Git Branching

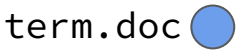
Software Engineering for Scientists

A **branch** is an independent line of development

(new file, changes to files)

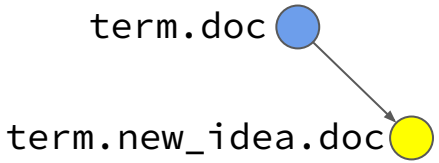
A **branch** is an independent line of development

(new file, changes to files)



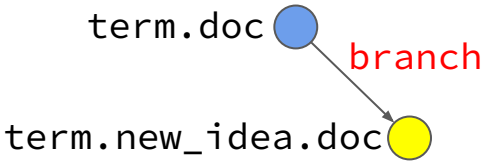
A **branch** is an independent line of development

(new file, changes to files)



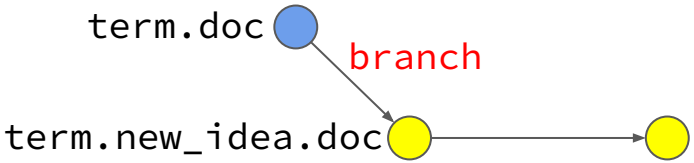
A **branch** is an independent line of development

(new file, changes to files)



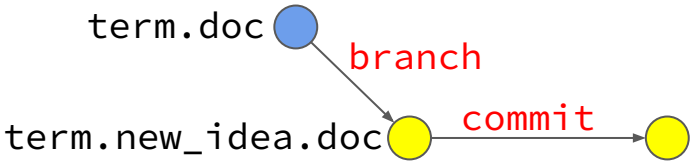
A **branch** is an independent line of development

(new file, changes to files)



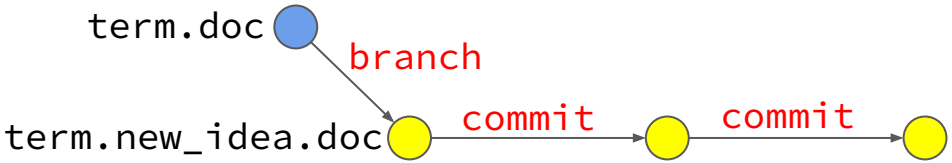
A **branch** is an independent line of development

(new file, changes to files)



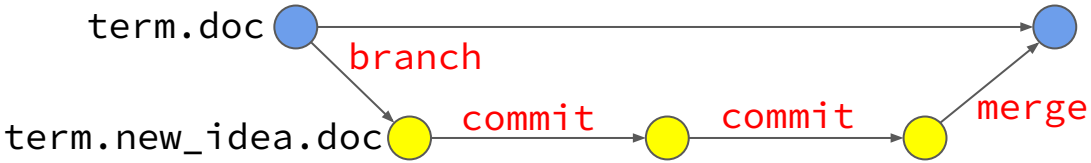
A **branch** is an independent line of development

(new file, changes to files)



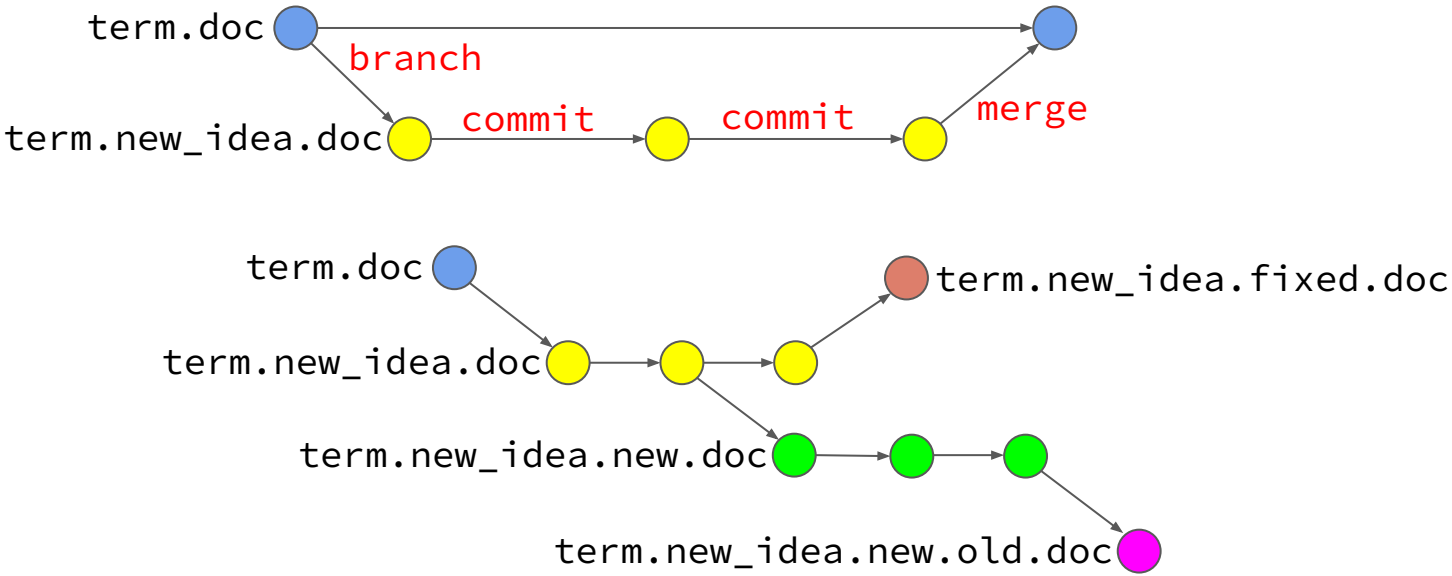
A **branch** is an independent line of development

(new file, changes to files)



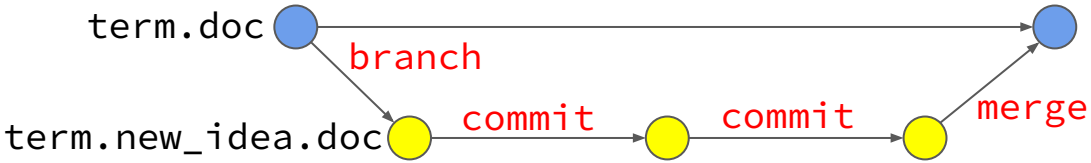
A **branch** is an independent line of development

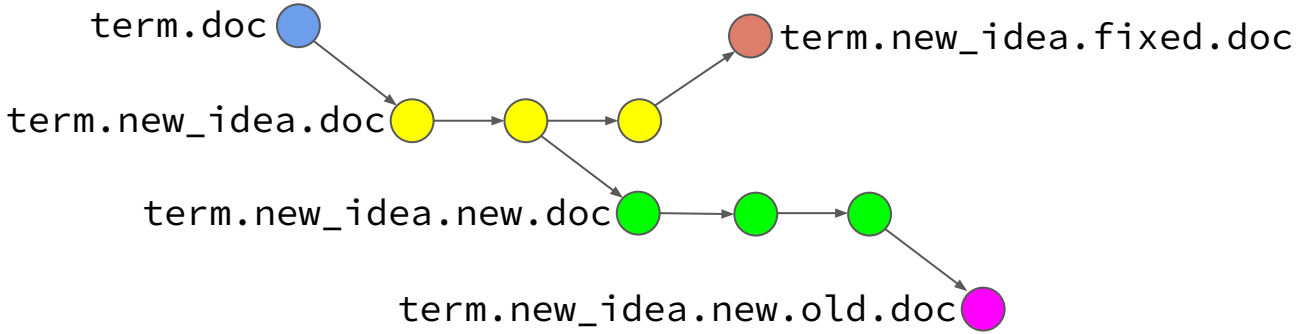
(new file, changes to files)



