

Git/Github, Dash, and Heroku

(IE481, Lab #3 Configuration)

...

Joonyoung Park

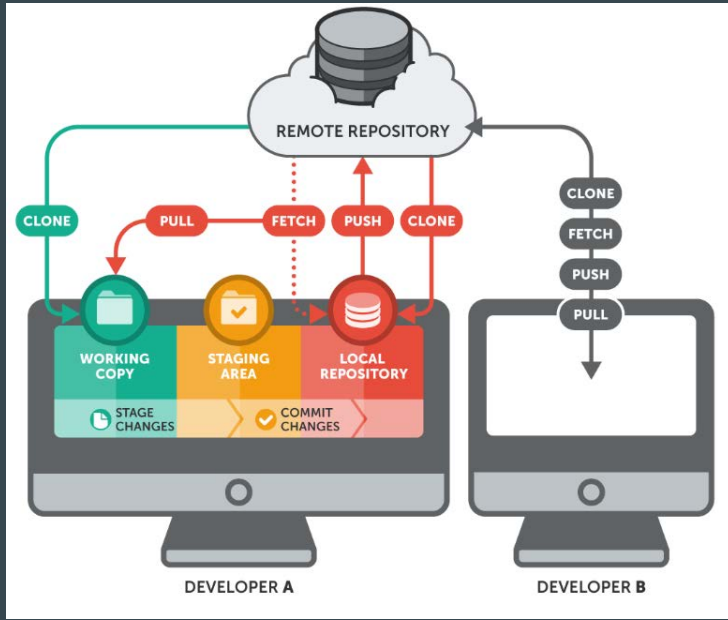
Contents

- Git/Github
 - Install and configure Git
 - Create a Github account
 - Guideline for using Git and Github
- Dash
 - Install and configure Dash
 - Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)
- Heroku
 - Install Heroku and create an account
 - A sample example of web hosting using Heroku
 - Guideline for hosting web services using Python with Heroku

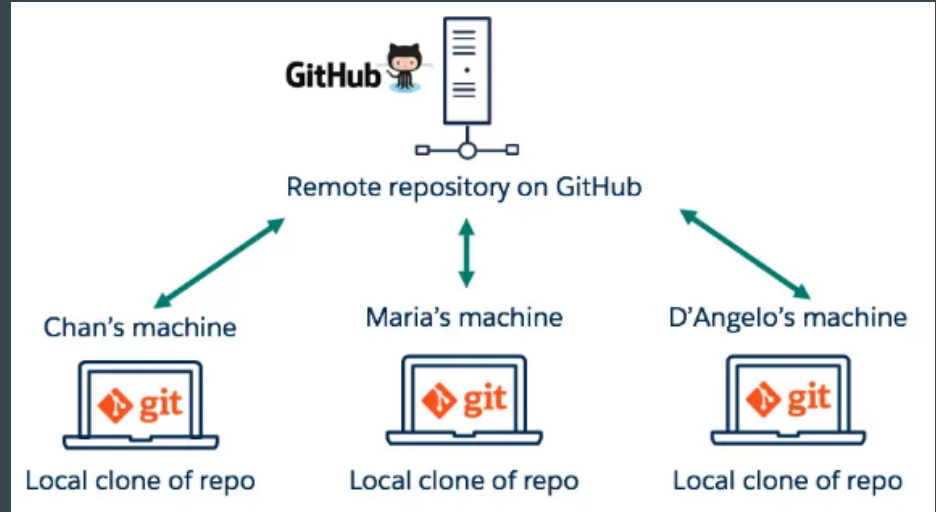
Git/Github

What is Git and Github?

