Version Control, Git, and GitHub
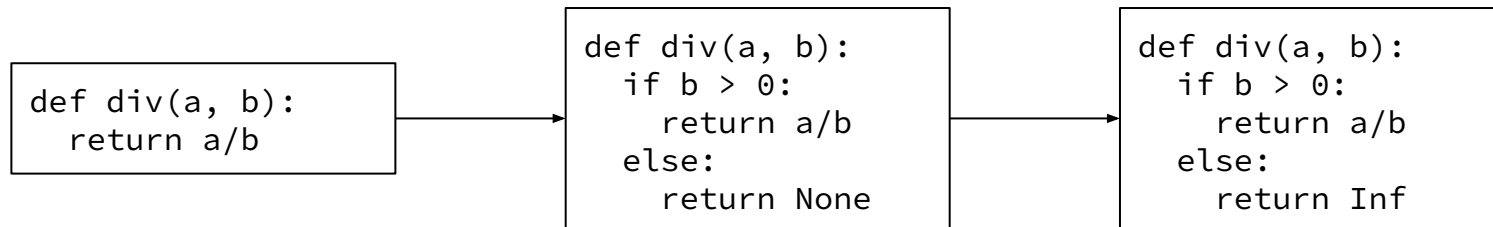
Software Engineering for Scientists

Version control manages changes to files for a project

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

→

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

→

```
def div(a, b):
  if b > 0:
    return a/b
  else:
    return Inf
```

- reverting back to an old version
- allowing developers to test changes without losing the original
- synchronizing code between developers and users

Alex

```
math.py
v1
```
→
```
math.py
v2
```

Betty

```
calc.py
v1
```

Jordan

```
UI.py
v1
```
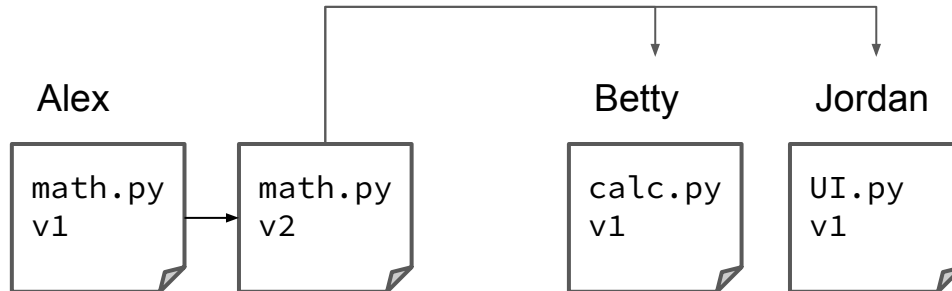
- tagging specific versions

Version control software

Software repository hosting company

git workflow

remote repository   local repository   local workspace

clone

commit

push

pull

git workflow

	remote repository	local repository	local workspace
clone			
commit			
push			
pull			