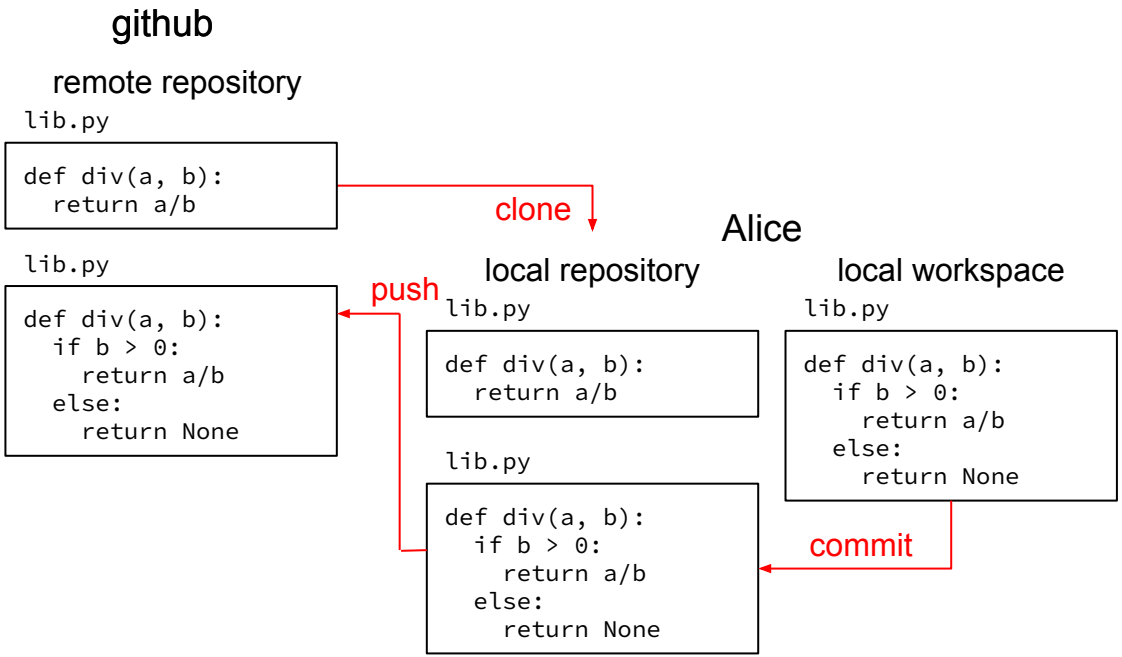
A **branch** is an independent line of development

(new file, changes to files)





git branching/merging is cheap and fast

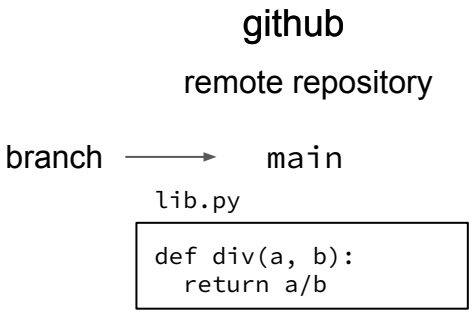

github
remote repository
lib.py
def div(a, b):
return a/b
clone
Alice
local repository
local workspace
lib.py
def div(a, b):
if b > 0:
return a/b
else:
return None
push
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b > 0:
return a/b
else:
return None
lib.py
def div(a, b):
if b > 0:
return a/b
else:
return None
commit