- **Git:** A distributed version control system for tracking and managing history of changes of files
- **Github:** A remote repository to upload and share the files managed on the local machine by Git



An example of Git operation



Relationship between Git and Github

Contents

- Git/Github

- Install and configure Git
- Create a Github account
- Guideline for using Git and Github

- Dash

- Install and configure Dash
- Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)

- Heroku

- Install Heroku and create an account
- A sample example of web hosting using Heroku
- Guideline for hosting web services using Python with Heroku

Install and configure Gt

- Go to the Git homepage: [link](#)
- Click 'Download' (Fig 1.) → Select your OS version (Fig 2.) → Start downloading automatically (Fig 3.)
- In step Fig 3., choose your own path to download executable file



Fig 1. Click 'Download'

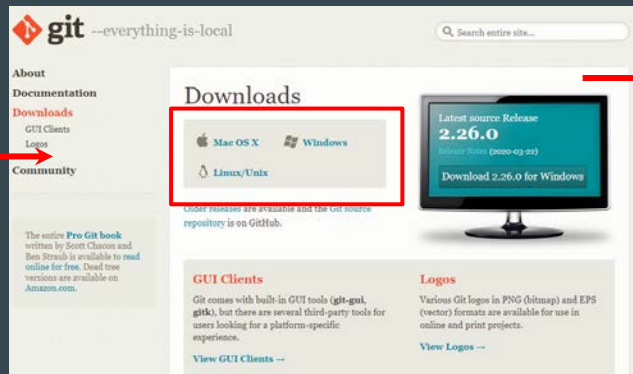


Fig 2. Select OS version

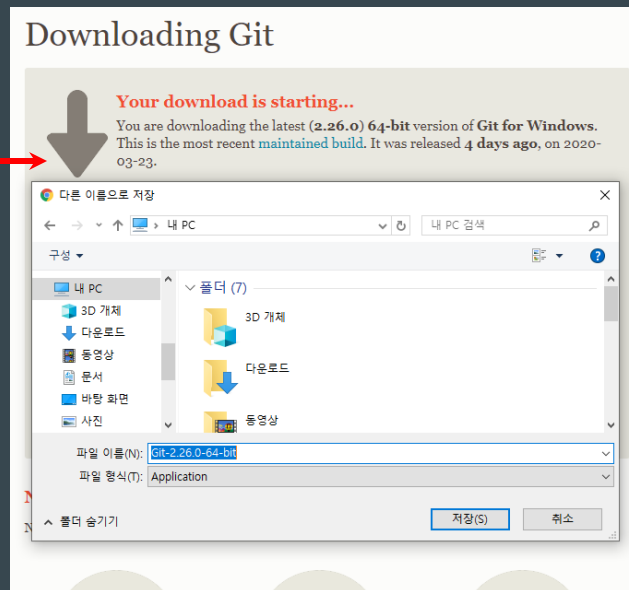


Fig 3. Start downloading automatically

Install and configure Git

- Iteratively click 'Next' and 'Install' button on the process of installation (Fig 4.)
- Uncheck the box of 'view release Notes' (Fig 5.)
- After installation, open a command line and Typing 'git --version' to check if the install completed successfully (Fig 6.)

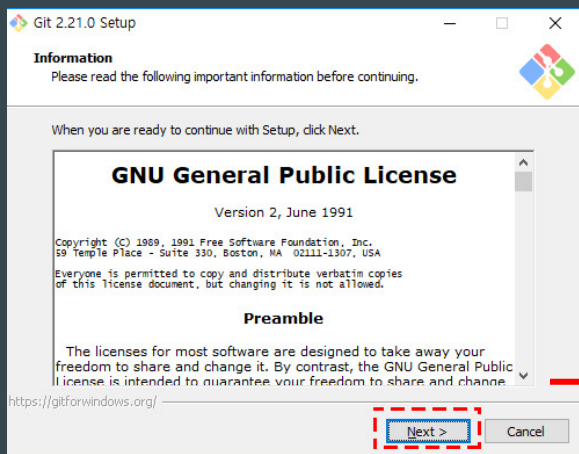


Fig 4. First step to install

...
(Some steps)

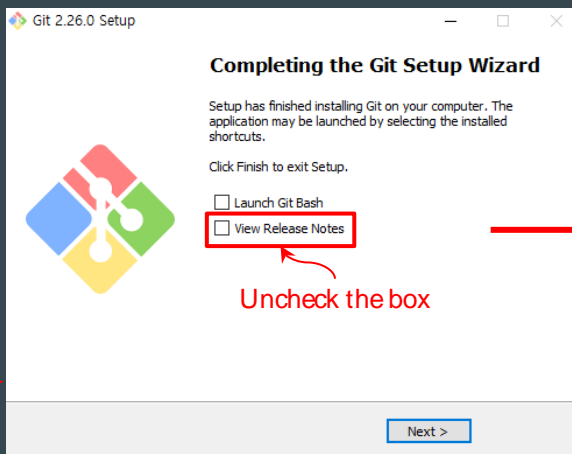


Fig 5. Uncheck the box of 'view release Notes'

