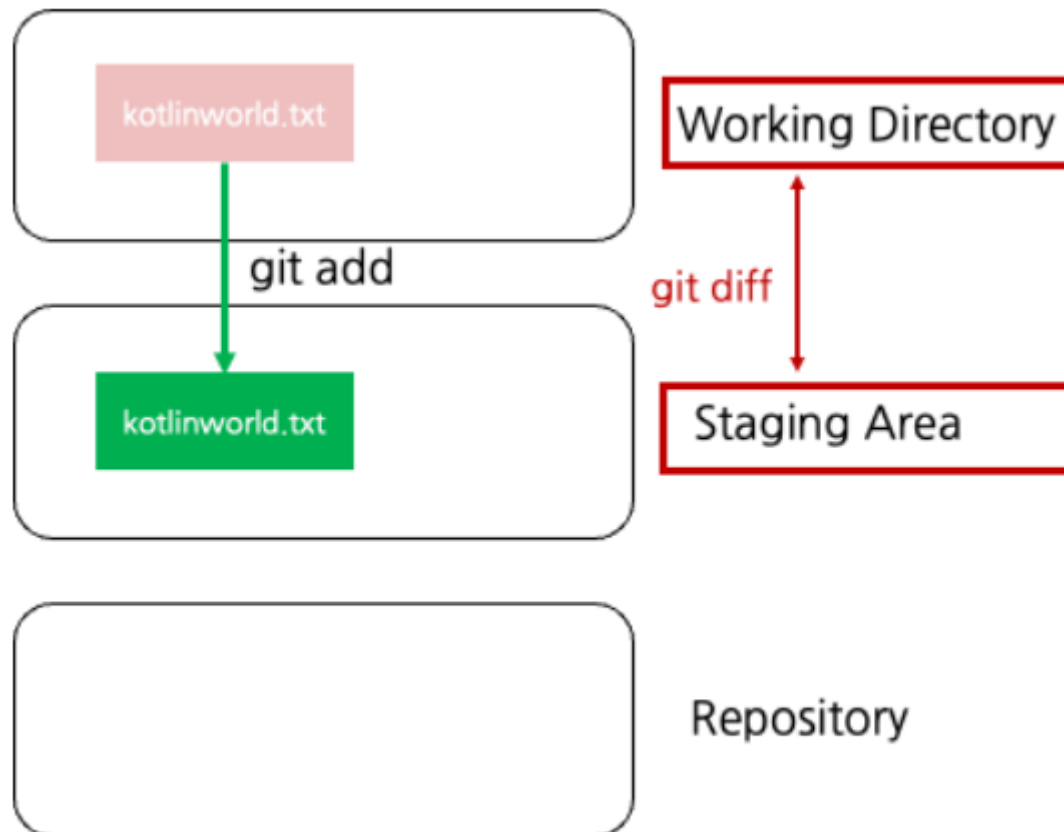
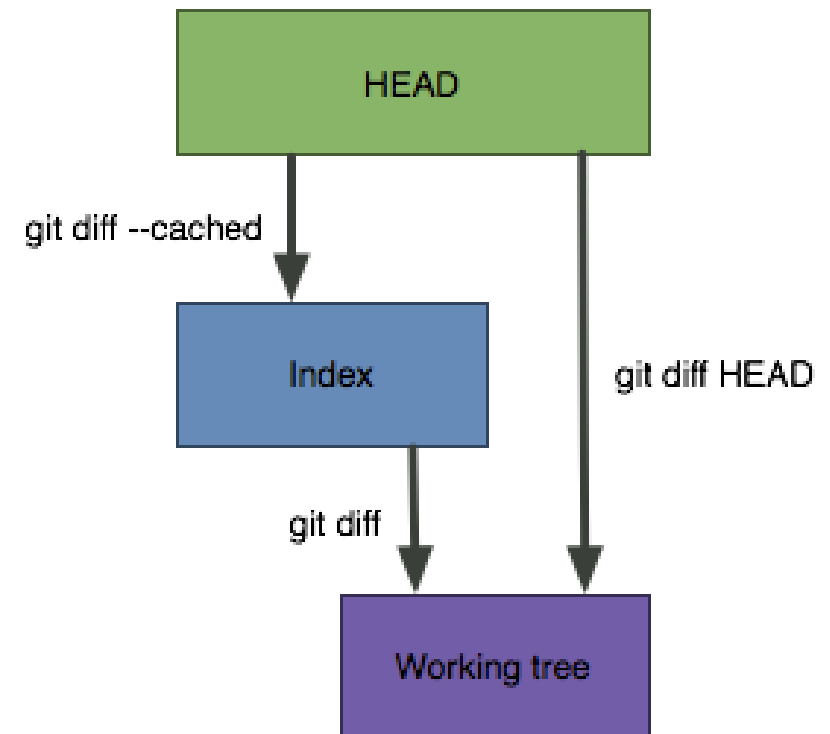
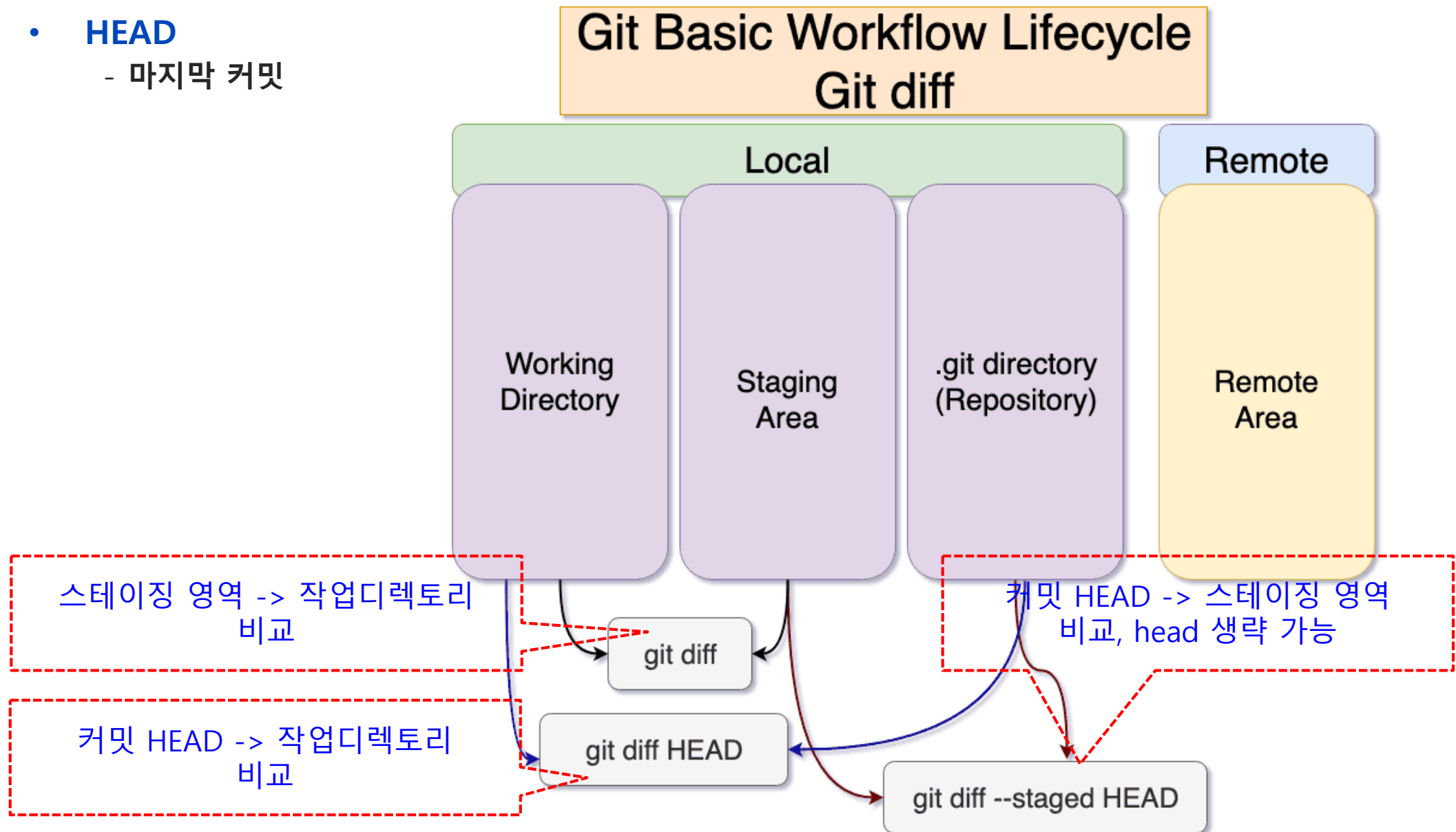


그림2. git diff

- **commit된 파일상태와 현재 수정중인 상태 비교**
 - \$ git diff
- **commit된 파일상태와 add된 파일 상태 비교**
 - \$ git diff --staged
 - \$ git diff --cached
- **commit간의 상태 비교하기 - commit hash 이용**
 - \$ git diff [비교할commit해쉬1] [비교할commit해쉬2]
 - git diff 048171 0c747d
- **commit간의 상태 비교하기 - HEAD 이용**
 - \$ git diff HEAD HEAD^
 - 가장 최근의 커밋과 그 전의 커밋을 비교한다
- **branch간의 상태 비교하기 - HEAD 이용**
 - git diff [비교할branch1] [비교할branch2]
 - ex) git diff feature/test origin/master
 - local의 feature/test브랜치와 remote의 master branch 비교