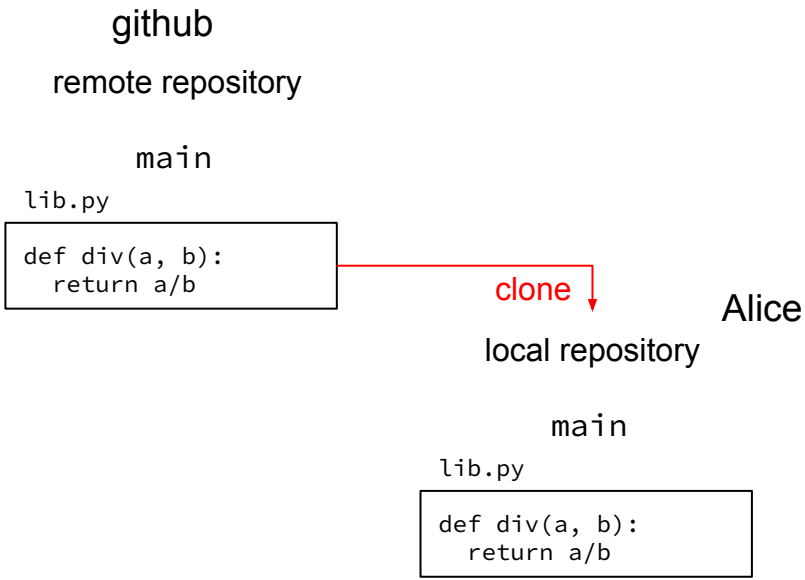
github

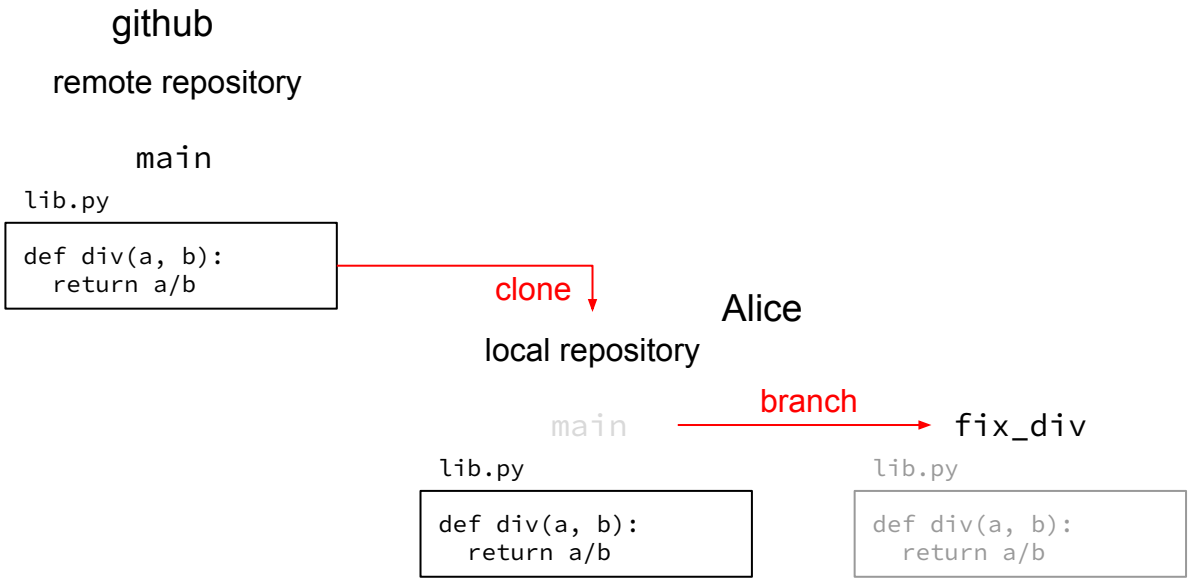
remote repository

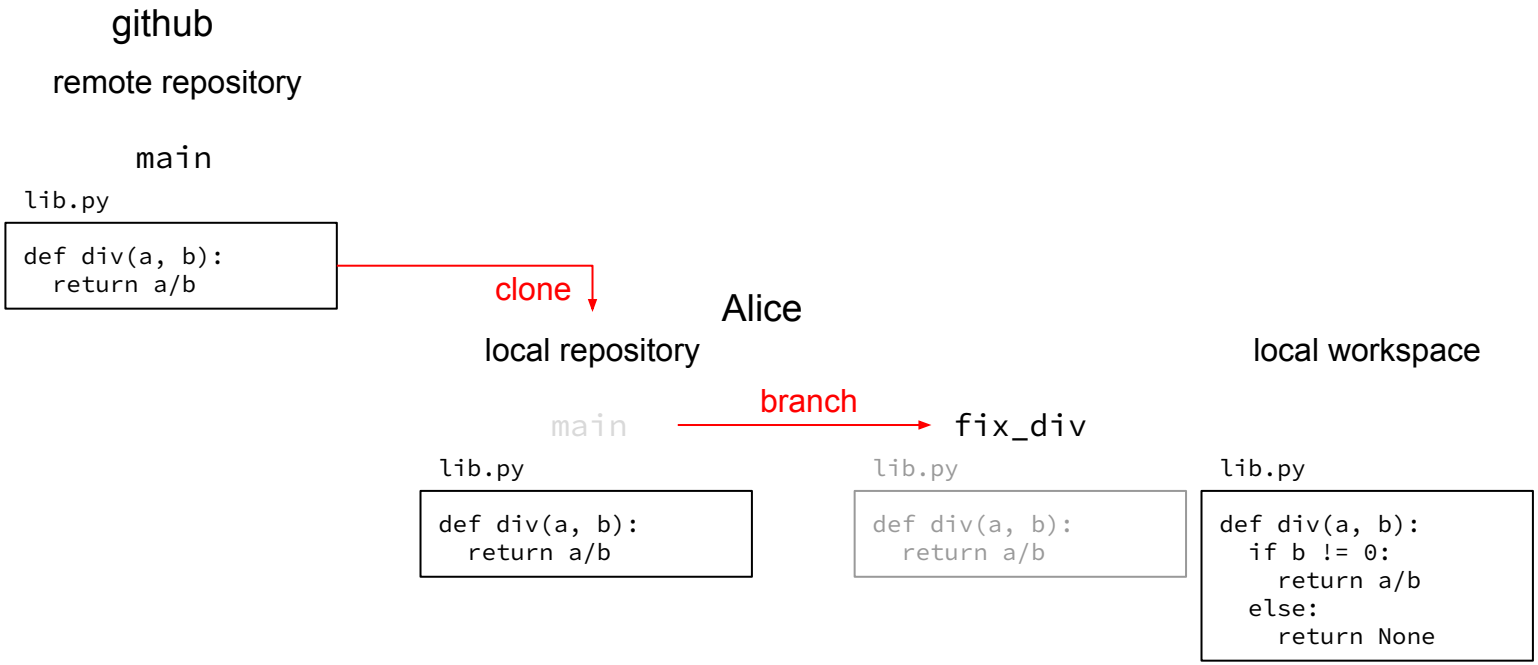
lib.py

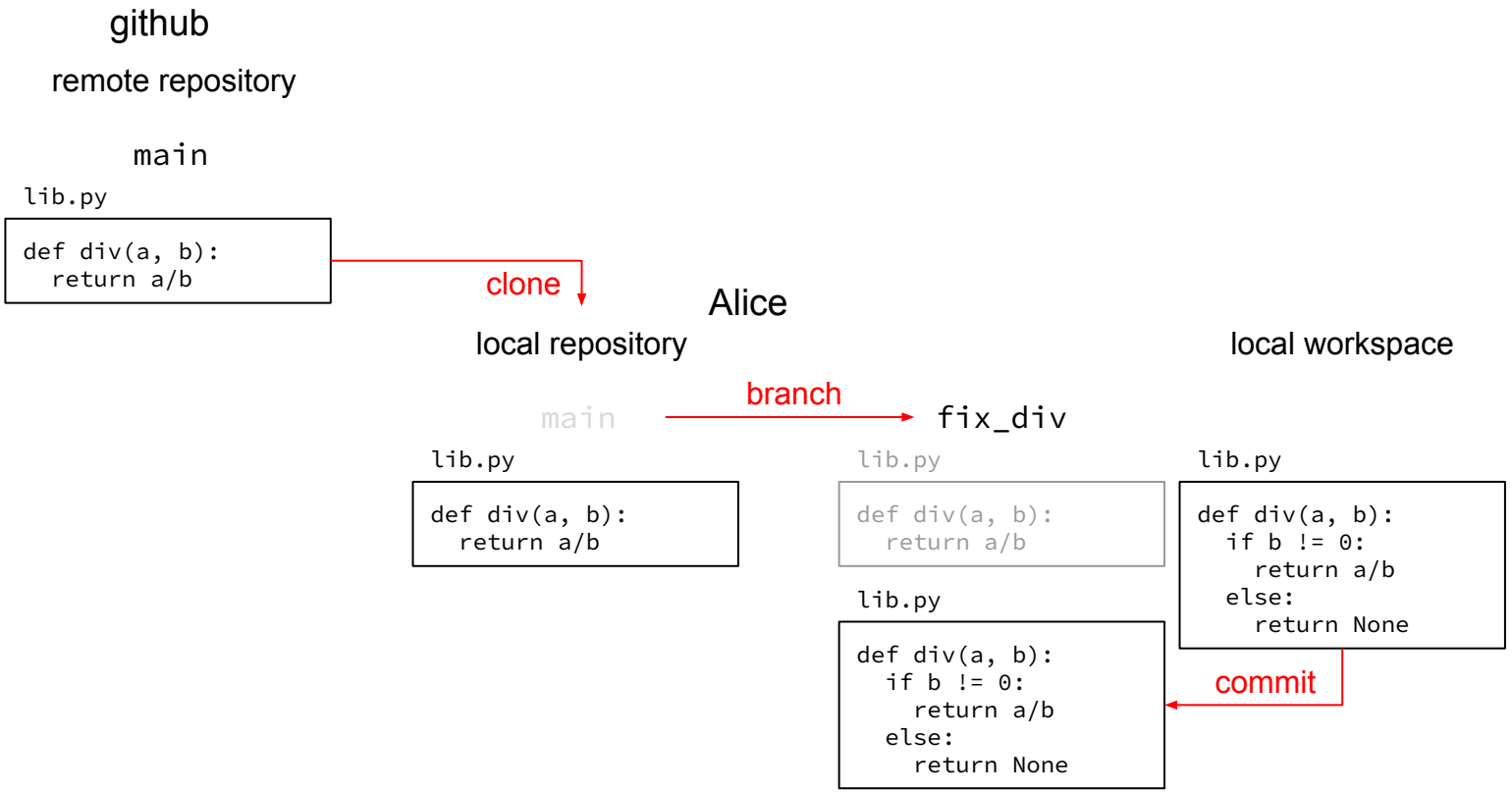
```
def div(a, b):
  return a/b
```

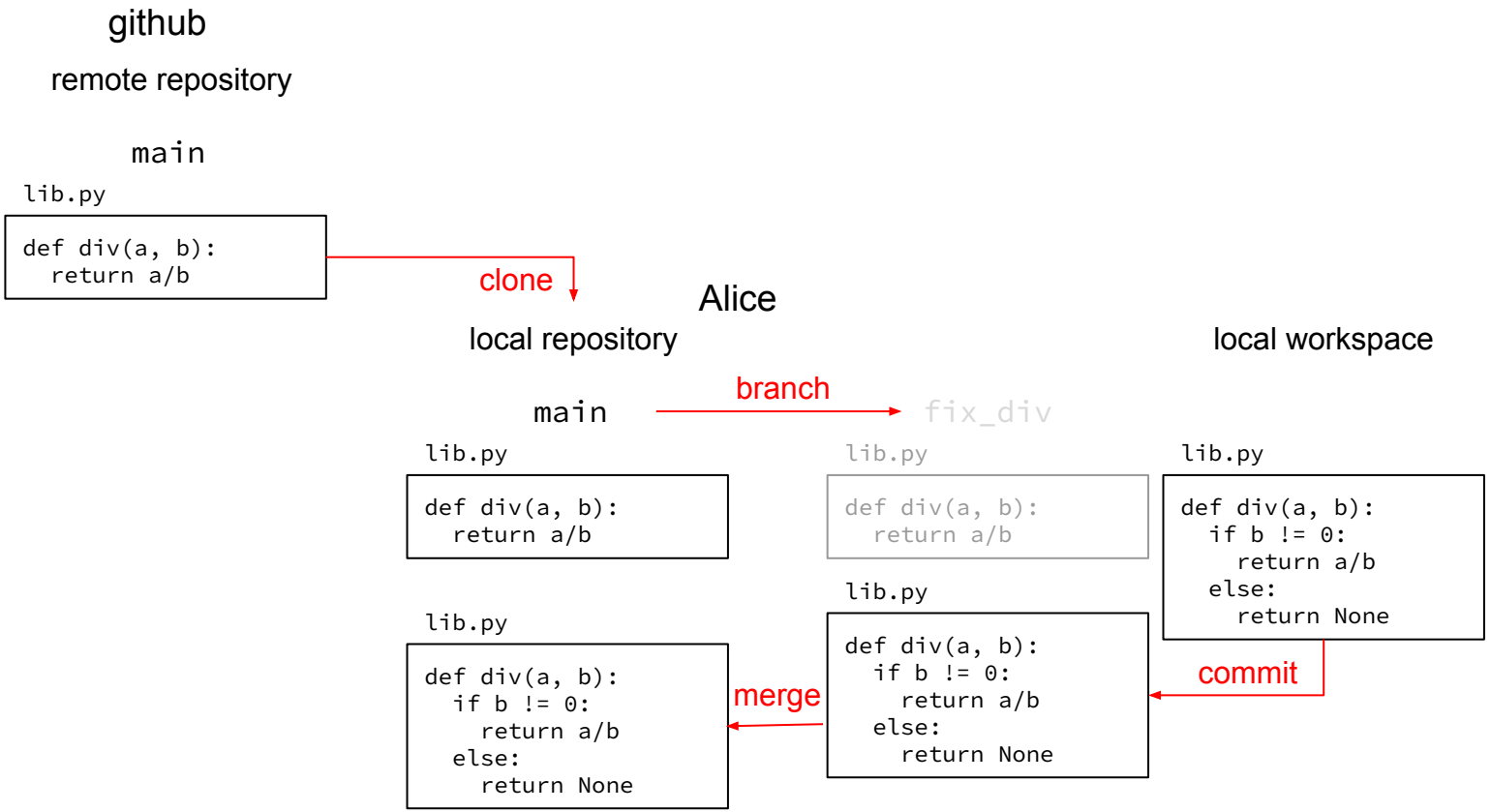

github
remote repository
branch
main
lib.py
def div(a, b):
return a/b

