Design Studio: Information and Visualization

4.032 / 4.033

PROGRAMMING PART

Irene de la Torre – Arenas

Email: delatorr@mit.edu

Office Hours: Mondays, 5:00 – 7:00 at CDC

Goal of the class

Beyond acquiring technological skills, learn fundamentals concepts
of information design and data visualization and develop the strategies to communicate
different types of information and data.

Goal of the class

Beyond acquiring technological skills, learn fundamentals concepts
of information design and data visualization and develop the strategies to communicate
different types of information and data.

Synthetize your design skills in a data visualization project

Goal of the class

Beyond acquiring technological skills, learn fundamentals concepts of information design and data visualization and develop the strategies to communicate different types of information and data.

Synthetize your design skills in a data visualization project

At the end of the course, you should be able to plan, conceptualize, develop and refine a data visualization project of any type.

Remember…

The goal of information design and, therefore, data visualization is to explain information. Sometimes it can try to persuade. But it will never try to obscure.

Remember…

The goal of information design and, therefore, data visualization is to explain information. Sometimes it can try to persuade. But it will never try to obscure.

And if it's doing it. Then, it's another thing…

Project 1 — Visualizing time

Sketch and develop 3 displays of time

Project 1 — Visualizing time

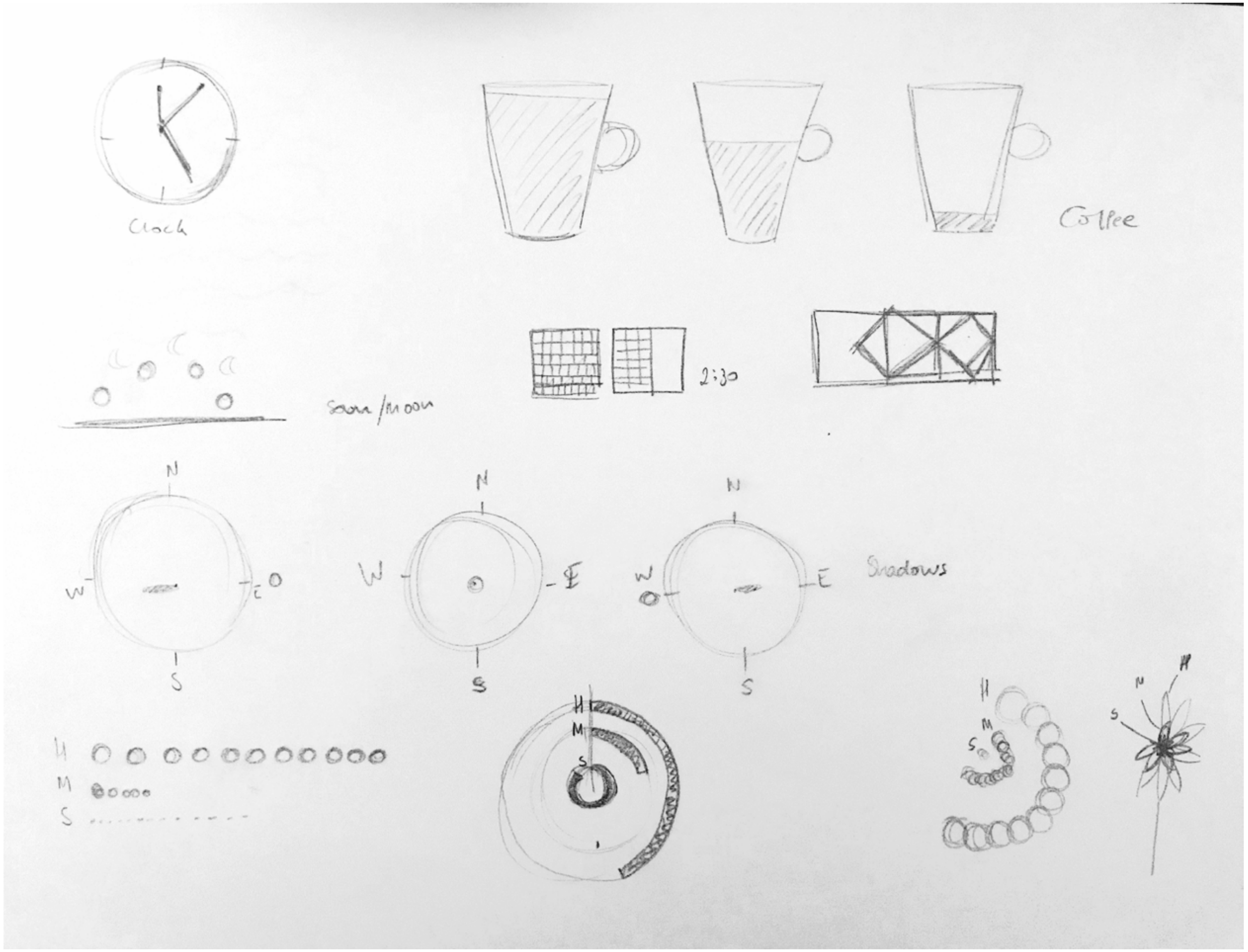
Sketch and develop 3 displays of time

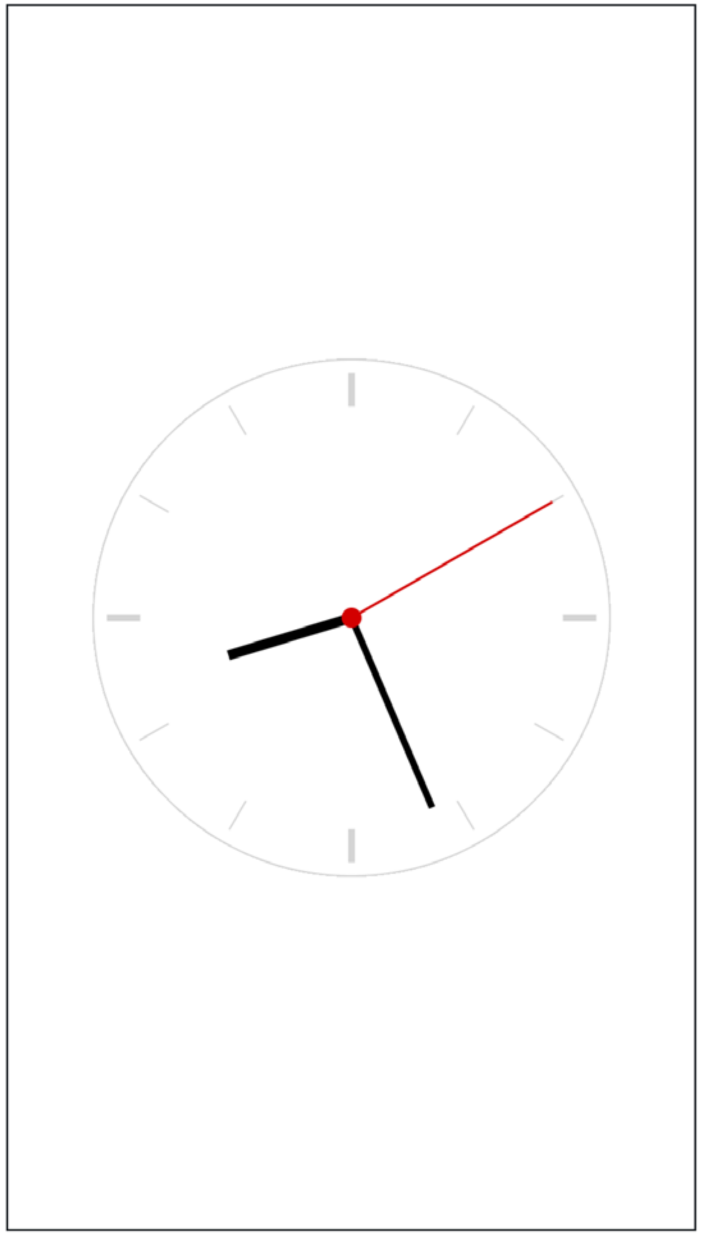
PROCESS

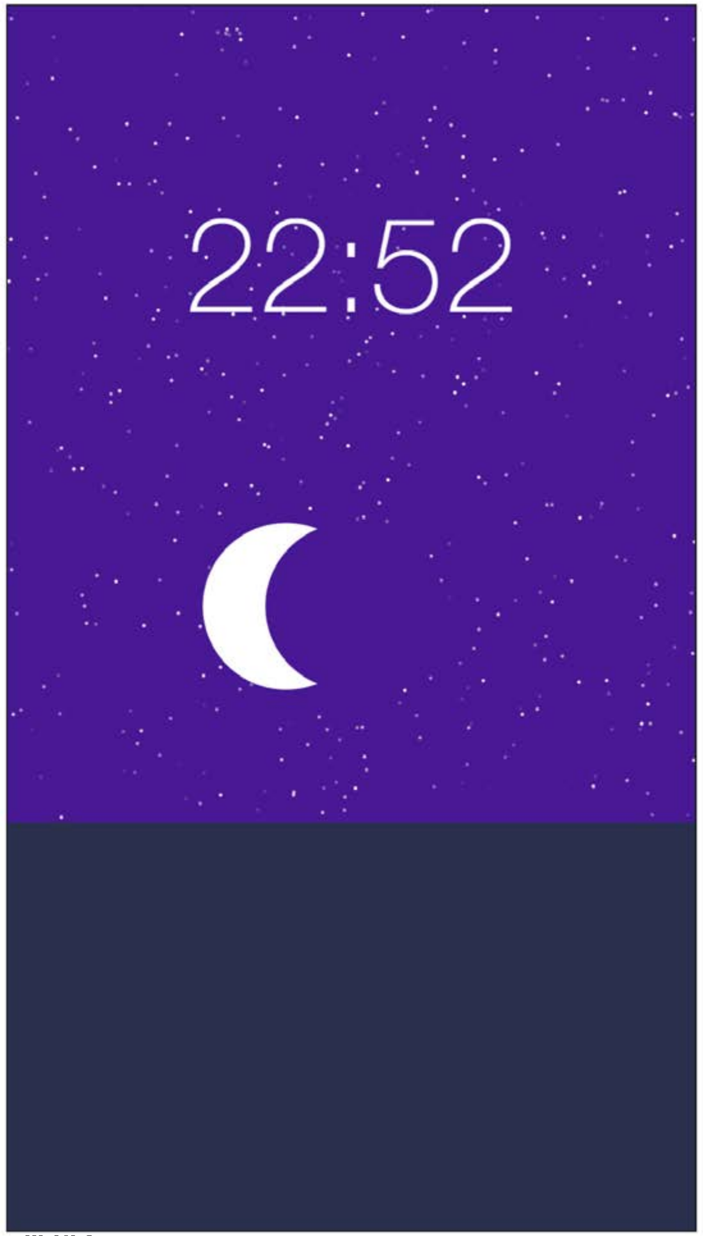
Work individually.