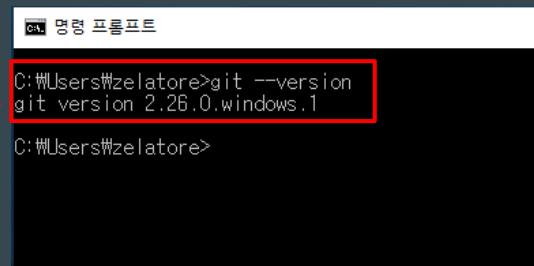


Fig 6. Checking your installation

Contents

- Git/Github

- Install and configure Git
- Create a Github account
- Guideline for using Git and Github

- Dash

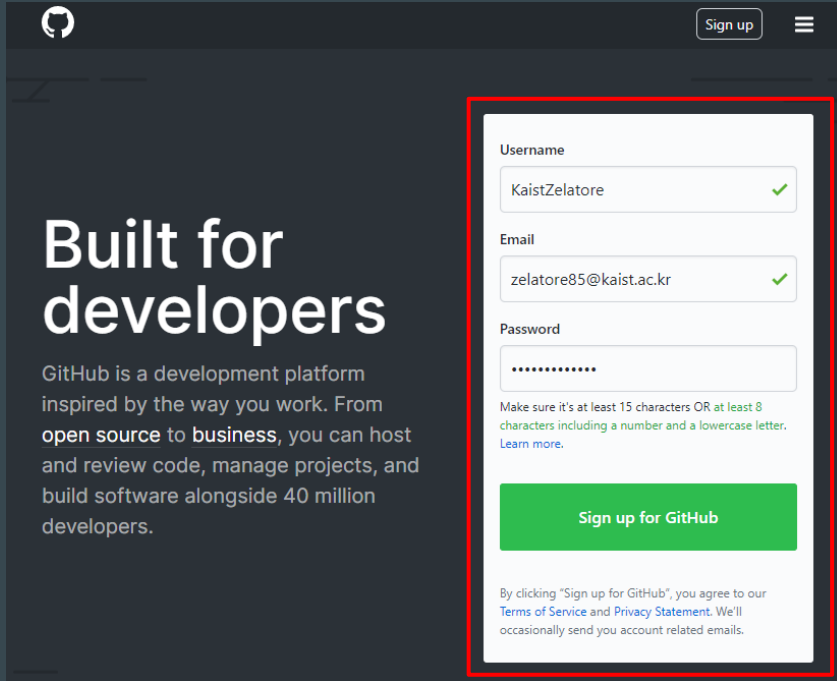
- Install and configure Dash
- Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)

- Heroku

- Install Heroku and create an account
- A sample example of web hosting using Heroku
- Guideline for hosting web services using Python with Heroku

Create a Github account

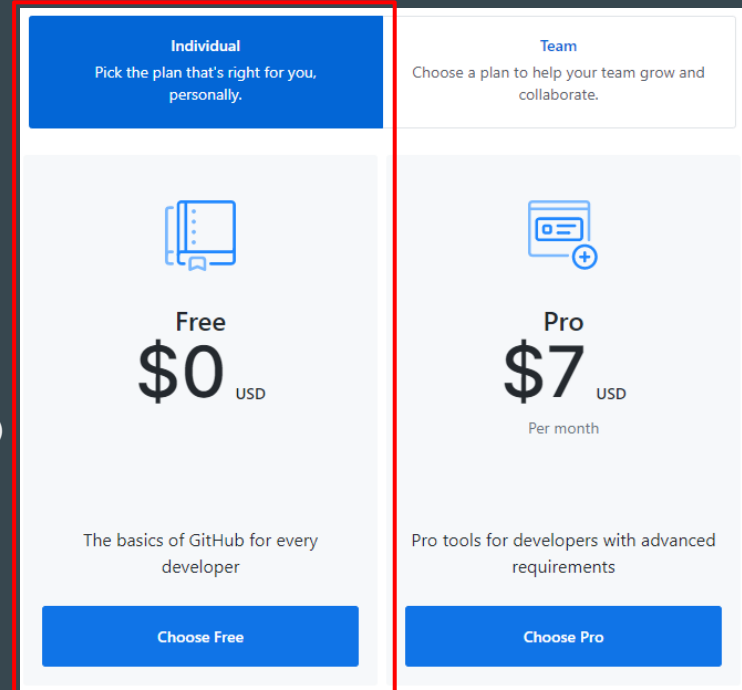
- Go to the Github homepage: [link](#)
- Fill in your account information (Fig 4.) and choose free version of account in the sign-up process (Fig 5.)