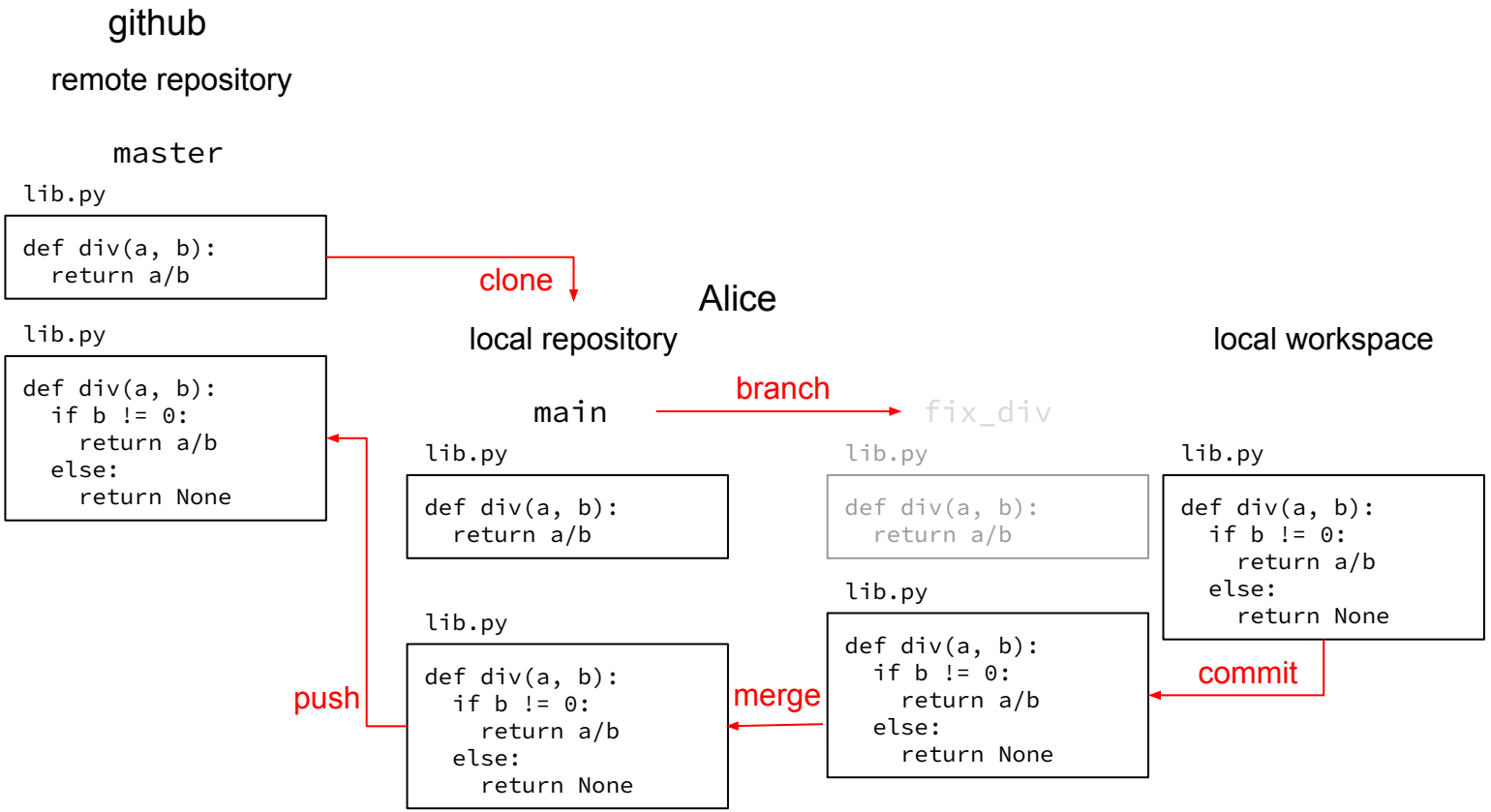

github
remote repository
main
lib.py
def div(a, b):
return a/b
clone
Alice
local repository
main
lib.py
def div(a, b):
return a/b


github
remote repository
main
lib.py
def div(a, b):
return a/b
clone
Alice
local repository
main
branch
fix_div
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b