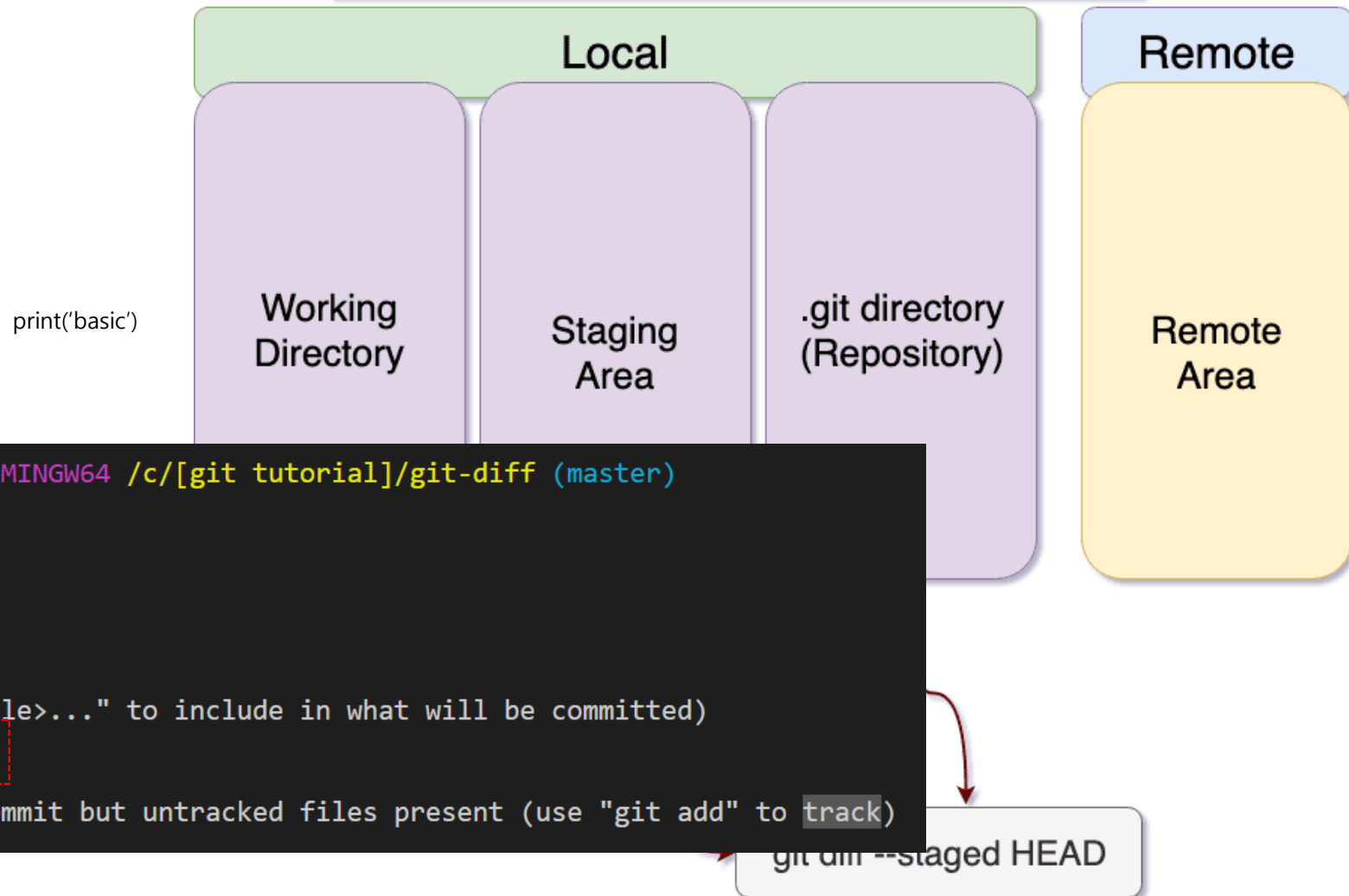
- **HEAD**
 - 마지막 커밋



- 저장소 git-diff 생성
 - \$ git init git-diff
- 파일 hello.py 생성
 - print('basic')
- \$ git add .
- 이후 파일 수정
 - print(list('python'))
- \$ git status
- \$ git diff