The screenshot shows the GitHub homepage with a dark background. On the left, the text "Built for developers" is prominent, followed by a description of GitHub as a development platform. On the right, a white sign-up form is highlighted with a red border. The form contains fields for Username (KaistZelatore), Email (zelatore85@kaist.ac.kr), and Password (masked with dots). A green "Sign up for GitHub" button is at the bottom of the form. Above the form, there is a "Sign up" button and a menu icon in the top right corner.

Fig 1. Creating your account

...
(Some steps)

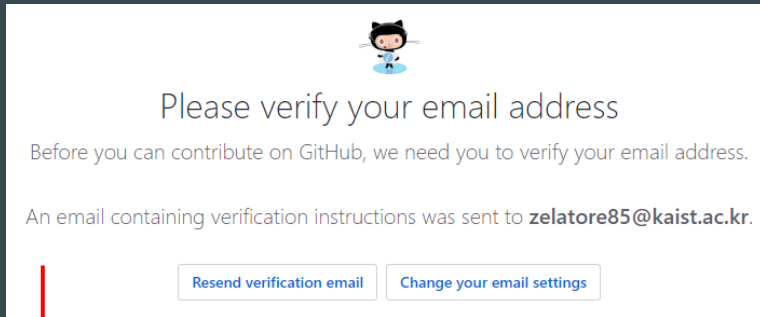


The screenshot shows the GitHub account selection page with two columns. The left column, highlighted with a red border, is for the "Individual" plan. It features a blue header with the text "Individual" and "Pick the plan that's right for you, personally." Below this is a light blue box with a laptop icon, the text "Free \$0 USD", and a description "The basics of GitHub for every developer". At the bottom is a blue button labeled "Choose Free". The right column is for the "Team" plan. It features a white header with the text "Team" and "Choose a plan to help your team grow and collaborate." Below this is a light blue box with a laptop icon and a plus sign, the text "Pro \$7 USD Per month", and a description "Pro tools for developers with advanced requirements". At the bottom is a blue button labeled "Choose Pro".

Fig2. Choosing free version of account

Create a Github account

- Verify your account through email (Fig 3.) and create the first new repository (Fig 4.) after verification
- In step Fig 4. , choose 'Public' option and check the box of README initialization



Open your email

Almost done, **@KaistZelatore!** To complete your GitHub sign up, we just need to verify your email address: **zelatore85@kaist.ac.kr**.