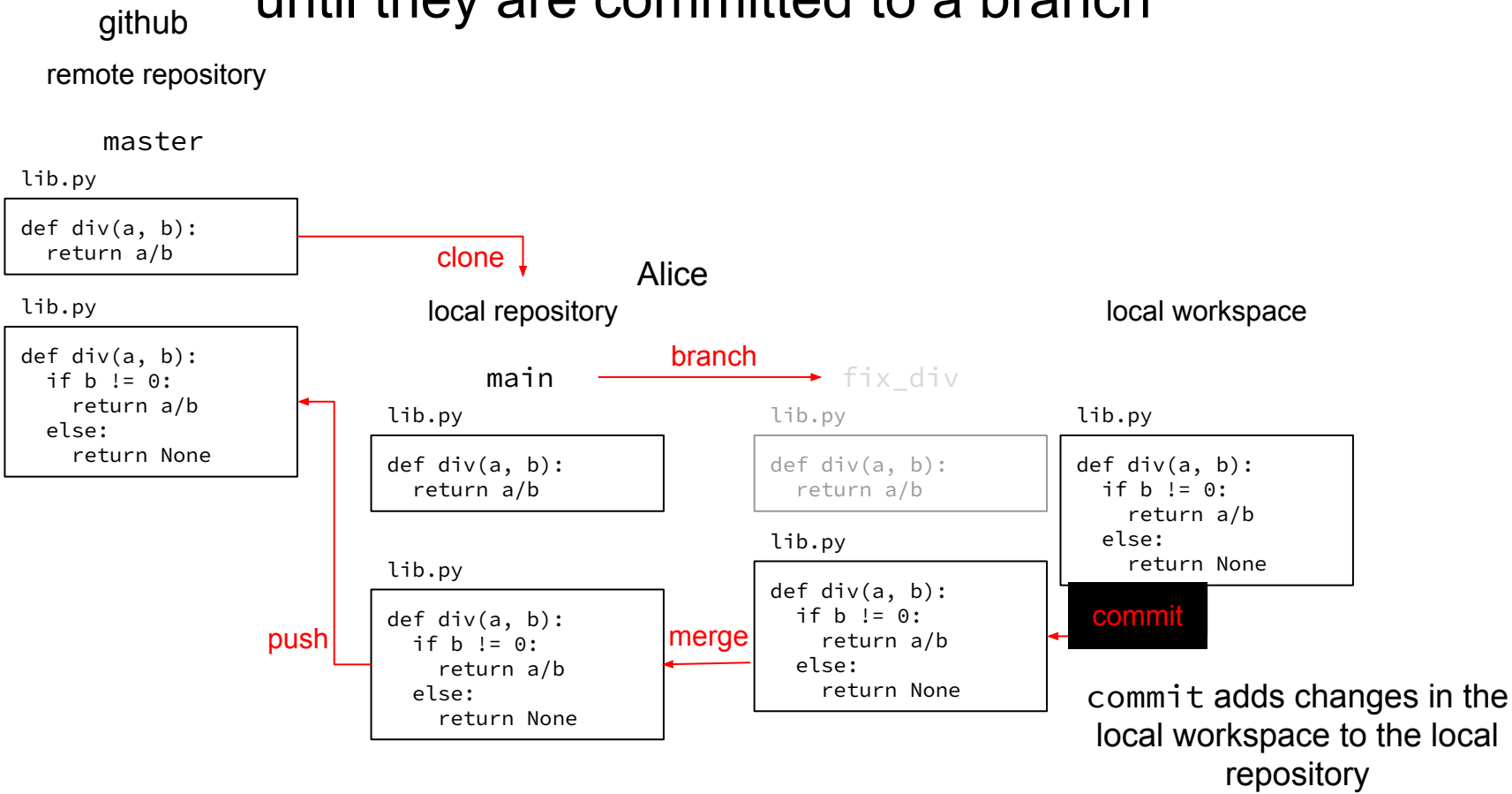

github
remote repository
main
lib.py
def div(a, b):
return a/b
clone
Alice
local repository
local workspace
branch
main
fix_div
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None


github
remote repository
main
lib.py
def div(a, b):
return a/b
clone
Alice
local repository
local workspace
main
branch
fix_div
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
commit


github
remote repository
main
lib.py
def div(a, b):
return a/b
clone
Alice
local repository
local workspace
main
branch
fix_div
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
merge
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
commit


github
remote repository
master
lib.py
def div(a, b):
return a/b
clone
Alice
local repository
local workspace
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
branch
main
fix_div
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
commit
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
merge
push
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None

Edits and new files will stay in your local workspace until they are committed to a branch



Alice

local repository

main

lib.py

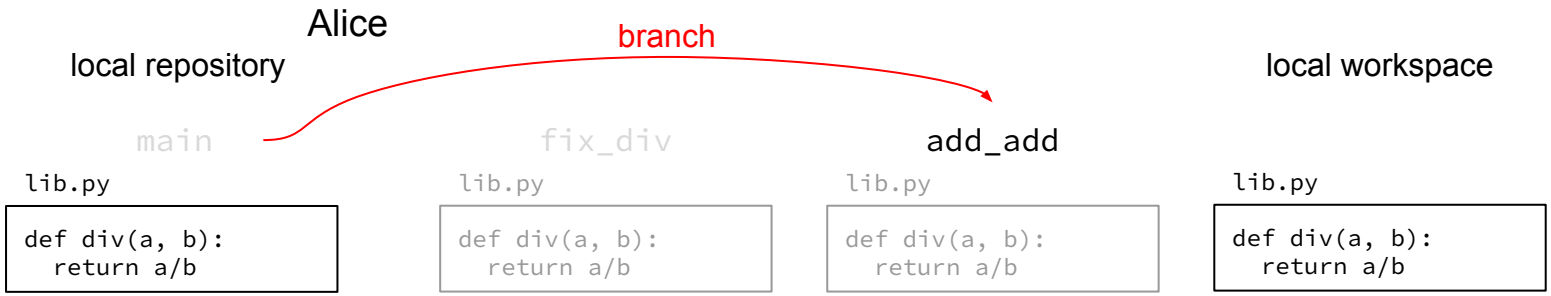
```
def div(a, b):
  return a/b
```

local workspace

lib.py

```
def div(a, b):
  return a/b
```


Alice
local repository
local workspace
branch
main
fix_div
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b


Alice
local repository
branch
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b