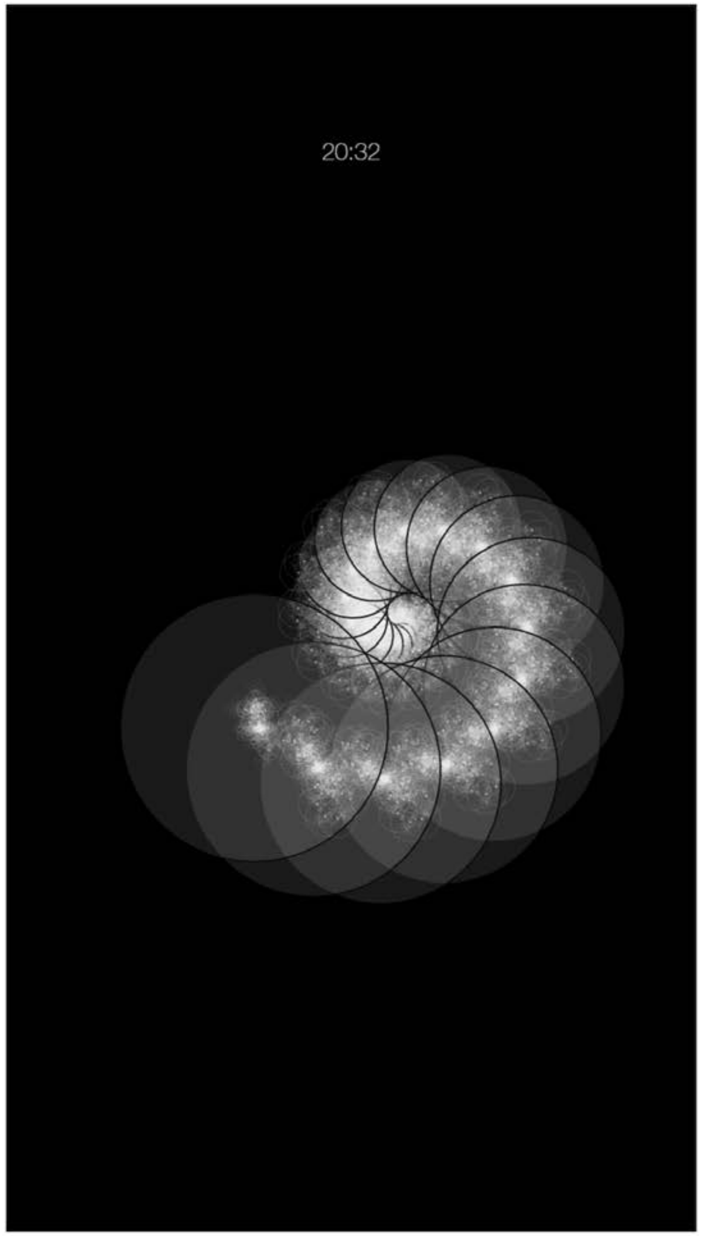
Create twenty sketches on paper, then select 3 to pursue.

Draw them with JavaScript.


Clock
Coffee
Sun/Moon
2:30
N
W
E
S
Shadows
H
M
S




22:52


20:32

Project 1 — Visualizing time

REQUIREMENTS

- Download and install Git, and WebStorm / Sublime Text / Brackets

Project 1 — Visualizing time

REQUIREMENTS

- Download and install Git, and WebStorm / Sublime Text / Brackets
- Sign up for an account on GitHub.mit.edu

Project 1 — Visualizing time

REQUIREMENTS

- Download and install Git, and WebStorm / Sublime Text / Brackets
- Sign up for an account on GitHub.mit.edu
- Use the GitHub repository MIT-Information-Design-and-Visualization . Clone the files on your own computer and work with the files.

Project 1 — Visualizing time

REQUIREMENTS

- Download and install Git, and WebStorm / Sublime Text / Brackets
- Sign up for an account on GitHub.mit.edu
- Use the GitHub repository MIT-Information-Design-and-Visualization . Clone the files on your own computer and work with them.
- Complete exercise by uploading it on GitHub by Feb 14

Project 1 — Visualizing time

REQUIREMENTS

- Use HTML, CSS and JavaScript. You can choose how to draw the sketches: using canvas, d3.js, P5.js, etc.

Project 1 — Visualizing time

REQUIREMENTS

- Use HTML, CSS and JavaScript. You can choose how to draw the sketches: using canvas, d3.js, P5.js, etc.
- The visualization should be optimized for viewing on a mobile device: approximately 414 x 736 pixels.

Setting up GitHub

1. Create a GitHub Account

Setting up GitHub

1. Create a GitHub Account

2. Download GitHub Desktop at https://desktop.github.com/
   You will manage all your repositories and files through this dashboard.

Setting up GitHub

1. Create a GitHub Account

2. Download GitHub Desktop at https://desktop.github.com/
   You will manage all your repositories and files through this dashboard.