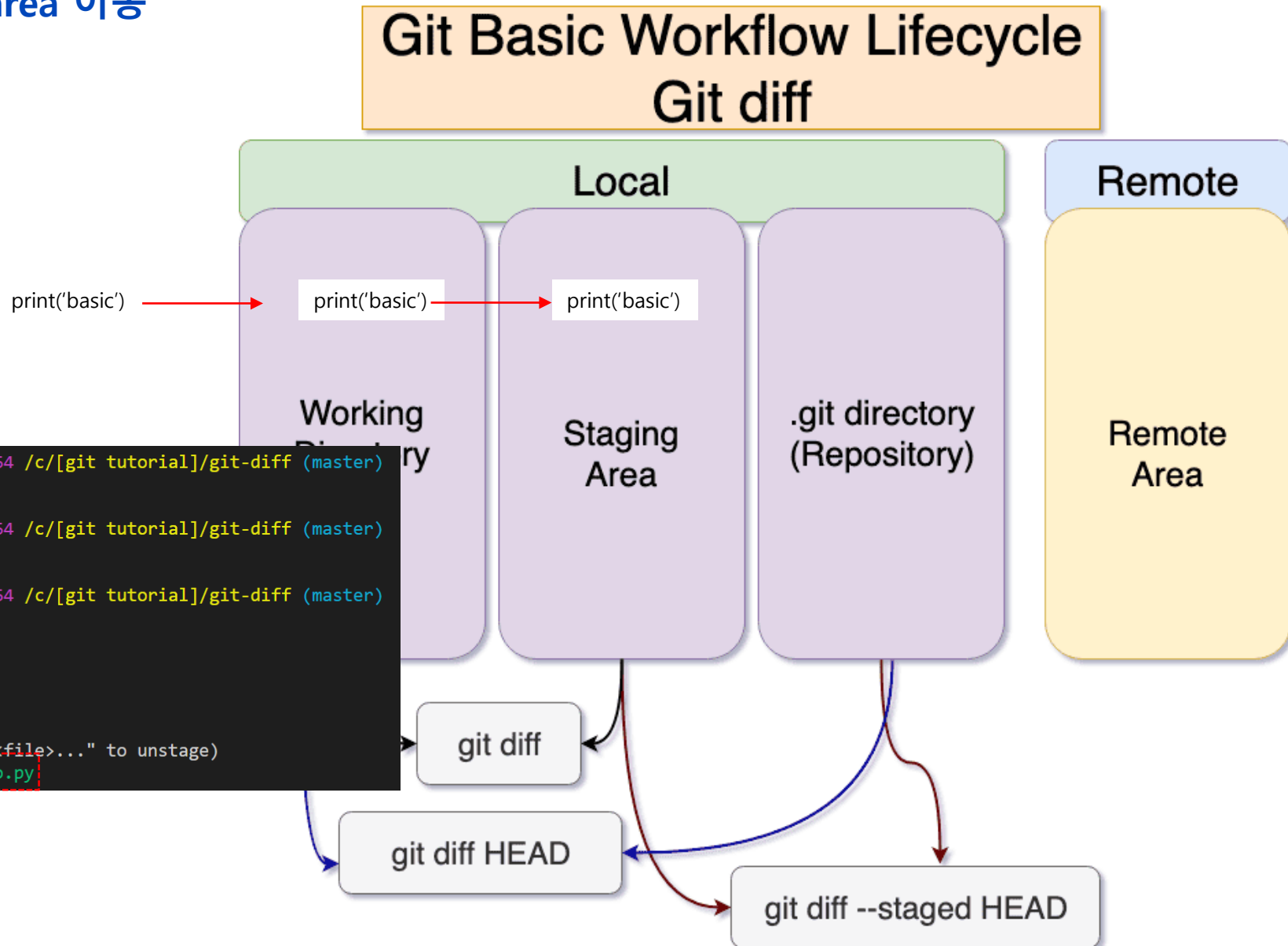
- 파일 `hello.py` 생성
 - `print('basic')`

Git Basic Workflow Lifecycle Git diff



\$ git add hello.py

- Staging area 이동
 - tracked



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git add hello.py

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git status
On branch master

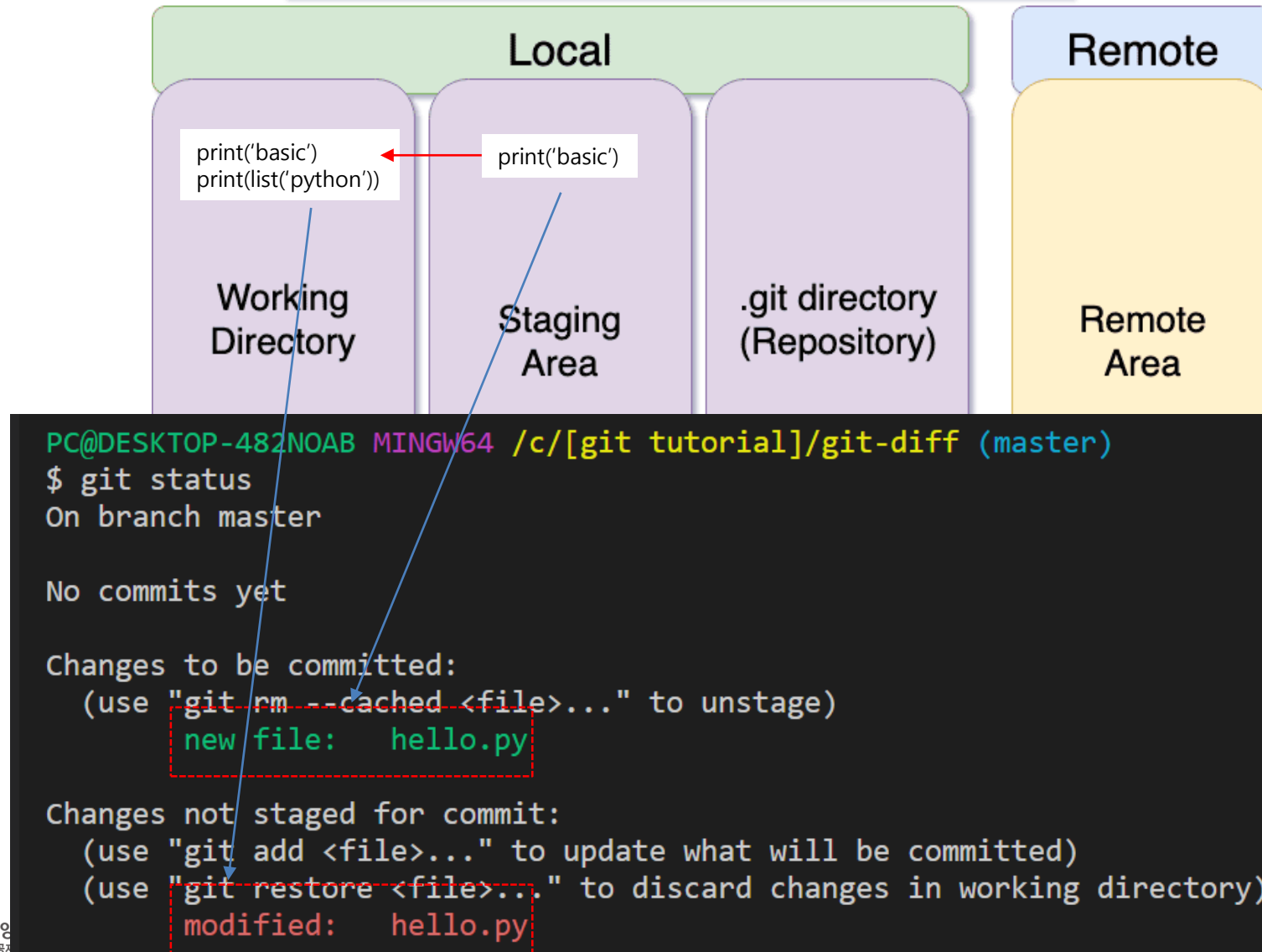
No commits yet

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

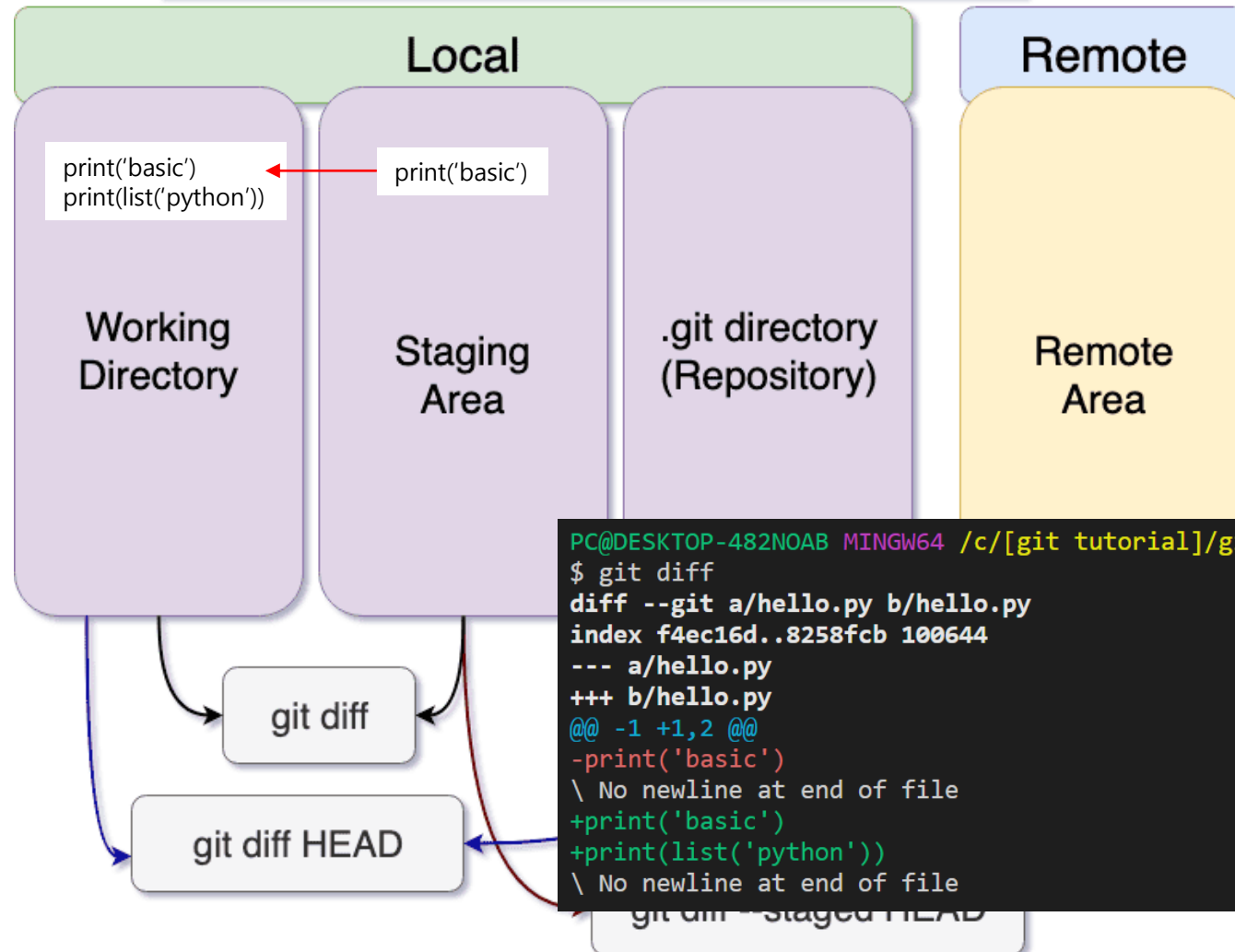

- `$ git rm --cached hello.py`
- 다시 복구
- `$ git add hello.py`

- hello.py
2번줄 추가

Git Basic Workflow Lifecycle Git diff



Git Basic Workflow Lifecycle Git diff



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff
diff --git a/hello.py b/hello.py
index f4ec16d..8258fcb 100644
--- a/hello.py
+++ b/hello.py
@@ -1,2 @@
-print('basic')
\ No newline at end of file
+print('basic')
+print(list('python'))
\ No newline at end of file
```

- 커밋 후

Git Basic Workflow Lifecycle

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git commit -m 'add file'
[master (root-commit) 9b0b279] add file
1 file changed, 1 insertion(+)
create mode 100644 hello.py
```

- 상태 확인

```
print('basic')
print(list('python'))
```

```
print('basic')
```

```
print('basic')
```

Working
Directory

Staging
Area

.git directory
(Repository)

Remote

Remote
Area

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git log
commit 9b0b279880a6155497ae54d16a1d655ce2226e50 (HEAD -> master)
Author: edu4py <edu4py@outlook.kr>
Date: Thu May 26 10:49:26 2022 +0900
```

add file

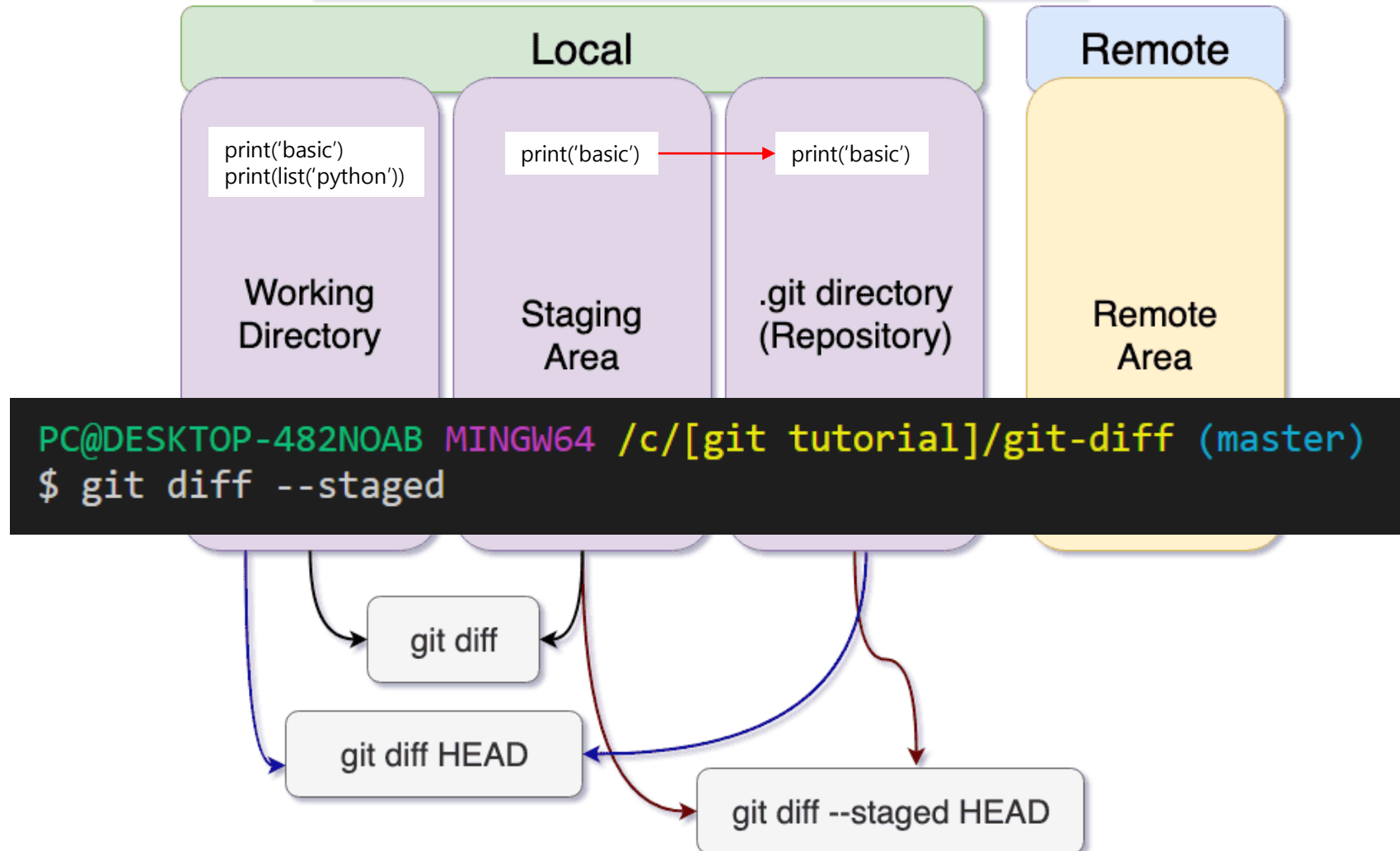
```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.py
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