Verify email address

Fig 3. Verification process

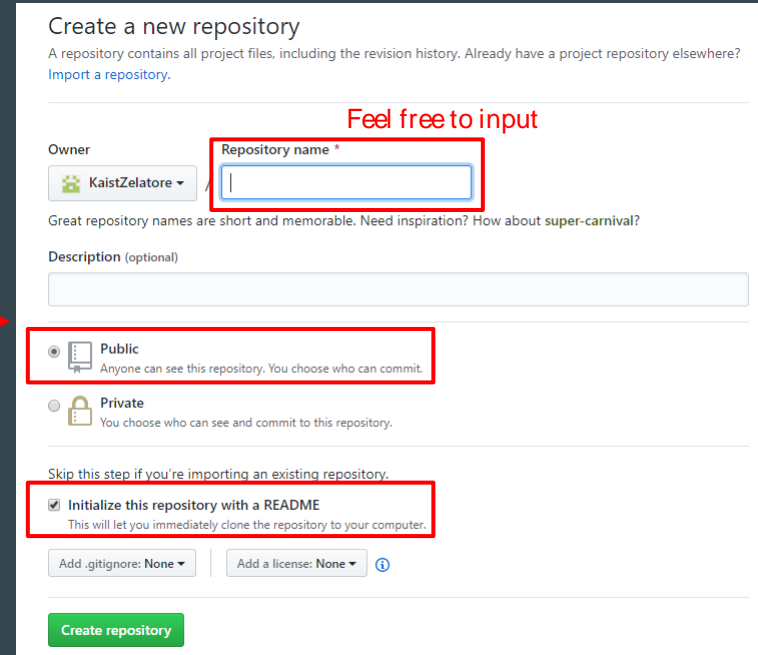


Fig 4. Create a new repository

Create a Github account

- Click the repository you made (Fig 5.) and see the repository to check if it completed successfully (Fig 6.)