3. Go to the class' repository
   https://github.com/irenedelatorre/MIT-Design-Studio-Information-and-Visualization

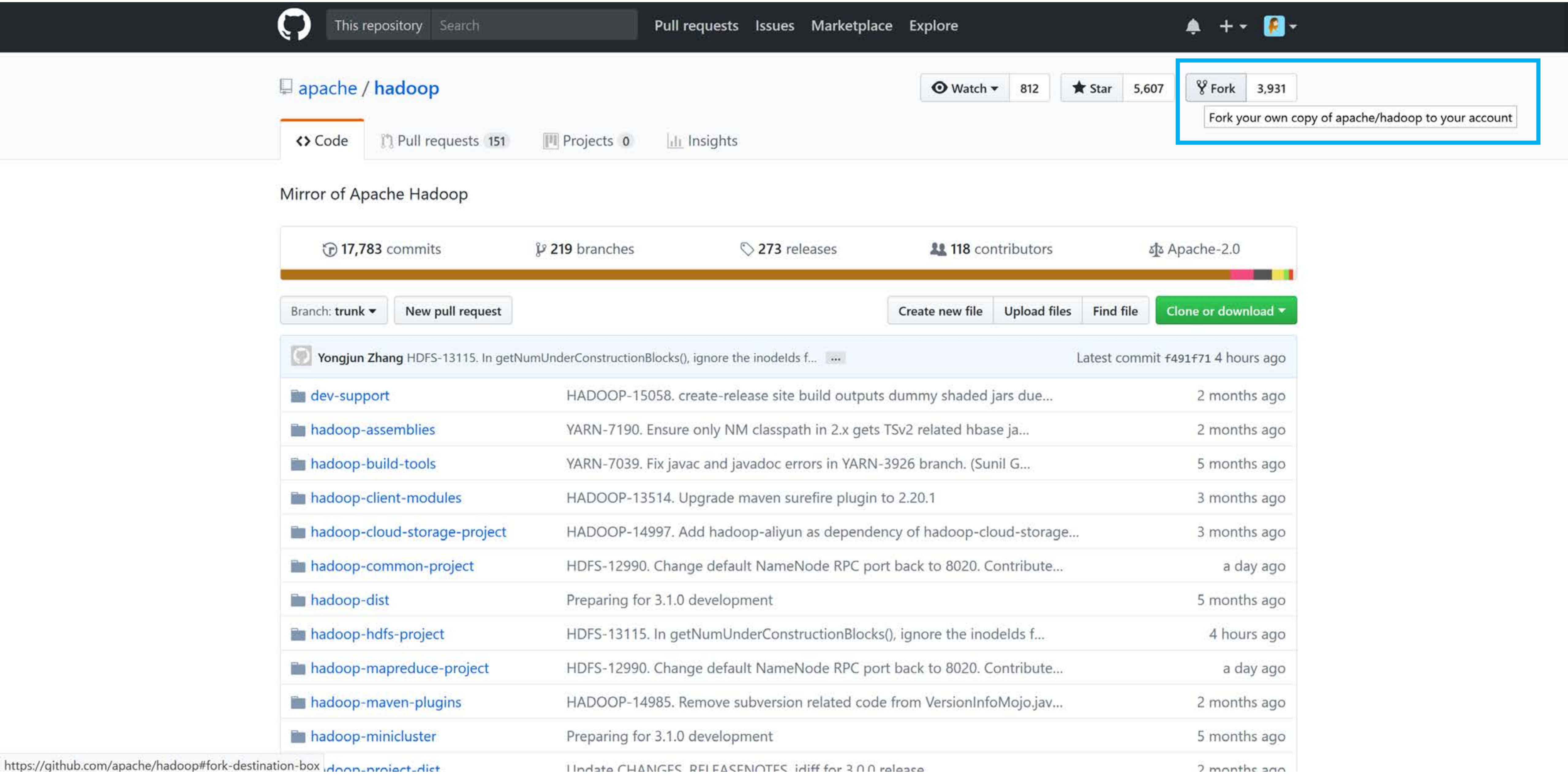
Setting up GitHub

1. Create a GitHub Account

2. Download GitHub Desktop at https://desktop.github.com/
   You will manage all your repositories and files through this dashboard.

3. Go to the class' repository
   https://github.com/irenedelatorre/MIT-Design-Studio-Information-and-Visualization

4. Fork the repository.
   It will copy the files in your own profile



apache / hadoop

Watch 812 | Star 5,607 | Fork 3,931

Fork your own copy of apache/hadoop to your account

Mirror of Apache Hadoop

17,783 commits | 219 branches | 273 releases | 118 contributors | Apache-2.0

Yongjun Zhang HDFS-13115. In getNumUnderConstructionBlocks(), ignore the inodeIds f... … Latest commit `f491f71` 4 hours ago