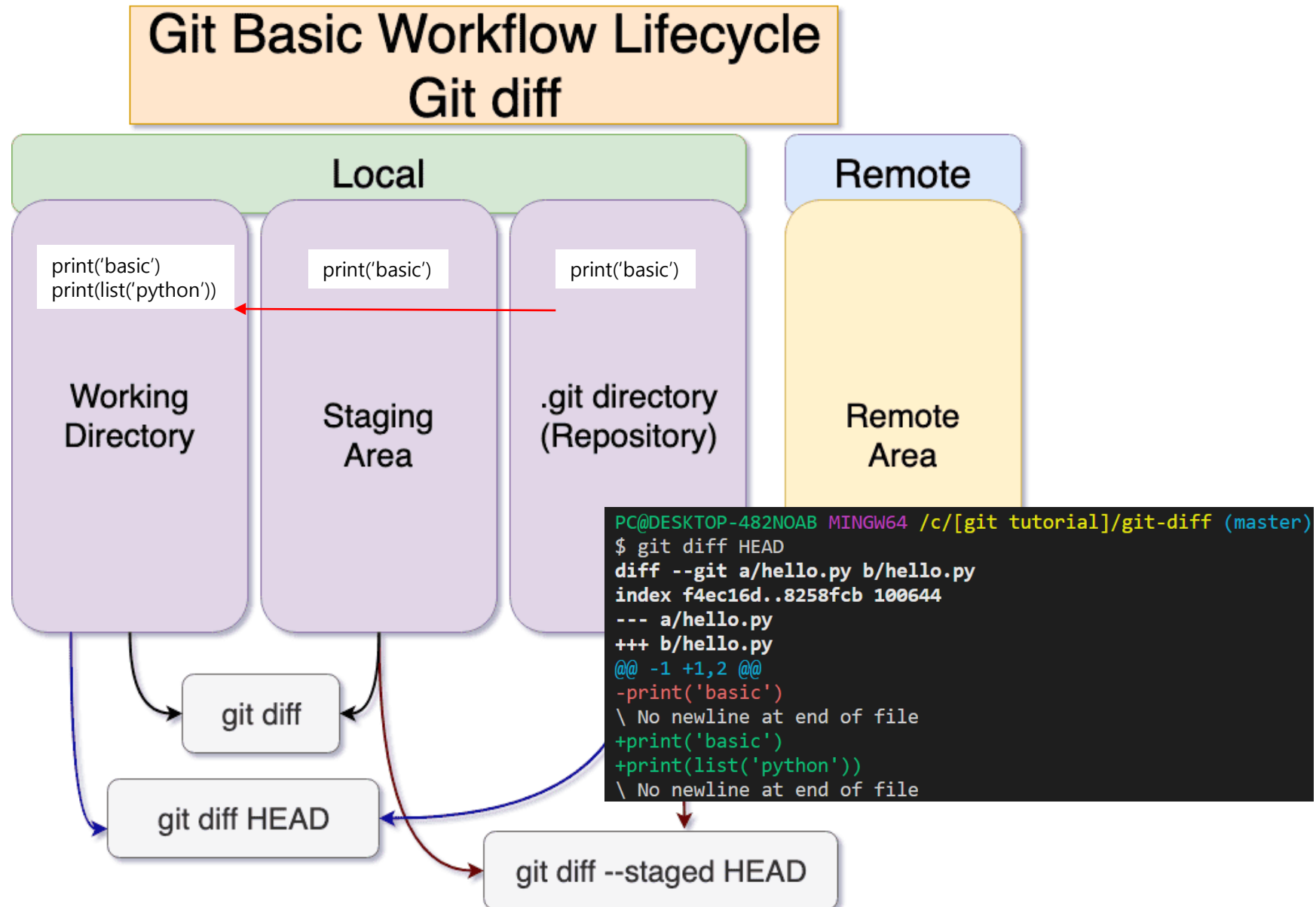
HEAD

- 차이 없음

Git Basic Workflow Lifecycle Git diff

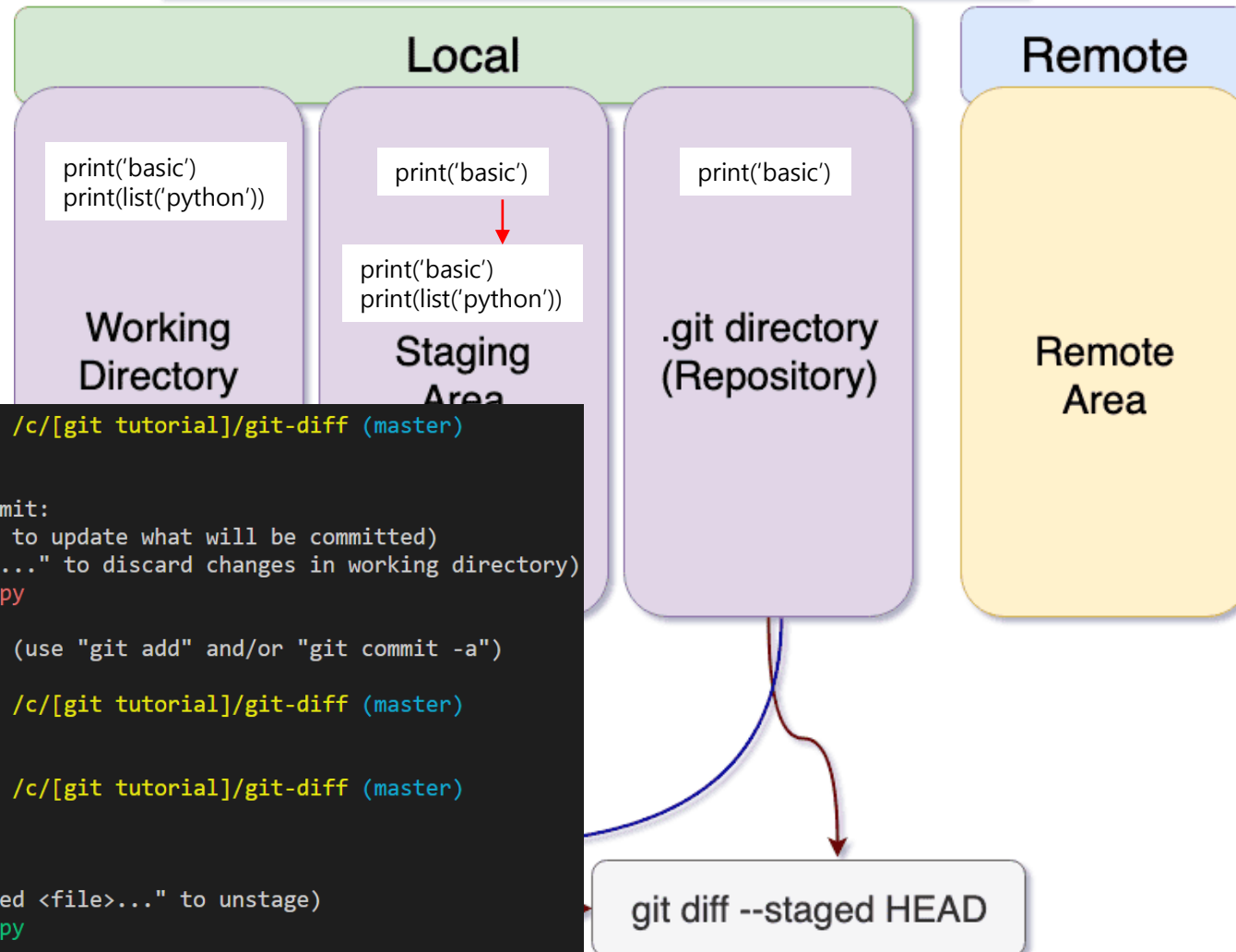


- 작업영역과
깃저장소
HEAD 비교



- Staging area
복사

Git Basic Workflow Lifecycle Git diff



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.py

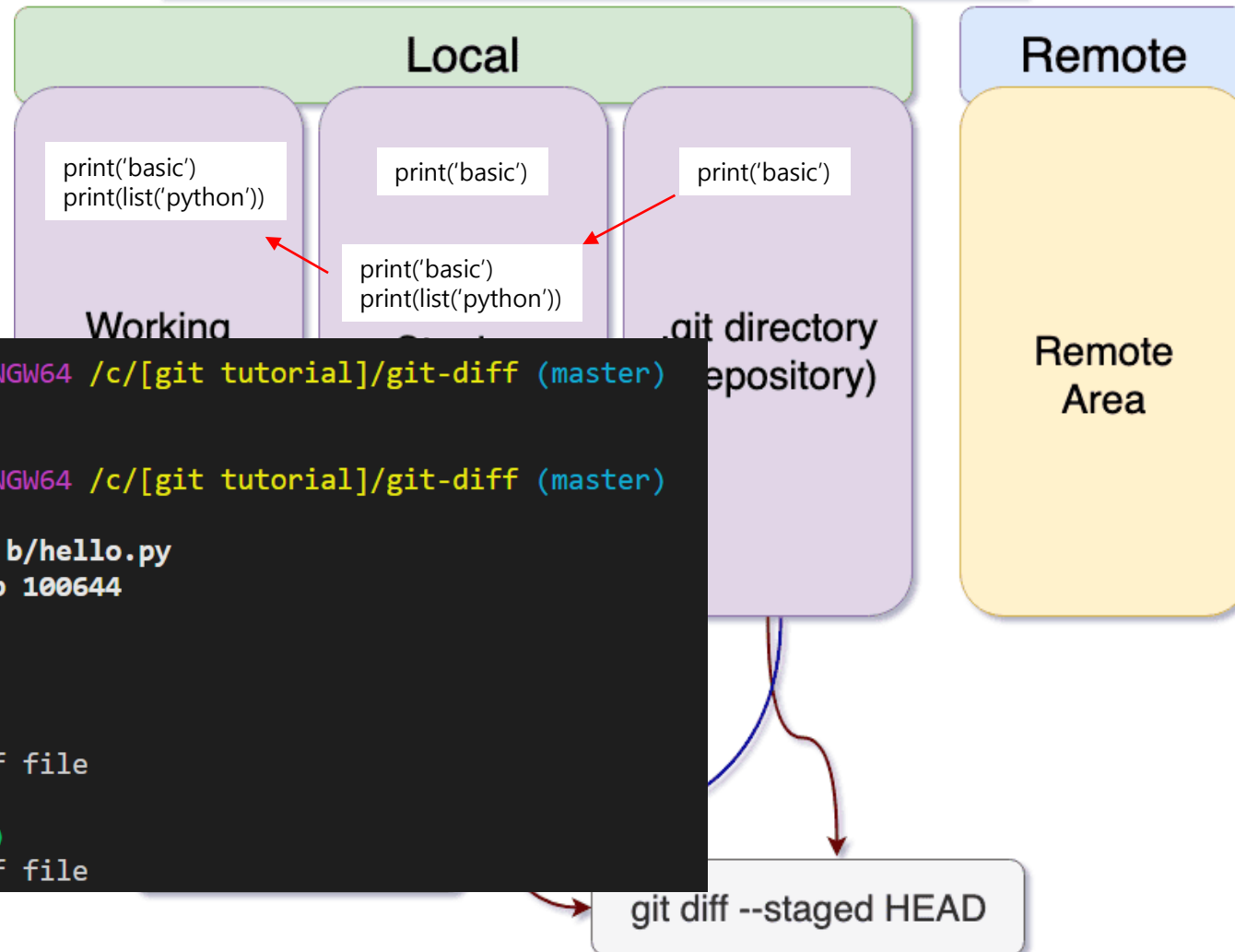
no changes added to commit (use "git add" and/or "git commit -a")