Alice

local repository

main

lib.py

```
def div(a, b):
  return a/b
```

fix_div

lib.py

```
def div(a, b):
  return a/b
```

add_add

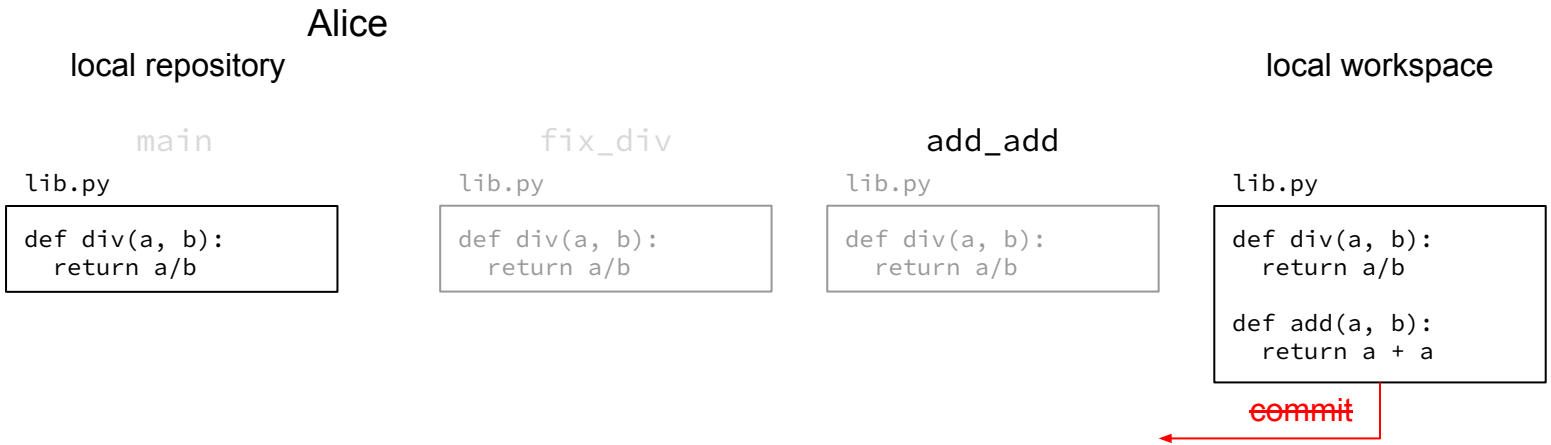
lib.py

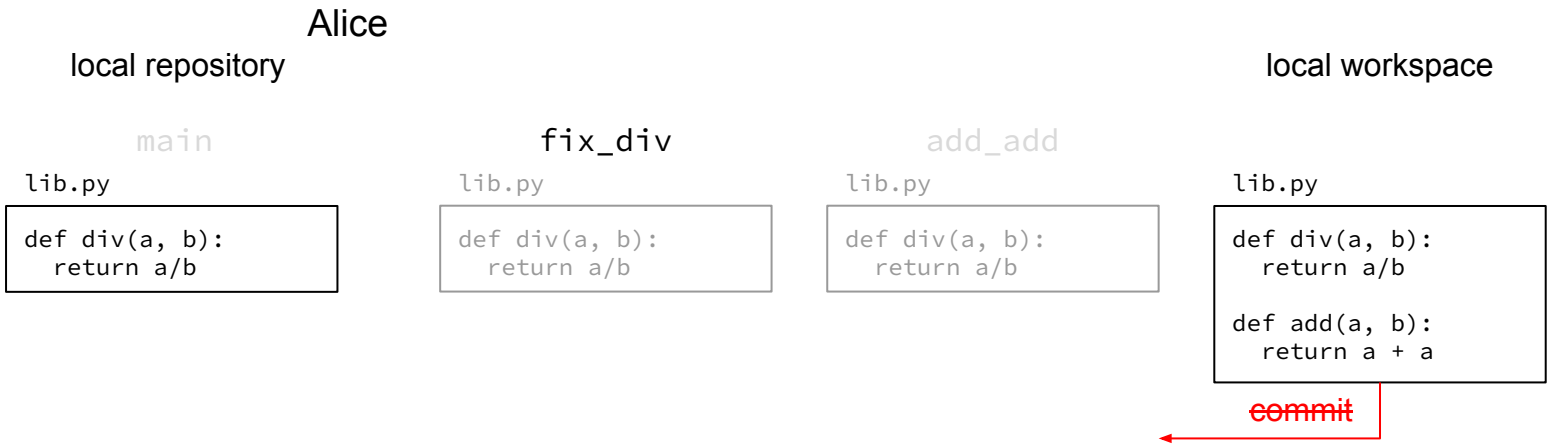
```
def div(a, b):
  return a/b
```

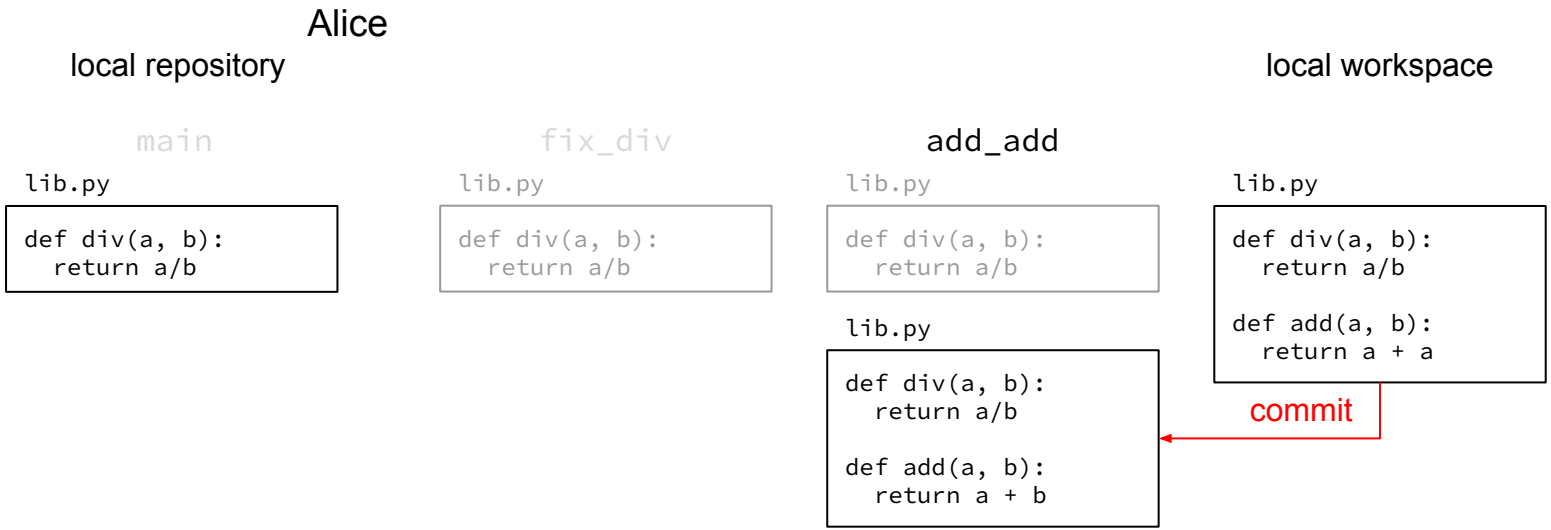
local workspace

lib.py

```
def div(a, b):
  return a/b

def add(a, b):
  return a + a
```


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + a
commit


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + a
commit


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + a
commit
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + b

Alice

local repository

main

lib.py

```
def div(a, b):
  return a/b
```

fix_div

lib.py

```
def div(a, b):
  return a/b
```

add_add

lib.py

```
def div(a, b):
  return a/b
```

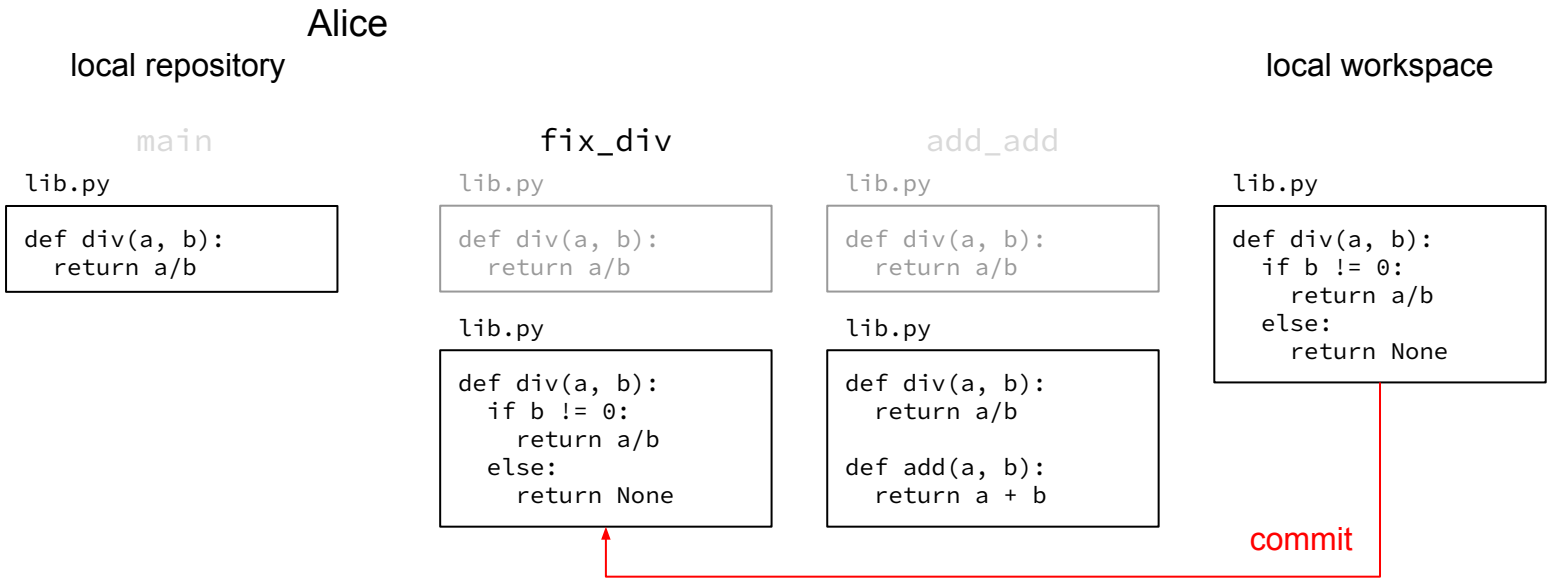
lib.py

```
def div(a, b):
  return a/b

def add(a, b):
  return a + b
```

local workspace

lib.py

```
def div(a, b):
  return a/b
```


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
commit

Alice

local repository

local workspace

main

lib.py

```
def div(a, b):
  return a/b
```

fix_div

lib.py

```
def div(a, b):
  return a/b
```

lib.py

```
def div(a, b):
  if b != 0:
    return a/b
  else:
    return None
```