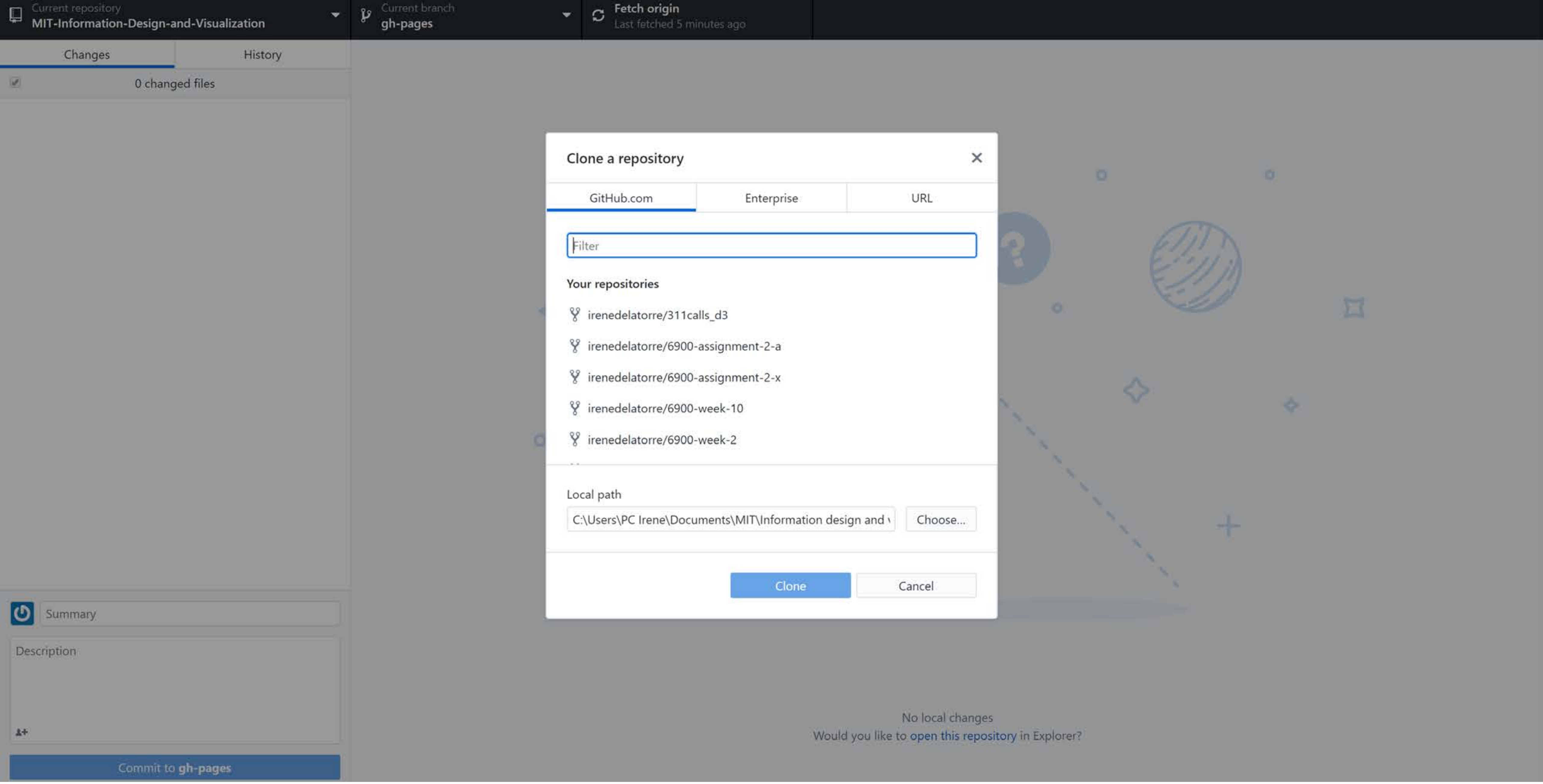
dev-support	HADOOP-15058. create-release site build outputs dummy shaded jars due...	2 months ago
hadoop-assemblies	YARN-7190. Ensure only NM classpath in 2.x gets TSv2 related hbase ja...	2 months ago
hadoop-build-tools	YARN-7039. Fix javac and javadoc errors in YARN-3926 branch. (Sunil G...	5 months ago
hadoop-client-modules	HADOOP-13514. Upgrade maven surefire plugin to 2.20.1	3 months ago
hadoop-cloud-storage-project	HADOOP-14997. Add hadoop-aliyun as dependency of hadoop-cloud-storage...	3 months ago
hadoop-common-project	HDFS-12990. Change default NameNode RPC port back to 8020. Contribute...	a day ago
hadoop-dist	Preparing for 3.1.0 development	5 months ago
hadoop-hdfs-project	HDFS-13115. In getNumUnderConstructionBlocks(), ignore the inodeIds f...	4 hours ago
hadoop-mapreduce-project	HDFS-12990. Change default NameNode RPC port back to 8020. Contribute...	a day ago
hadoop-maven-plugins	HADOOP-14985. Remove subversion related code from VersionInfoMojo.jav...	2 months ago
hadoop-minicluster	Preparing for 3.1.0 development	5 months ago
doop-project-dist	Update CHANGES, RELEASENOTES, jdiff for 3.0.0 release	2 months ago

https://github.com/apache/hadoop#fork-destination-box

Setting up GitHub

1. Go to GitHub Desktop, log in with your user, and

    - Go to File - Clone Repository
    - Search for yourUserName/MIT-Design-Studio-Information-and-Visualization
      It will copy all the files of your online repository to your computer



Clone a repository

Your repositories

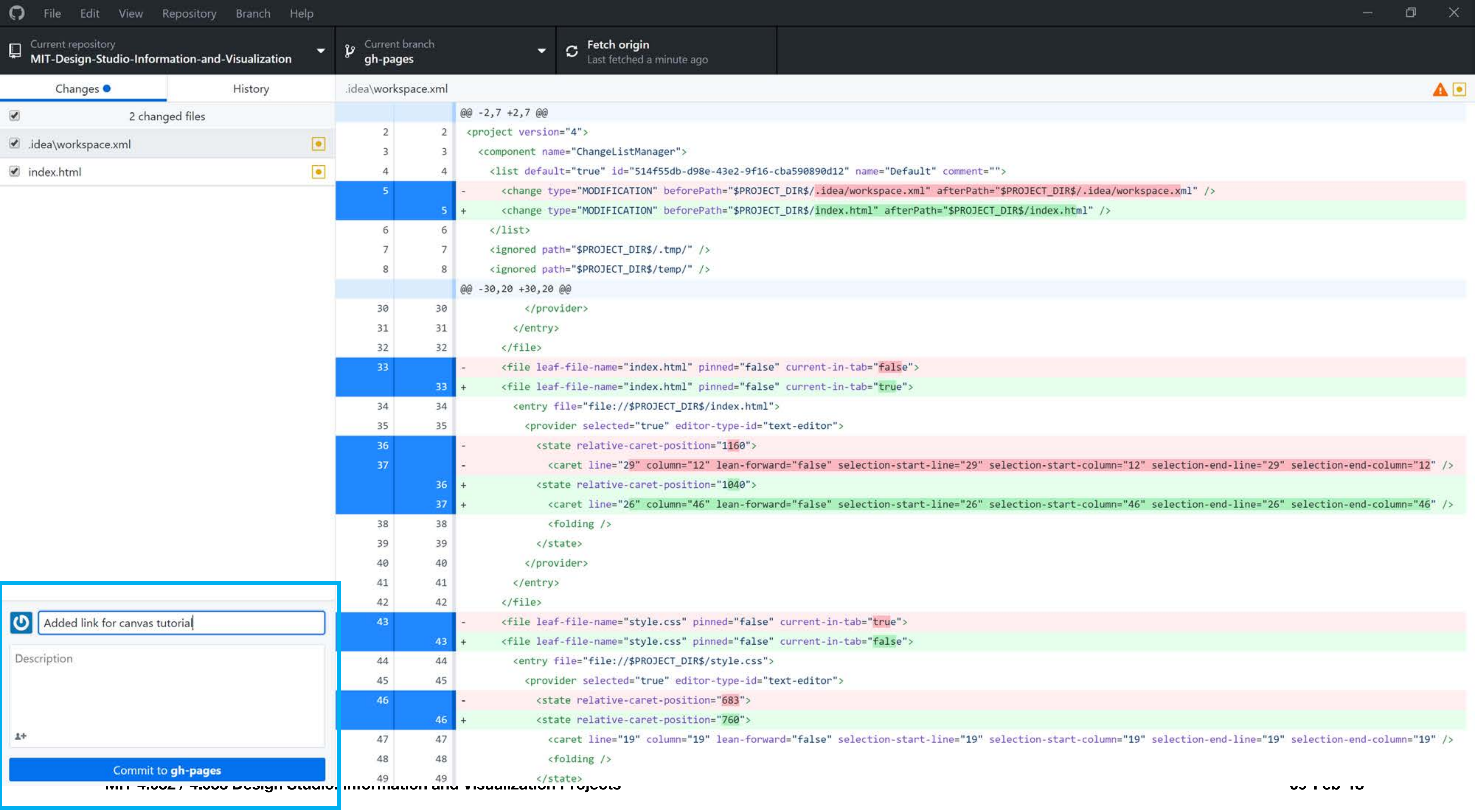
- irenedelatorre/311calls_d3
- irenedelatorre/6900-assignment-2-a
- irenedelatorre/6900-assignment-2-x
- irenedelatorre/6900-week-10
- irenedelatorre/6900-week-2