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git add hello.py

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello.py
```

- **Git diff**
 - 차이 없음

Git Basic Workflow Lifecycle Git diff



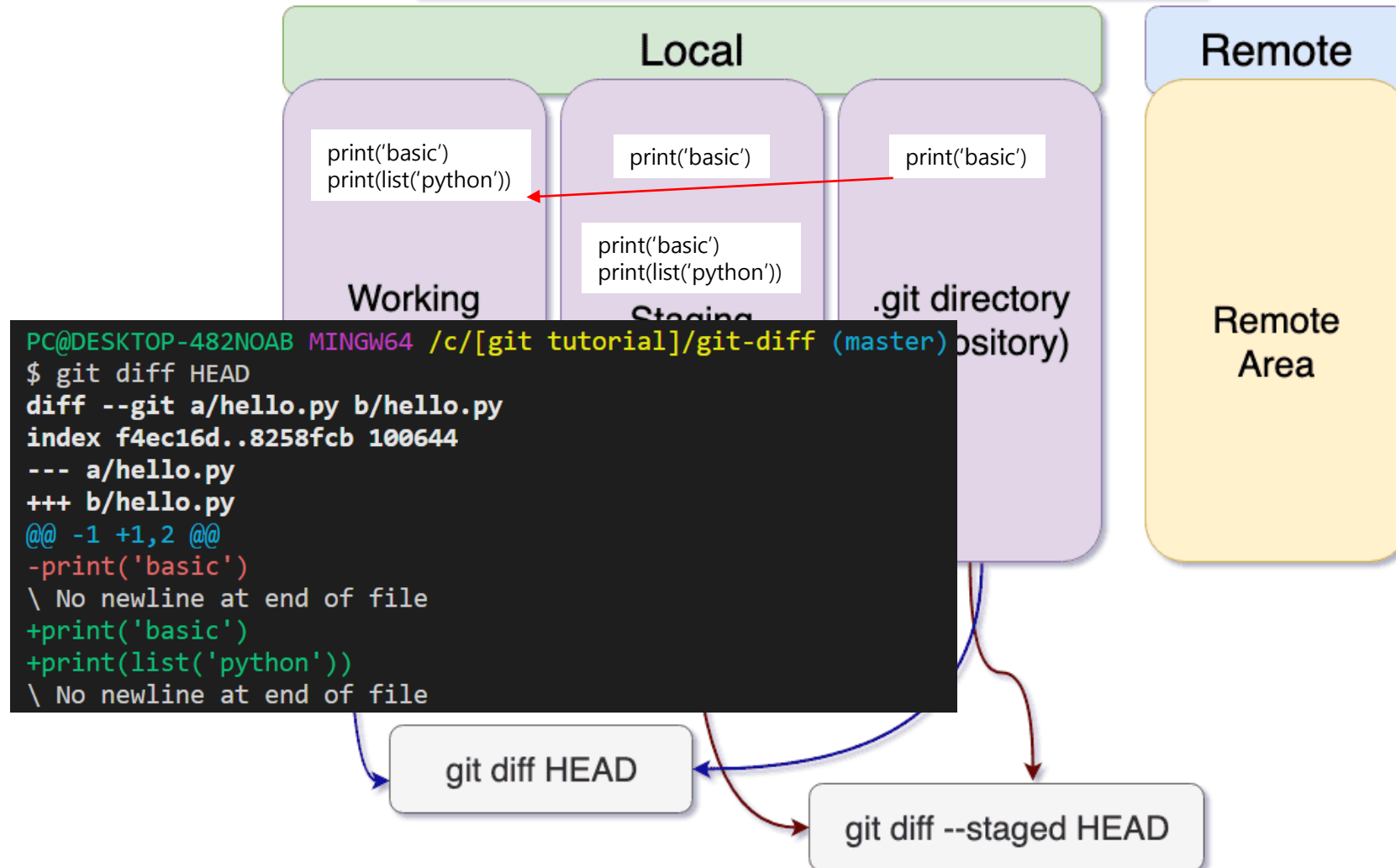
```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff
```

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff --staged
diff --git a/hello.py b/hello.py
index f4ec16d..8258fcb 100644
--- a/hello.py
+++ b/hello.py
@@ -1,2 @@
-print('basic')
\ No newline at end of file
+print('basic')
+print(list('python'))
\ No newline at end of file
```

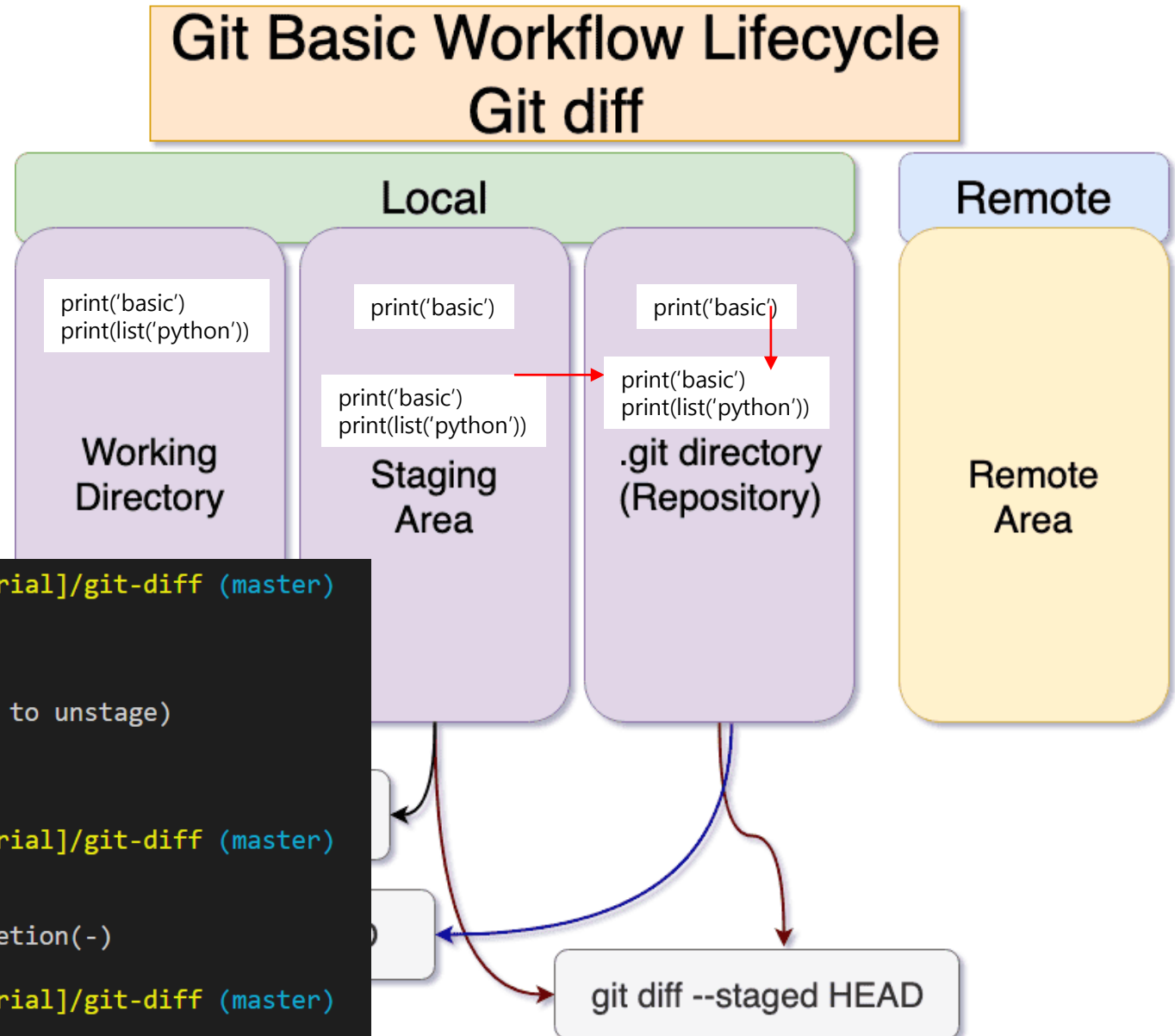
git diff --staged HEAD

- Git diff
 - 차이 없음

Git Basic Workflow Lifecycle Git diff



- Staging area에서 저장소로 이동
 - 3 장소가 모두 동일



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.py

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git commit -m 'insert 2 line'
[master 6a123f1] insert 2 line
1 file changed, 2 insertions(+), 1 deletion(-)