add_add

lib.py

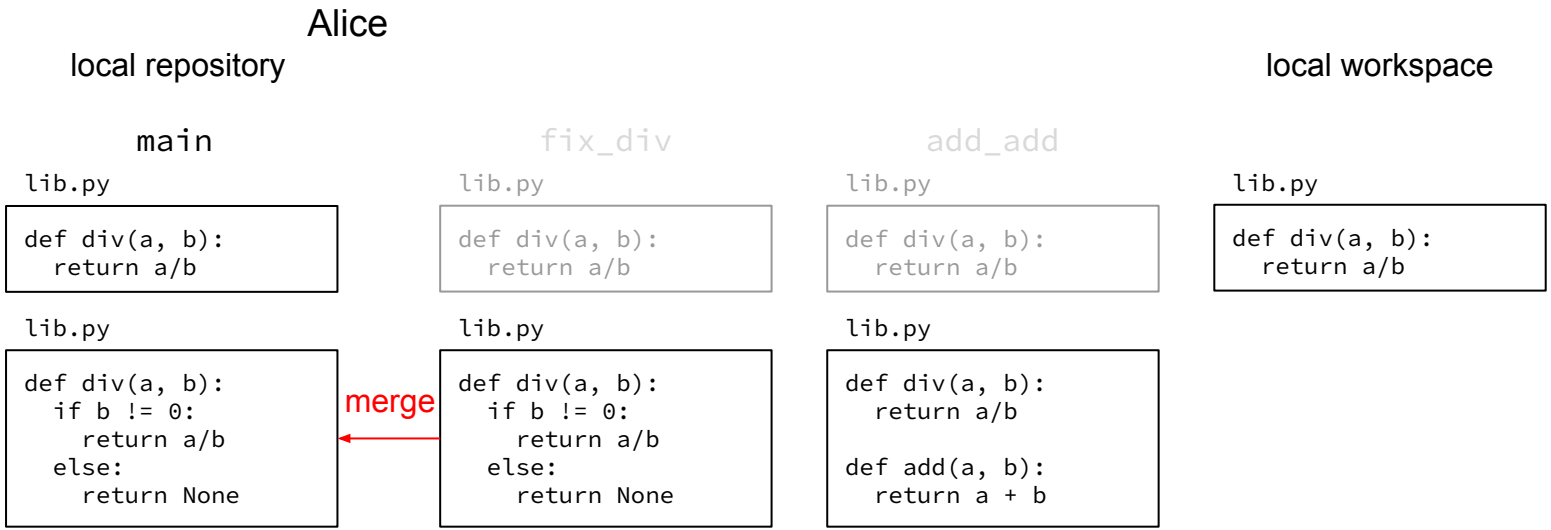
```
def div(a, b):
  return a/b
```

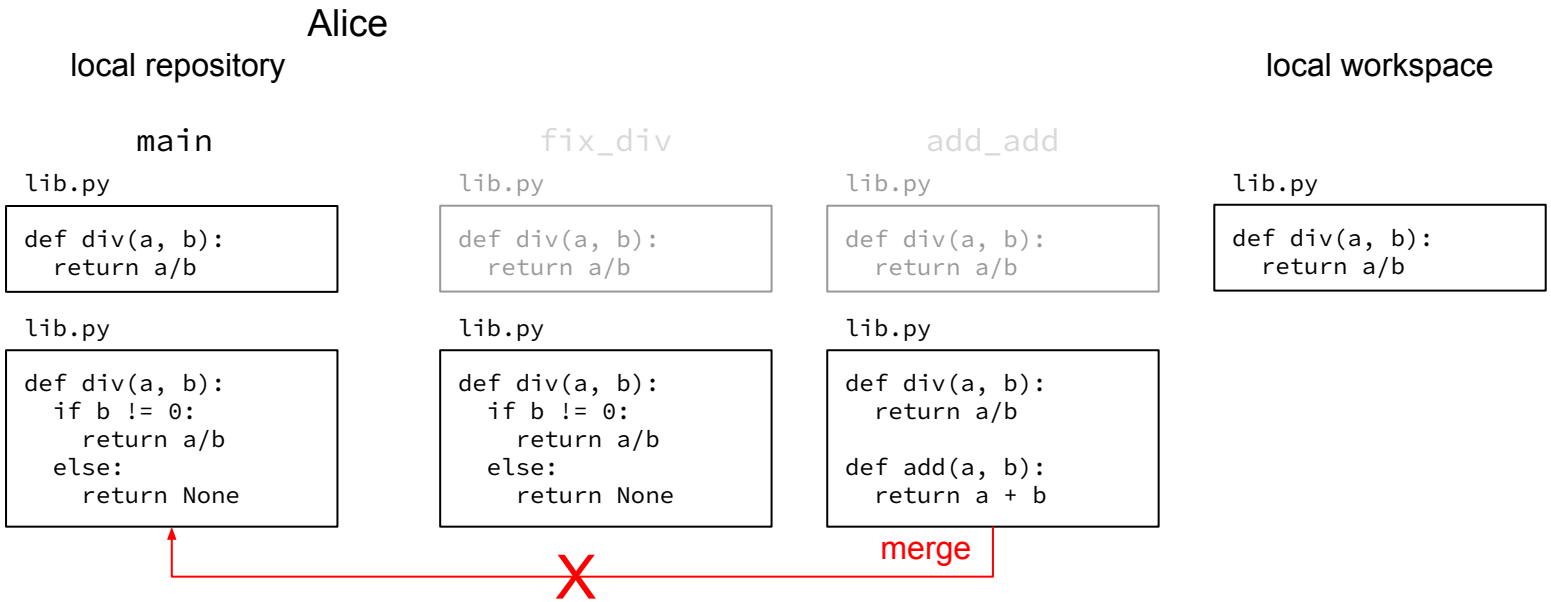
lib.py

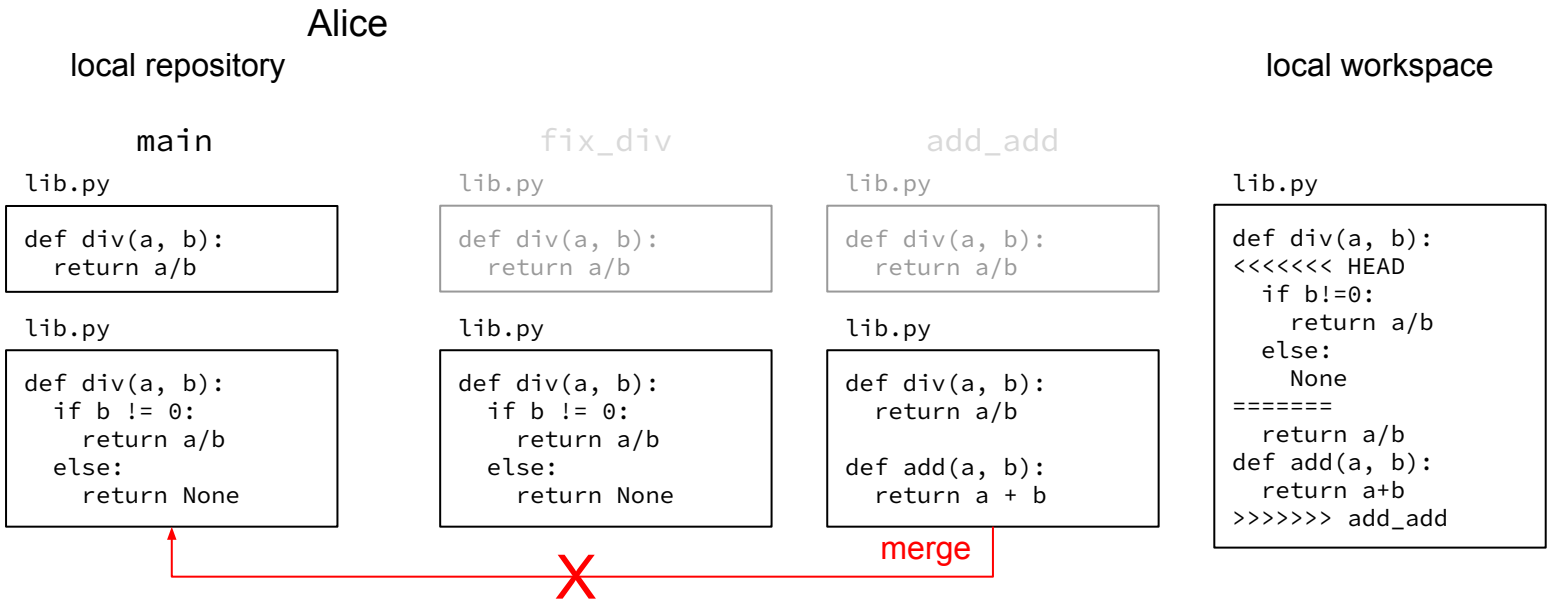
```
def div(a, b):
  return a/b

def add(a, b):
  return a + b
```