Local path: C:\Users\PC Irene\Documents\MIT\Information design and \

No local changes
Would you like to open this repository in Explorer?

Setting up GitHub

1. Go to GitHub Desktop, log in with your user, and
   - Go to File - Clone Repository
   - Search for yourUserName/MIT-Design-Studio-Information-and-Visualization
     It will copy all the files of your online repository to your computer
2. Once you have worked with the files, and modified them, GitHub Desktop will keep track of your changes. In order for you to save them in the online repository, you need to commit those changes.



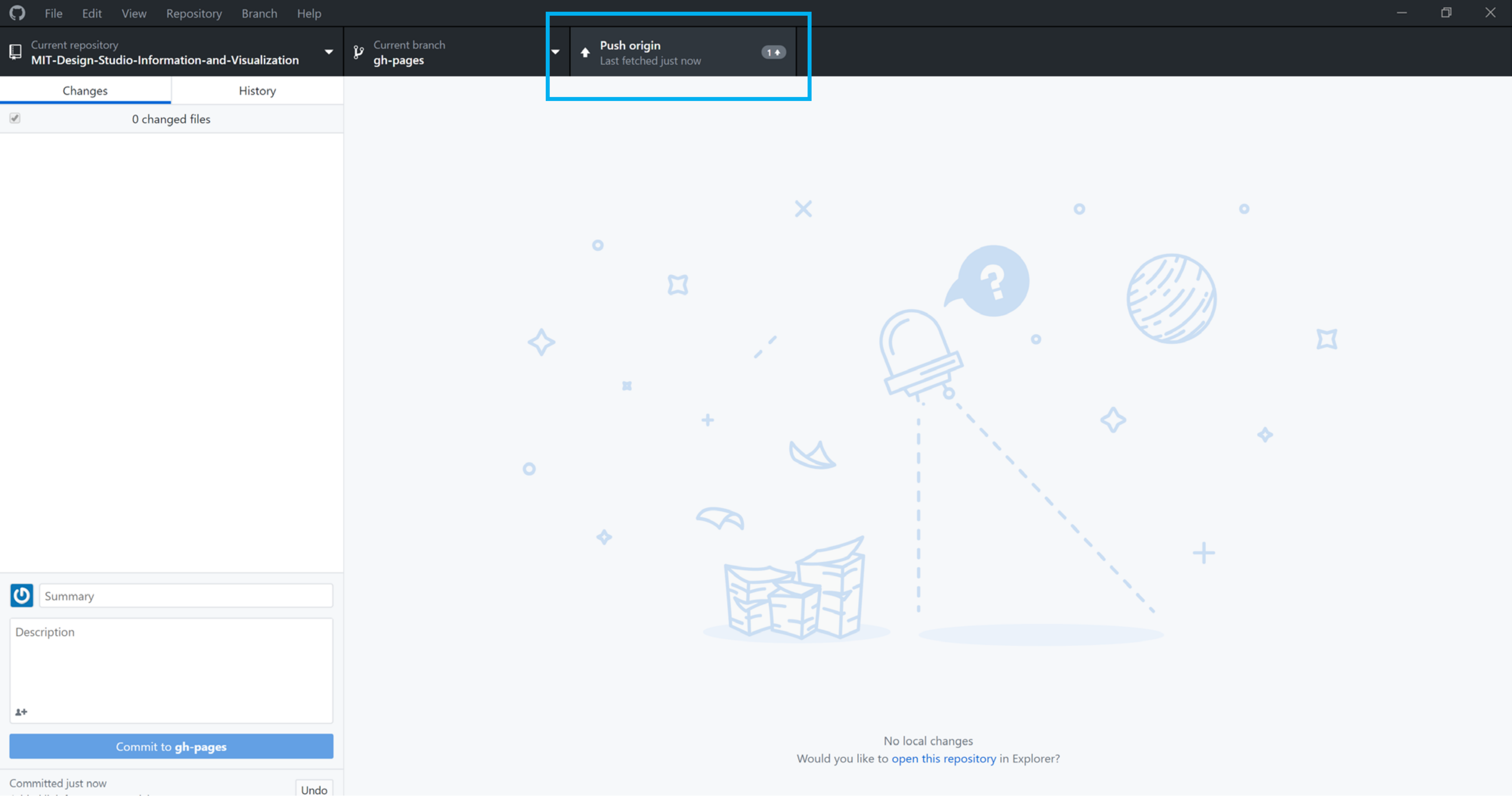
Added link for canvas tutorial

.idea\workspace.xml

```
@@ -2,7 +2,7 @@
 <project version="4">
   <component name="ChangeListManager">
     <list default="true" id="514f55db-d98e-43e2-9f16-cba590890d12" name="Default" comment="">
-      <change type="MODIFICATION" beforePath="$PROJECT_DIR$/.idea/workspace.xml" afterPath="$PROJECT_DIR$/.idea/workspace.xml" />
+      <change type="MODIFICATION" beforePath="$PROJECT_DIR$/index.html" afterPath="$PROJECT_DIR$/index.html" />
     </list>
     <ignored path="$PROJECT_DIR$/.tmp/" />
     <ignored path="$PROJECT_DIR$/temp/" />
@@ -30,20 +30,20 @@
           </provider>
         </entry>
       </file>
-      <file leaf-file-name="index.html" pinned="false" current-in-tab="false">
+      <file leaf-file-name="index.html" pinned="false" current-in-tab="true">
         <entry file="file://$PROJECT_DIR$/index.html">
           <provider selected="true" editor-type-id="text-editor">
-            <state relative-caret-position="1160">
-              <caret line="29" column="12" lean-forward="false" selection-start-line="29" selection-start-column="12" selection-end-line="29" selection-end-column="12" />
+            <state relative-caret-position="1040">
+              <caret line="26" column="46" lean-forward="false" selection-start-line="26" selection-start-column="46" selection-end-line="26" selection-end-column="46" />
               <folding />
             </state>
           </provider>
         </entry>
       </file>
-      <file leaf-file-name="style.css" pinned="false" current-in-tab="true">
+      <file leaf-file-name="style.css" pinned="false" current-in-tab="false">
         <entry file="file://$PROJECT_DIR$/style.css">