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git status
On branch master
nothing to commit, working tree clean
```

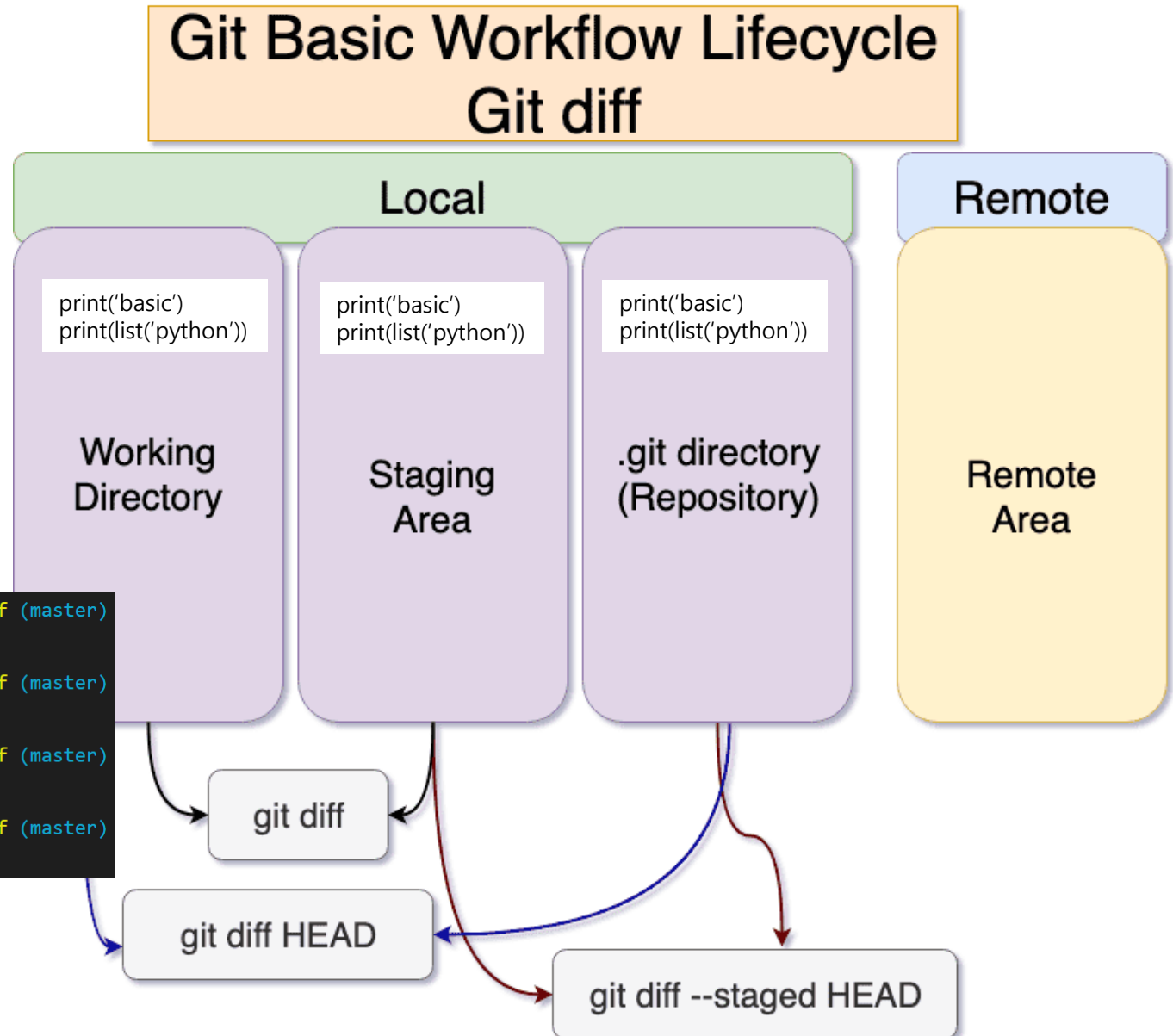
- **\$ git diff**
 - 차이 없음
- **\$ git diff --staged**
 - 차이 없음
- **\$ git diff HEAD**
 - 차이 없음

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff --staged

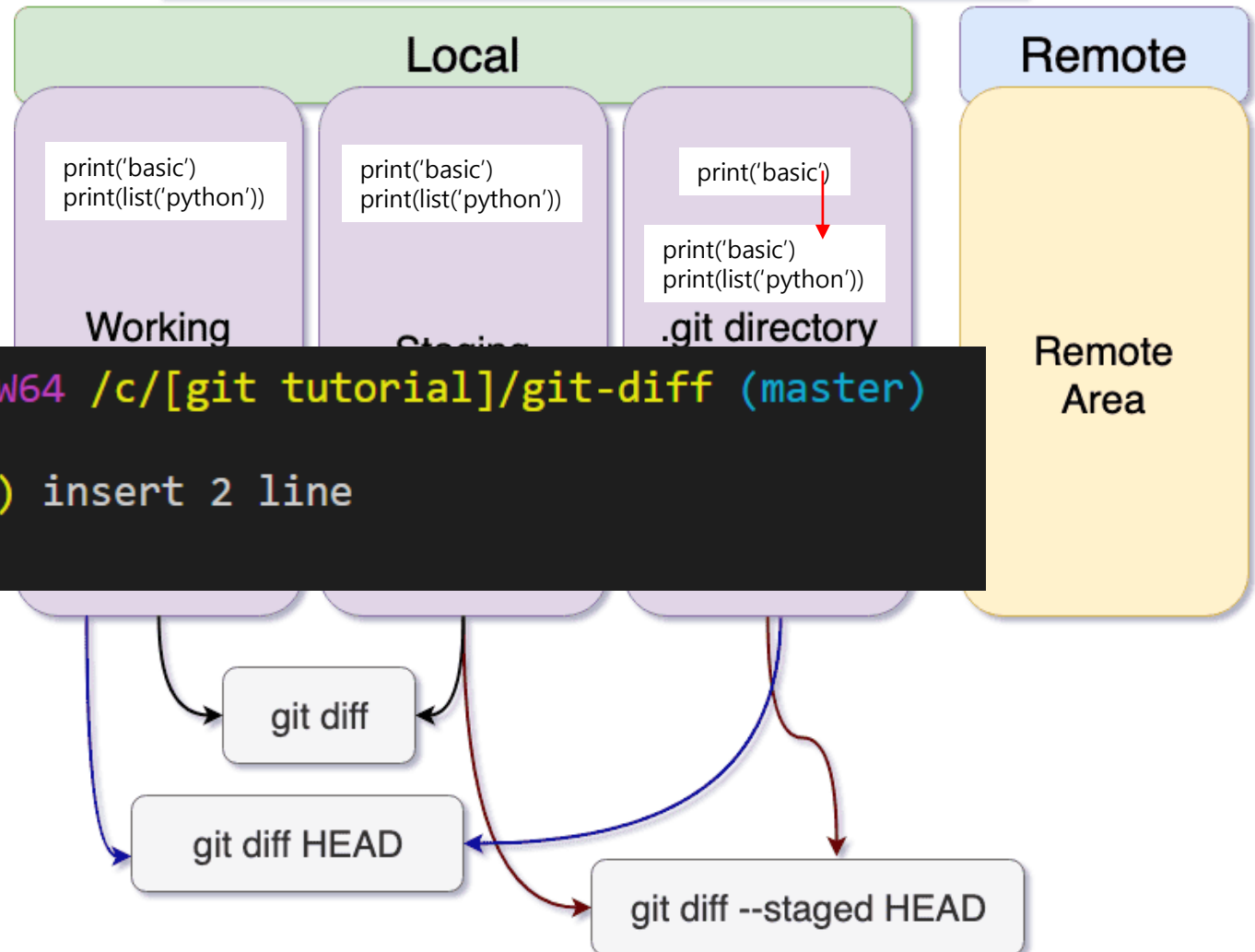
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff HEAD

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff --staged HEAD
```