github

remote repository

lib.py

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

github

remote repository

`lib.py`

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

clone

Alice

local repository

`lib.py`

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

github

remote repository

`lib.py`

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

clone

Alice

local repository

`lib.py`

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

local workspace

`lib.py`

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

github

remote repository

`lib.py`

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

clone

Alice

local repository

`lib.py`

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

local workspace

`lib.py`

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

github

remote repository

`lib.py`

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

clone

Alice

local repository

`lib.py`

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

`lib.py`

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

local workspace

`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
```

clone

clone

Alice

local repository

`lib.py`

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

`lib.py`

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

local workspace

`lib.py`

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

commit

Bob

local repository

`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

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

commit

github

remote repository

`lib.py`

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

`lib.py`

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

clone

clone

push

Alice

local repository

`lib.py`

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

`lib.py`

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

local workspace

`lib.py`

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

commit

Bob

local repository

`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

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

commit

github

remote repository

`lib.py`

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

`lib.py`

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

clone

clone

Alice

local repository

`lib.py`

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

`lib.py`

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

local workspace

`lib.py`

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

push

commit

Bob

local repository

`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

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

commit

push

X

github

remote repository

lib.py

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

lib.py

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

push

clone

clone

Alice

Bob

local repository

local workspace

local repository

local workspace

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

lib.py

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

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

commit

pull

lib.py

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

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

merge

github

remote repository

`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

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

clone

clone

push

Alice

local repository

`lib.py`

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

`lib.py`

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

local workspace

`lib.py`

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

commit

Bob

local repository

`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
```
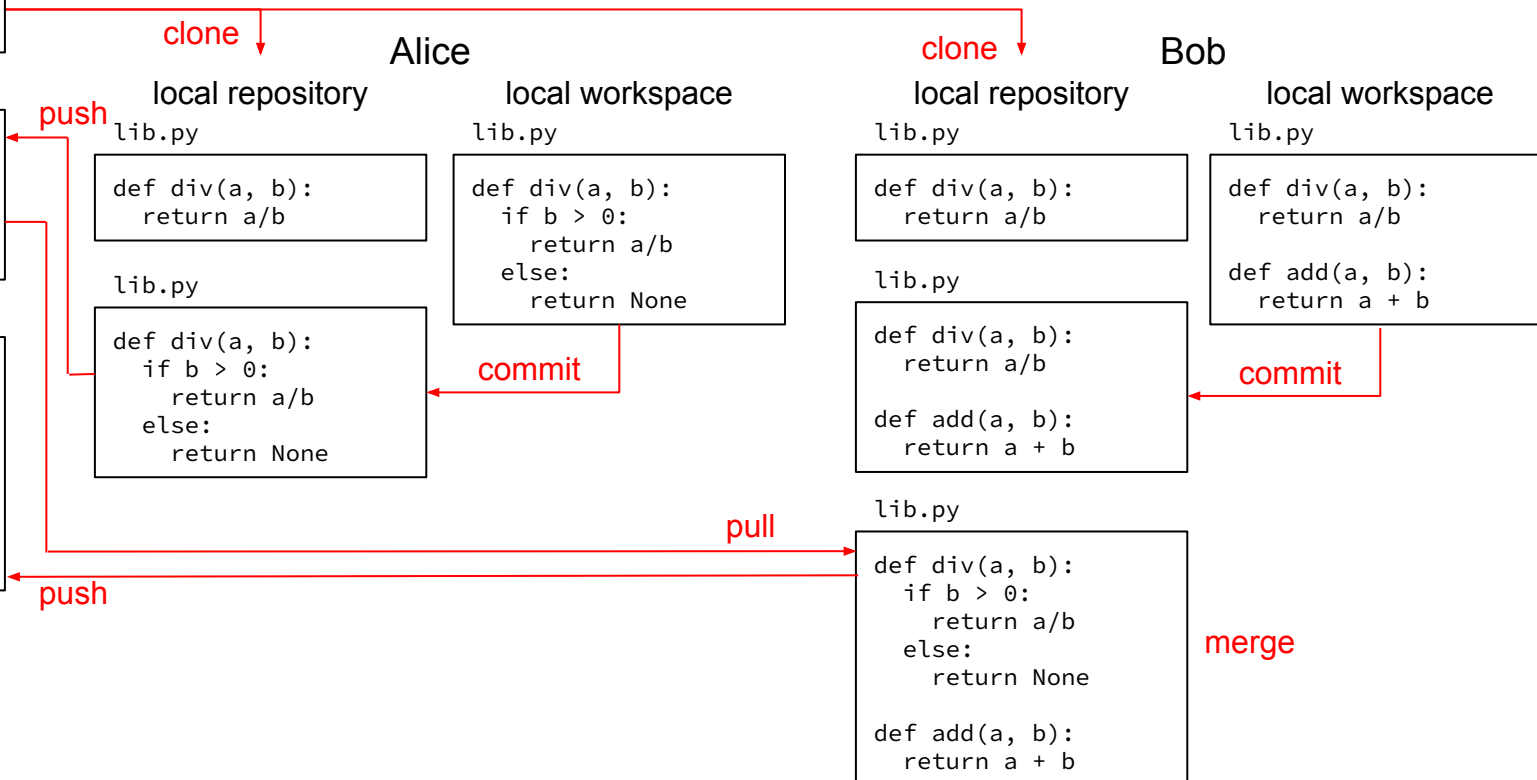
local workspace

`lib.py`

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

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

commit

pull

push

`lib.py`

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

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

merge

github

remote repository

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

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

`lib.py` commit history

github

remote repository

```
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

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

```
setup.py
```

```
viz.py
```

```
run.py
```

repo commit history

github

remote repository

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

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

setup.py

viz.py

run.py

`$ git clone repo` grabs the most recent commit for every file

github

remote repository

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

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

setup.py

viz.py

run.py

Recent commits
cause stability
issues

github

remote repository

`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

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

`setup.py`

`viz.py`

`run.py`

`v1.0`

Tag the commit state of the repo to provide consistency as development continues

`$ git clone repo --branch v1.0`

github

remote repository

`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

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

`setup.py`

`viz.py`

`run.py`

`$ git clone repo --branch v1.0`