           <provider selected="true" editor-type-id="text-editor">
-            <state relative-caret-position="683">
+            <state relative-caret-position="760">
               <caret line="19" column="19" lean-forward="false" selection-start-line="19" selection-start-column="19" selection-end-line="19" selection-end-column="19" />
               <folding />
             </state>
```

Setting up GitHub

1. However, with committing the changes is not enough. After that, you will need to push the changes. Only then the changes will appear in your online repository.



0 changed files

Commit to gh-pages

Committed just now

No local changes
Would you like to open this repository in Explorer?

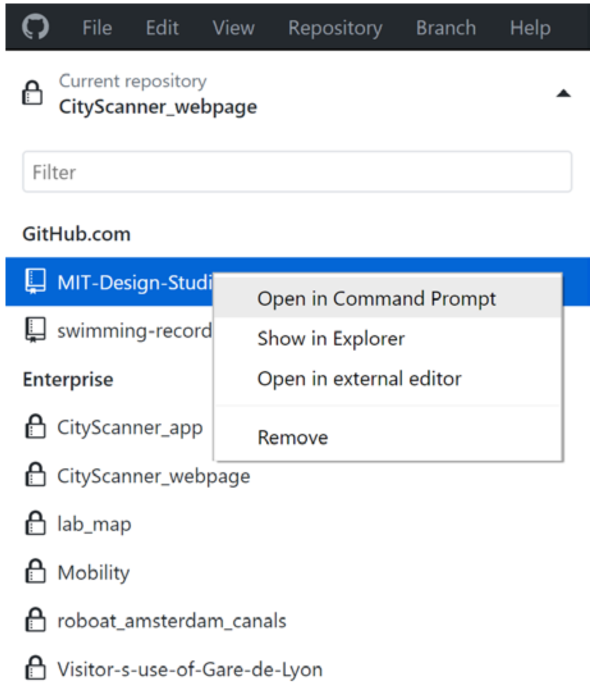
Setting up GitHub

1. However, with committing the changes is not enough. After that, you will need to push the changes. Only then the changes will appear in your online repository.

2. GitHub also gives you the possibility of creating branches, where the code might be different – imagine that you want to test some code without losing the original one. Later, you can merge those branches together.

Updating your repository from the original fork

1. You can do this either by installing Git Bash or by right clicking your repository and selecting Open in Command Prompt


File
Edit
View
Repository
Branch
Help
Current repository
CityScanner_webpage
Filter
GitHub.com
MIT-Design-Studi
swimming-record
Enterprise
CityScanner_app
CityScanner_webpage
lab_map
Mobility
roboat_amsterdam_canals
Visitor-s-use-of-Gare-de-Lyon
Open in Command Prompt
Show in Explorer
Open in external editor
Remove

Updating your repository from the original fork

1. You can do this either by installing Git Bash or by right clicking your repository and selecting Open in Command Prompt

- Change the current working directory to your local project.

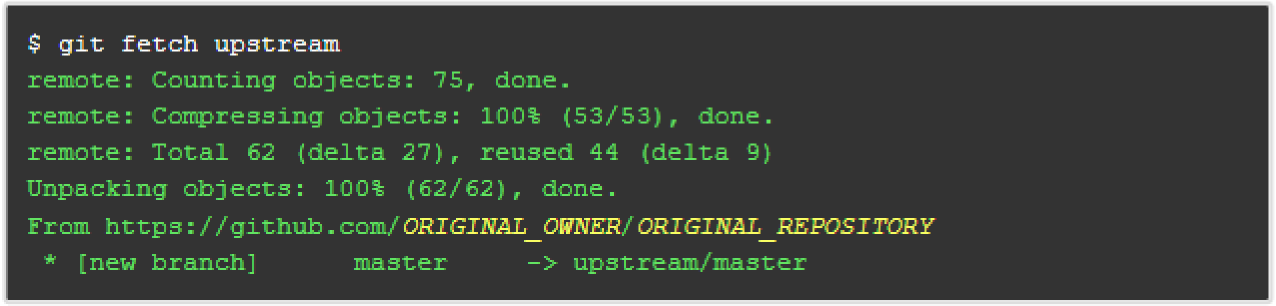
Updating your repository from the original fork

1. You can do this either by installing Git Bash or by right clicking your repository and selecting Open in Command Prompt

2. Change the current working directory to your local project.

3. Fetch the branches and their respective commits from the upstream repository. Commits to master will be stored in a local branch, upstream/master.