Git Basic Workflow Lifecycle

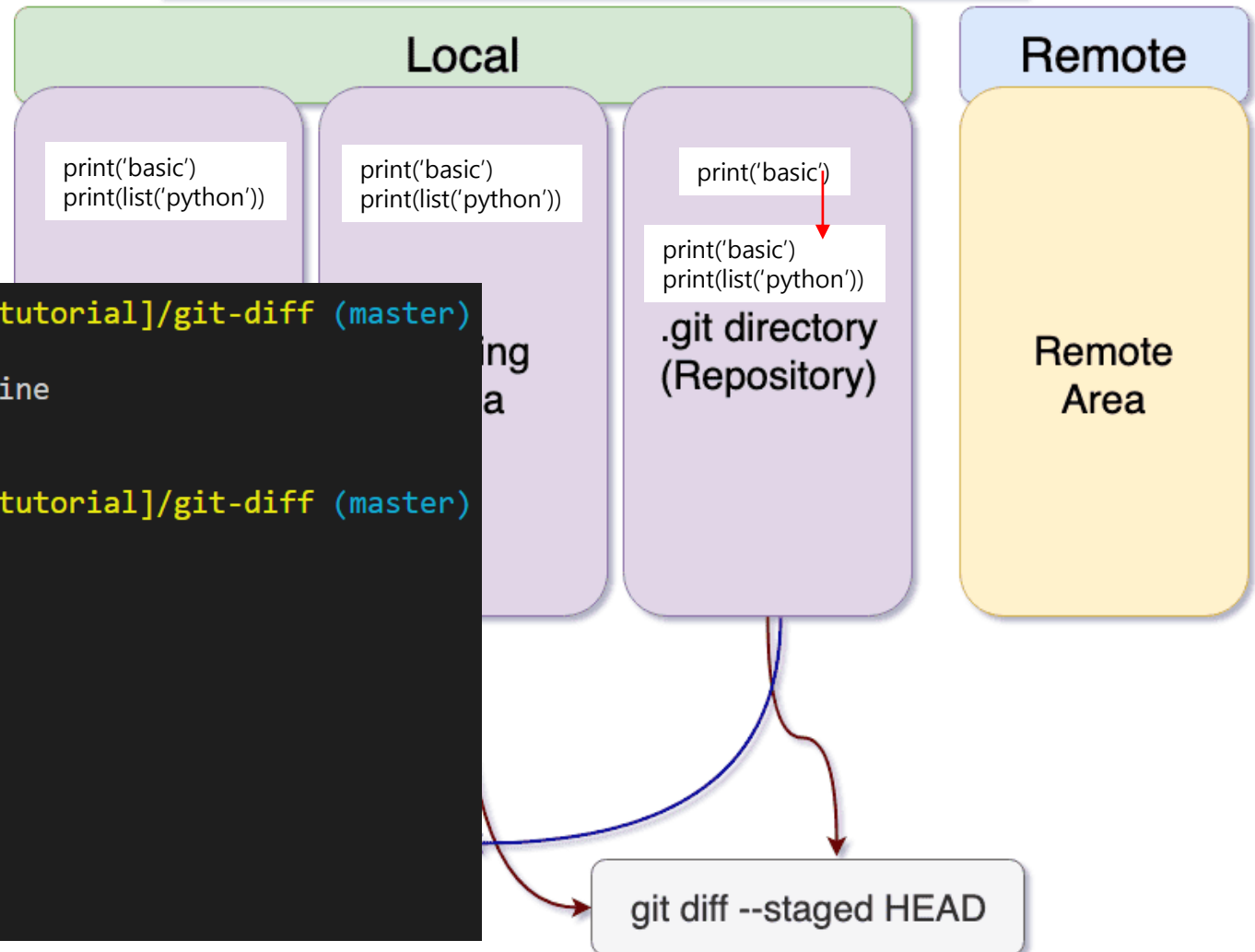
Git diff



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git log --oneline
6a123f1 (HEAD -> master) insert 2 line
9b0b279 add file
```

- [이전커밋] [이후커밋]

Git Basic Workflow Lifecycle Git diff

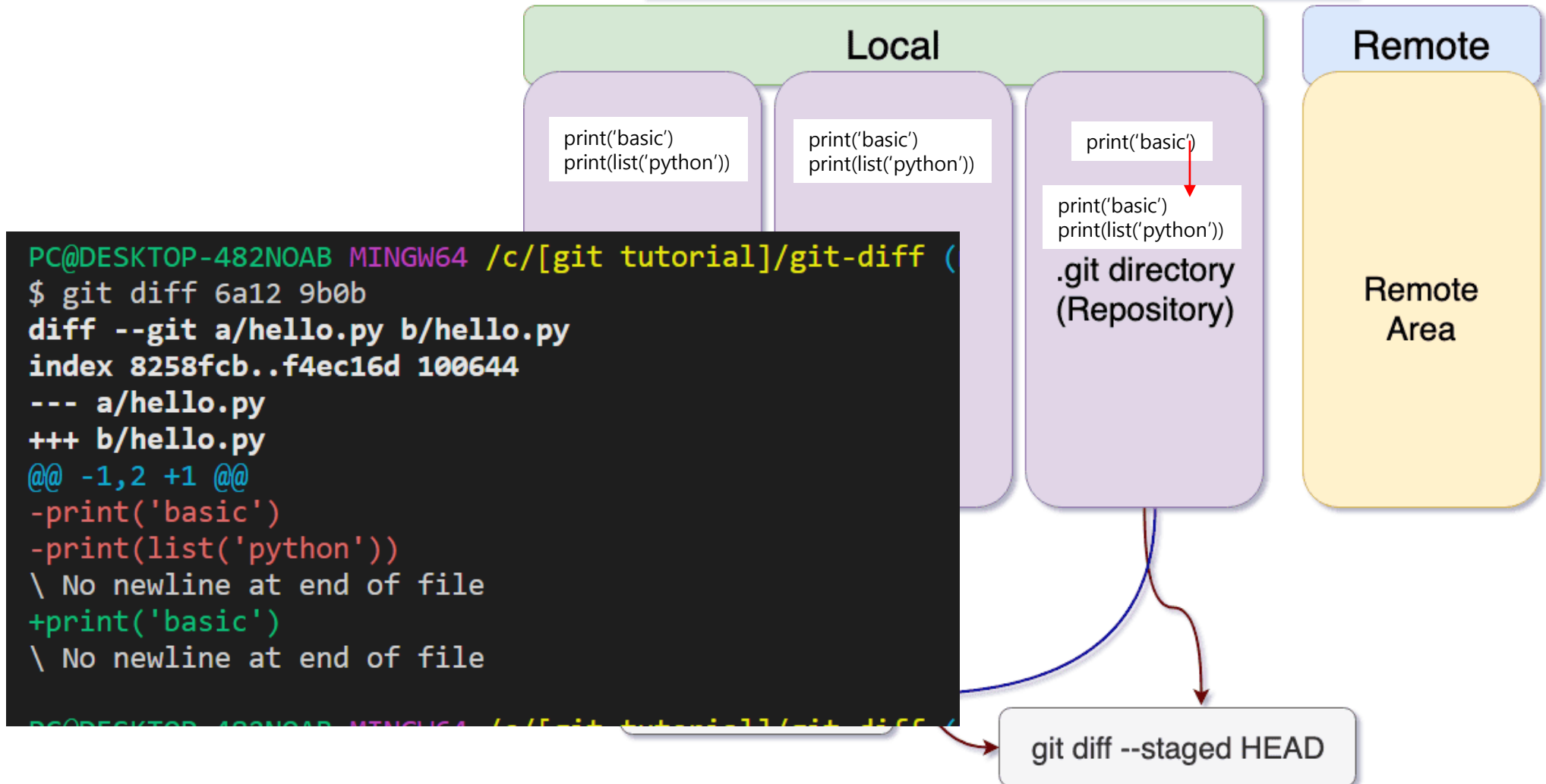


```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git log --oneline
6a123f1 (HEAD -> master) insert 2 line
9b0b279 add file

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff 9b0b279 6a12
diff --git a/hello.py b/hello.py
index f4ec16d..8258fcb 100644
--- a/hello.py
+++ b/hello.py
@@ -1,2 @@
-print('basic')
\ No newline at end of file
+print('basic')
+print(list('python'))
\ No newline at end of file
```

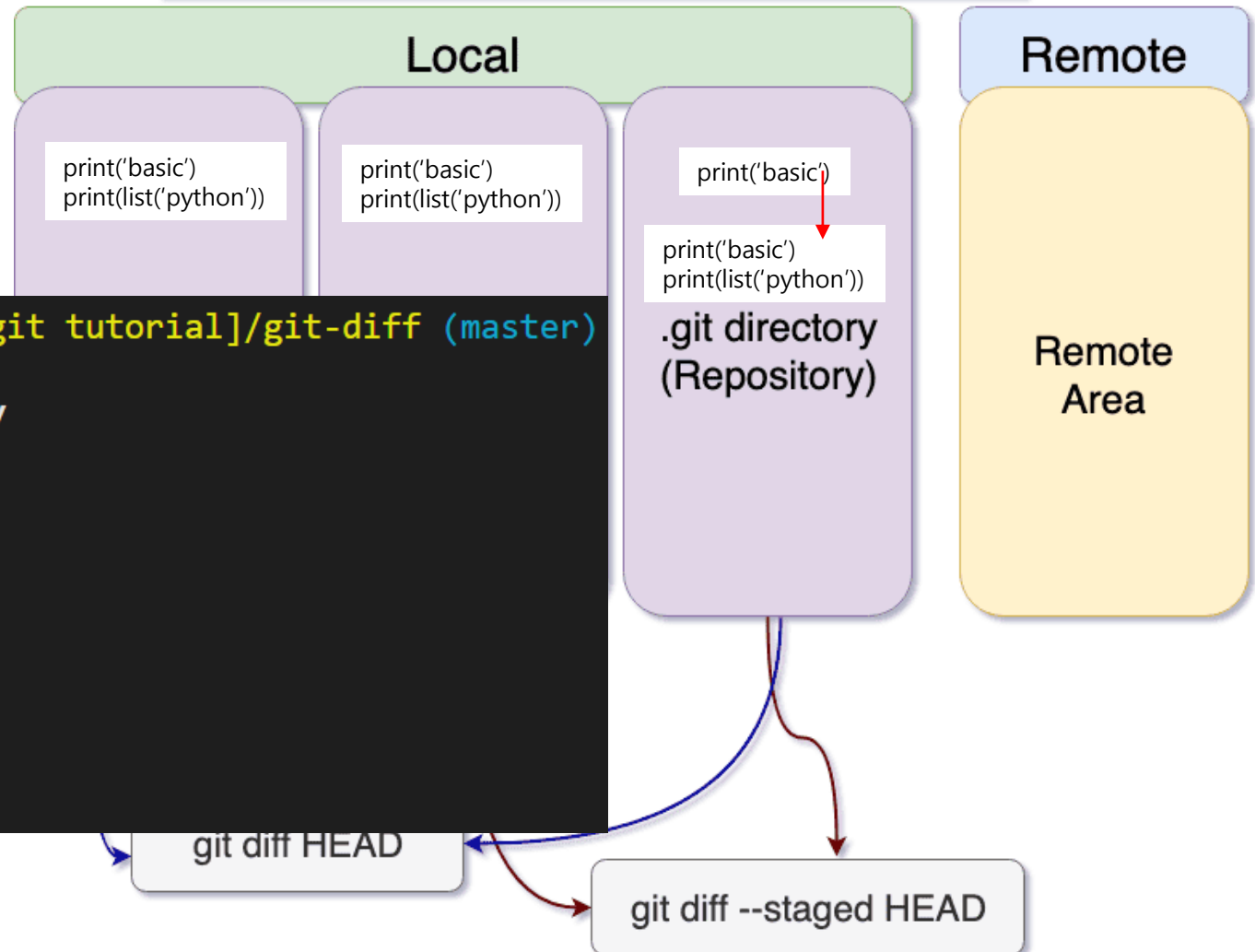
- [이후커밋] [이전커밋]

Git Basic Workflow Lifecycle Git diff



- HEAD^ HEAD

Git Basic Workflow Lifecycle Git diff

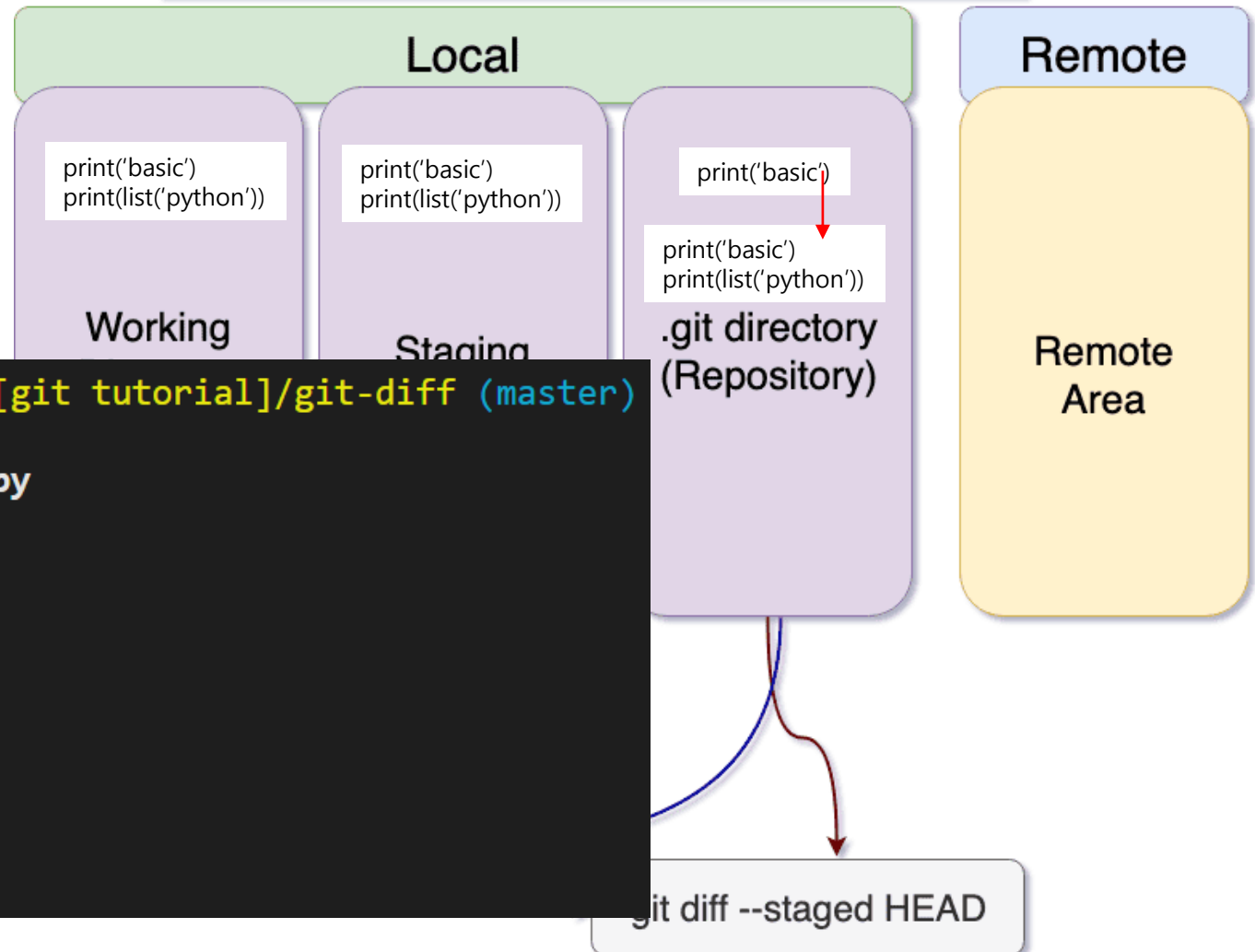