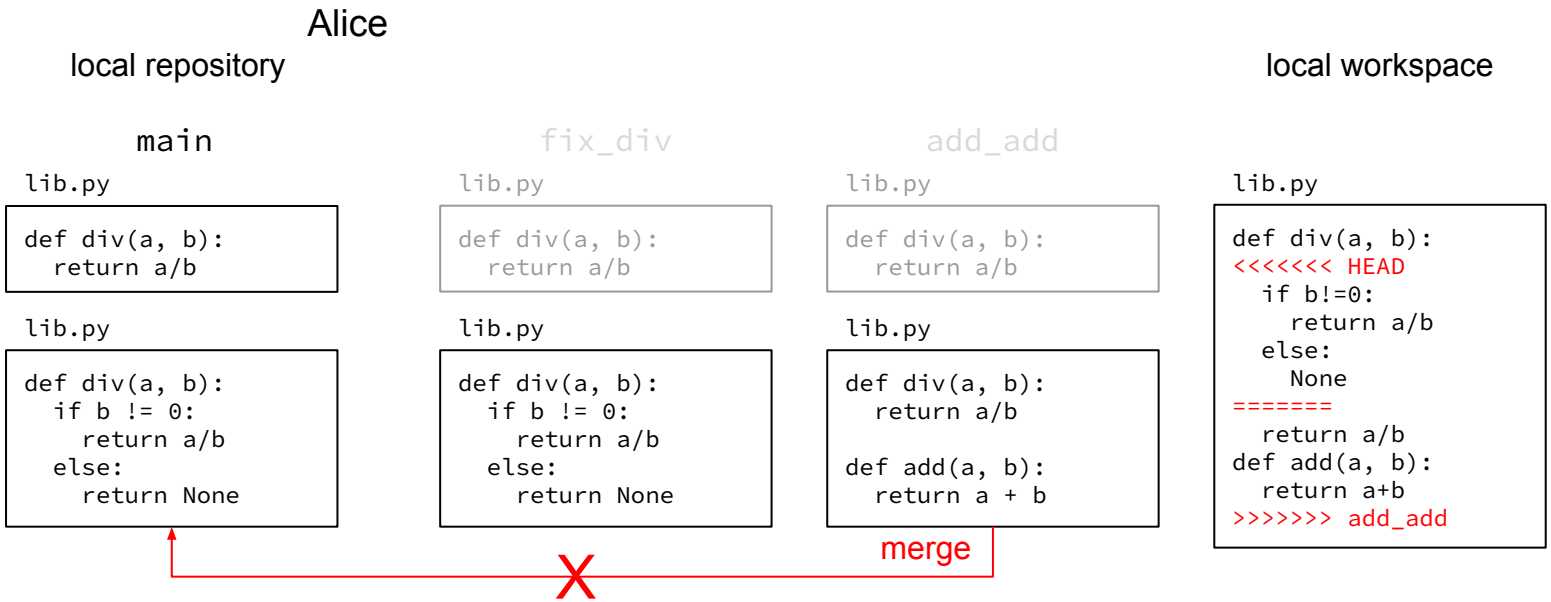
lib.py

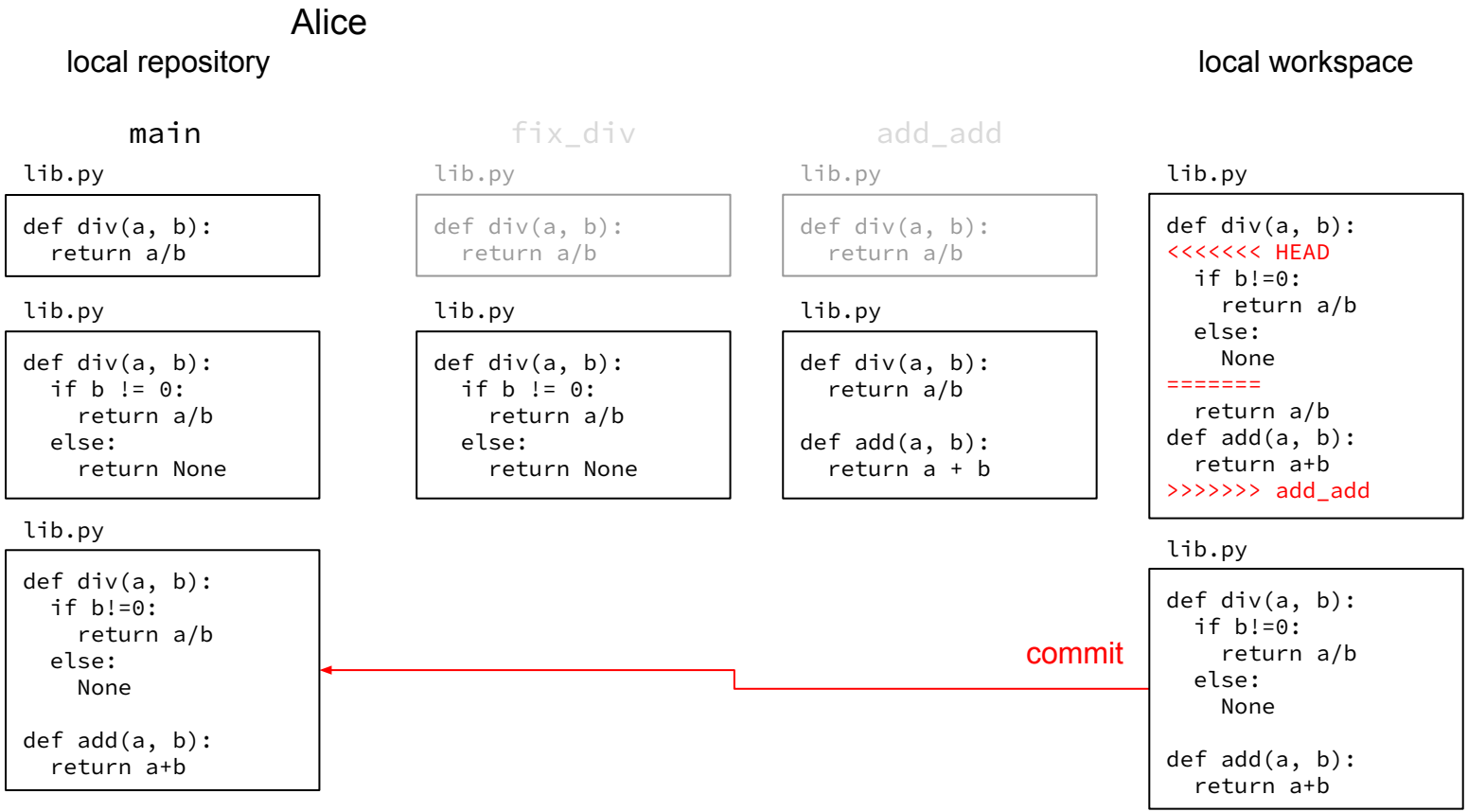
```
def div(a, b):
  return a/b
```


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
merge
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + b


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + b
merge


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + b
merge
lib.py
def div(a, b):
<<<<<<< HEAD
if b!=0:
return a/b
else:
None
=======
return a/b
def add(a, b):
return a+b
>>>>>>> add_add


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + b
merge
lib.py
def div(a, b):
<<<<<<< HEAD
if b!=0:
return a/b
else:
None
=======
return a/b
def add(a, b):
return a+b
>>>>>>> add_add


Alice
local repository
local workspace
main
fix_div
add_add
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
if b!=0:
return a/b
else:
None
def add(a, b):
return a+b
lib.py
def div(a, b):
return a/b
lib.py
def div(a, b):
if b != 0:
return a/b
else:
return None
lib.py
def div(a, b):
return a/b
def add(a, b):
return a + b
lib.py
def div(a, b):
<<<<<<< HEAD
if b!=0:
return a/b
else:
None
=======
return a/b
def add(a, b):
return a+b
>>>>>>> add_add
lib.py
def div(a, b):
if b!=0:
return a/b
else:
None
def add(a, b):
return a+b
commit