   Write git fetch upstream



```
$ git fetch upstream
remote: Counting objects: 75, done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 62 (delta 27), reused 44 (delta 9)
Unpacking objects: 100% (62/62), done.
From https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY
 * [new branch]      master     -> upstream/master
```

Updating your repository from the original fork

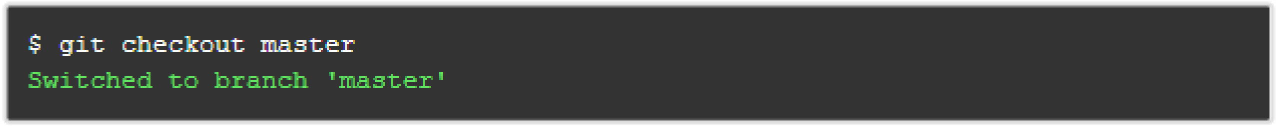
1. You can do this either by installing Git Bash or by right clicking your repository and selecting Open in Command Prompt

2. Change the current working directory to your local project.

3. Fetch the branches and their respective commits from the upstream repository. Commits to master will be stored in a local branch, upstream/master.

   Write `git fetch upstream`

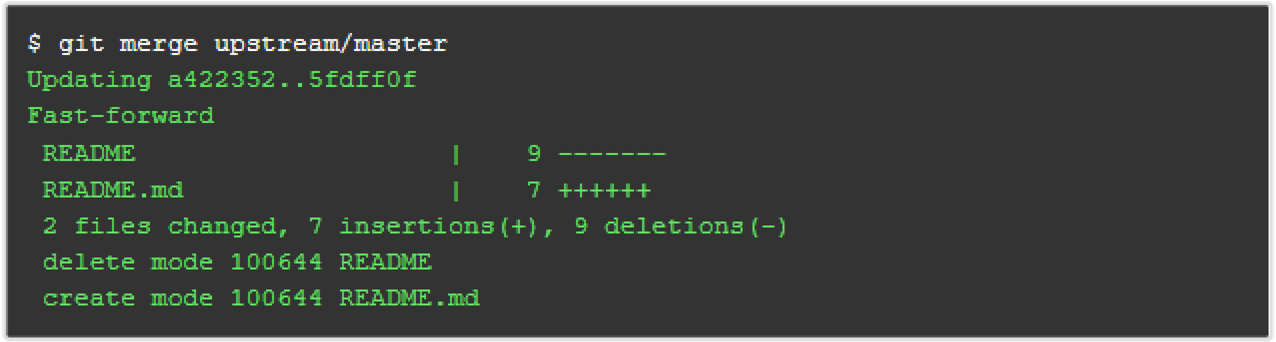
4. Check out your fork's local master branch (or the branch that you want to update).
   Write `git checkout master`



```
$ git checkout master
Switched to branch 'master'
```

Updating your repository from the original fork

1. Merge the changes from upstream/master into your local master branch. This brings your fork's master branch into sync with the upstream repository, without losing your local changes.
   Write git merge upstream/master

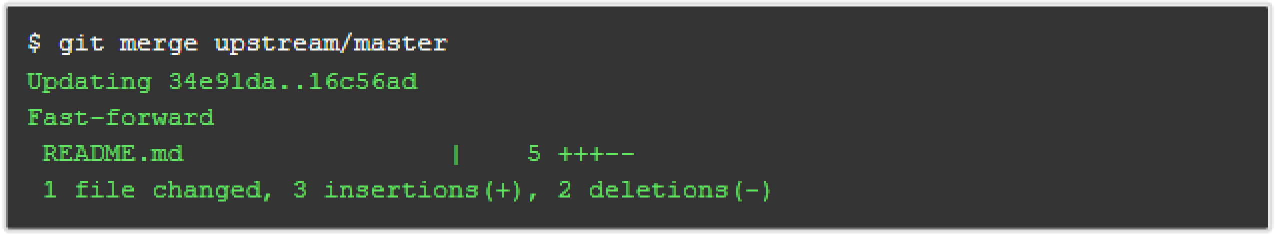


```
$ git merge upstream/master
Updating a422352..5fdff0f
Fast-forward
 README                        |    9 -------
 README.md                     |    7 ++++++
 2 files changed, 7 insertions(+), 9 deletions(-)
 delete mode 100644 README
 create mode 100644 README.md
```

Updating your repository from the original fork

1. Merge the changes from upstream/master into your local master branch. This brings your fork's master branch into sync with the upstream repository, without losing your local changes.
   Write `git merge upstream/master`

2. If your local branch didn't have any unique commits (to be push), Git will instead perform a "fast-forward“.



```
$ git merge upstream/master
Updating 34e91da..16c56ad
Fast-forward
 README.md                 |    5 +++--
 1 file changed, 3 insertions(+), 2 deletions(-)
```

Updating your repository from the original fork

1. Merge the changes from upstream/master into your local master branch. This brings your fork's master branch into sync with the upstream repository, without losing your local changes.
   Write `git merge upstream/master`

2. If your local branch didn't have any unique commits (to be push), Git will instead perform a "fast-forward“.

3. Syncing your fork only updates your local copy of the repository. To update your fork on GitHub, you must push your changes.

More information…

- How to merge branches
    - You can do it using GitHub Desktop
    - Or go to your remote repository, Pull Requests and open a New Pull Request. If there is no conflict between the branches, you will be able to merge them without any problem. If there are conflicts, you will need to solve them by choosing which code from which branch must stay.