```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff HEAD^ HEAD
diff --git a/hello.py b/hello.py
index f4ec16d..8258fcb 100644
--- a/hello.py
+++ b/hello.py
@@ -1,2 @@
-print('basic')
\ No newline at end of file
+print('basic')
+print(list('python'))
\ No newline at end of file
```

git diff [id1] [id2]

- HEAD HEAD^
- HEAD HEAD~
- ~
 - 한번에 한 커밋 아래로 움직이는 ^
 - 한번에 여러 커밋 아래로 움직이면 ~<num>

Git Basic Workflow Lifecycle Git diff



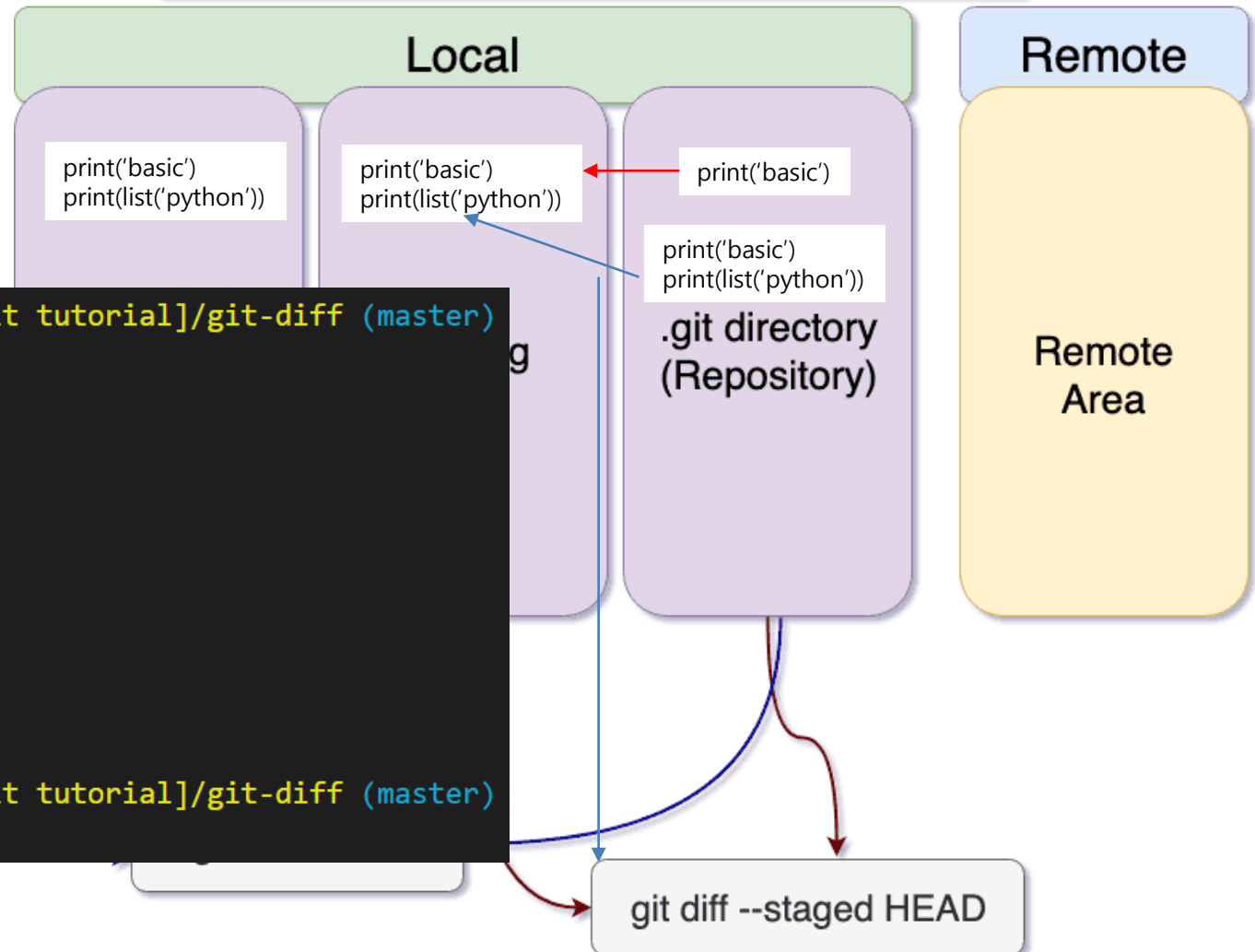
```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff HEAD HEAD~
diff --git a/hello.py b/hello.py
index 8258fcb..f4ec16d 100644
--- a/hello.py
+++ b/hello.py
@@ -1,2 +1 @@
-print('basic')
-print(list('python'))
\ No newline at end of file
+print('basic')
\ No newline at end of file
```


git diff --staged HEAD^

깃과 깃허브 Python language

Git Basic Workflow Lifecycle Git diff

- ~
- 한번에 한 커밋 아래로 움직이는 ^
- 한번에 여러 커밋 아래로 움직이면 ~<num>



```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff --staged HEAD^
diff --git a/hello.py b/hello.py
index f4ec16d..8258fcb 100644
--- a/hello.py
+++ b/hello.py
@@ -1,2 @@
-print('basic')
\ No newline at end of file
+print('basic')
+print(list('python'))
\ No newline at end of file

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git diff --staged HEAD
```

- 파일 수정

```
print('basic')
print(list('python'))

lst = list('python')
for i in lst:
    print(i, '*' * (ord(i) -
ord('a'))))
```

- \$ git commit -am 'add graph'

- \$ git status

- \$ git log

```
PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git status
On branch master
nothing to commit, working tree clean

PC@DESKTOP-482NOAB MINGW64 /c/[git tutorial]/git-diff (master)
$ git log --oneline
9eac172 (HEAD -> master) add graph
6a123f1 insert 2 line
9b0b279 add file
```

- **commit된 파일상태와 현재 수정중인 상태 비교**
 - `git diff`
- **commit된 파일상태와 add된 파일 상태 비교**
 - `git diff --staged`
- **commit간의 상태 비교하기 - commit hash 이용**
 - `git diff [비교할commit해쉬1] [비교할commit해쉬2]`
 - `git diff 048171 0c747d`
- **commit간의 상태 비교하기 - HEAD 이용**
 - `git diff HEAD HEAD^`
 - 가장 최근의 커밋과 그 전의 커밋을 비교한다
- **branch간의 상태 비교하기 - HEAD 이용**
 - `git diff [비교할branch1] [비교할branch2]`
 - ex) `git diff feature/test origin/master`
 - local의 feature/test브랜치와 remote의 master branch 비교