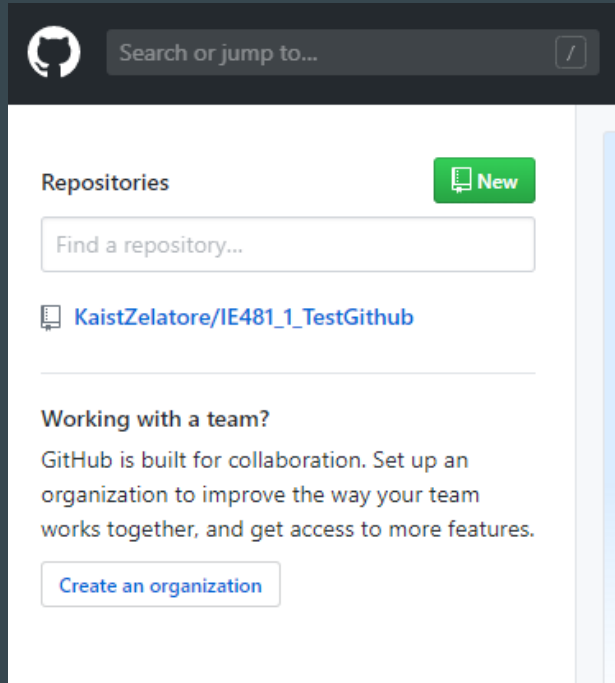


Fig 5. Your repository list

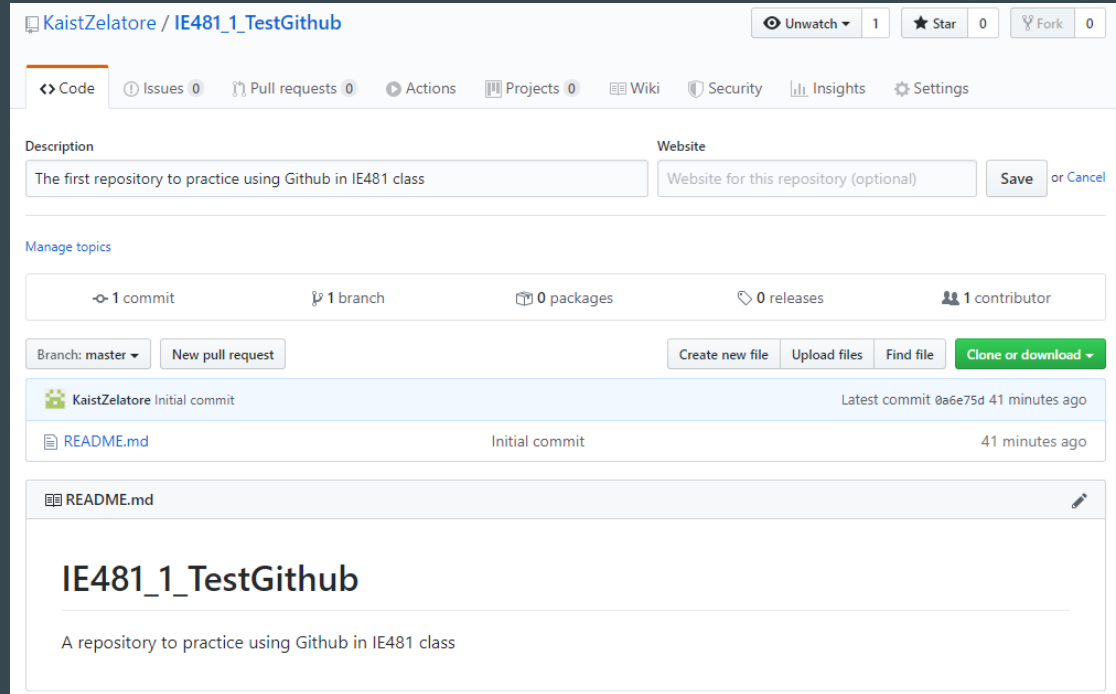


Fig 6. Selected repository view (No any files so far)

Now, Let's practice Git and Github!

Contents

- **Git/Github**

- Install and configure Git
- Create a Github account
- **Guideline for using Git and Github**

- **Dash**

- Install and configure Dash
- Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)

- **Heroku**

- Install Heroku and create an account
- A sample example of web hosting using Heroku
- Guideline for hosting web services using Python with Heroku

Save files to remote server (Local → Remote)

- Step 1: Go to the project directory path to upload
- Step 2: Initialize Git and commit all files to the local repository

```
C:\Users\zelatore\Desktop\flaskexample-class\ch01>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: CE12-FEF2

C:\Users\zelatore\Desktop\flaskexample-class\ch01 디렉터리

2020-03-25 오후 10:24 <DIR>          .
2020-03-25 오후 10:24 <DIR>          ..
2020-03-25 오후 10:24                865 hello.py
                1개 파일              865 바이트
                2개 디렉터리 26,226,577,408 바이트 남음

C:\Users\zelatore\Desktop\flaskexample-class\ch01>git remote
origin

C:\Users\zelatore\Desktop\flaskexample-class\ch01>git init .
Initialized empty Git repository in C:/Users/zelatore/Desktop/flaskexample-class/ch01/.git/

C:\Users\zelatore\Desktop\flaskexample-class\ch01>git add .
warning: LF will be replaced by CRLF in hello.py.
The file will have its original line endings in your working directory

C:\Users\zelatore\Desktop\flaskexample-class\ch01>git commit -m "This is the first commit"
[master (root-commit) dd3fc72] This is the first commit
1 file changed, 26 insertions(+)
create mode 100644 hello.py
```

Step 1

Step 2

Save files to remote server (Local → Remote)

- Step 3: Choose a remote repository to upload your files
- Step 4: (Login credentials) → push all files

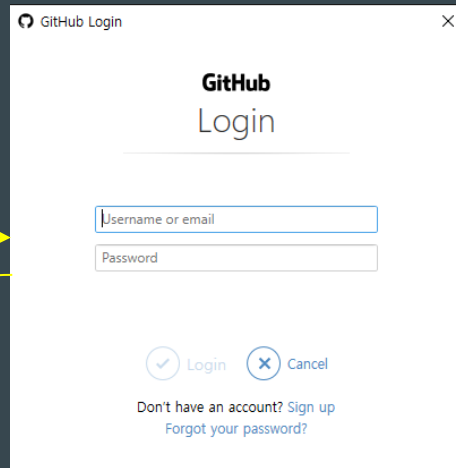
Choose a remote repository

```
C:\Users\zelatore\Desktop\flaskexample-class\ch01>git remote add origin https://github.com/KaistZelatore/TestGit
```

```
C:\Users\zelatore\Desktop\flaskexample-class\ch01>git remote -v
origin https://github.com/KaistZelatore/TestGit (fetch)
origin https://github.com/KaistZelatore/TestGit (push)
```

```
C:\Users\zelatore\Desktop\flaskexample-class\ch01>git push origin master
```

Login window
popped up

A screenshot of the GitHub Login window. It has a title bar with the GitHub logo and the text 'GitHub Login'. The main content area has the GitHub logo and the word 'Login' in a large font. Below this are two input fields: 'Username or email' and 'Password'. At the bottom, there are two buttons: 'Login' (with a checkmark icon) and 'Cancel' (with an 'X' icon). Below the buttons, there is a link that says 'Don't have an account? Sign up' and another link that says 'Forgot your password?'.

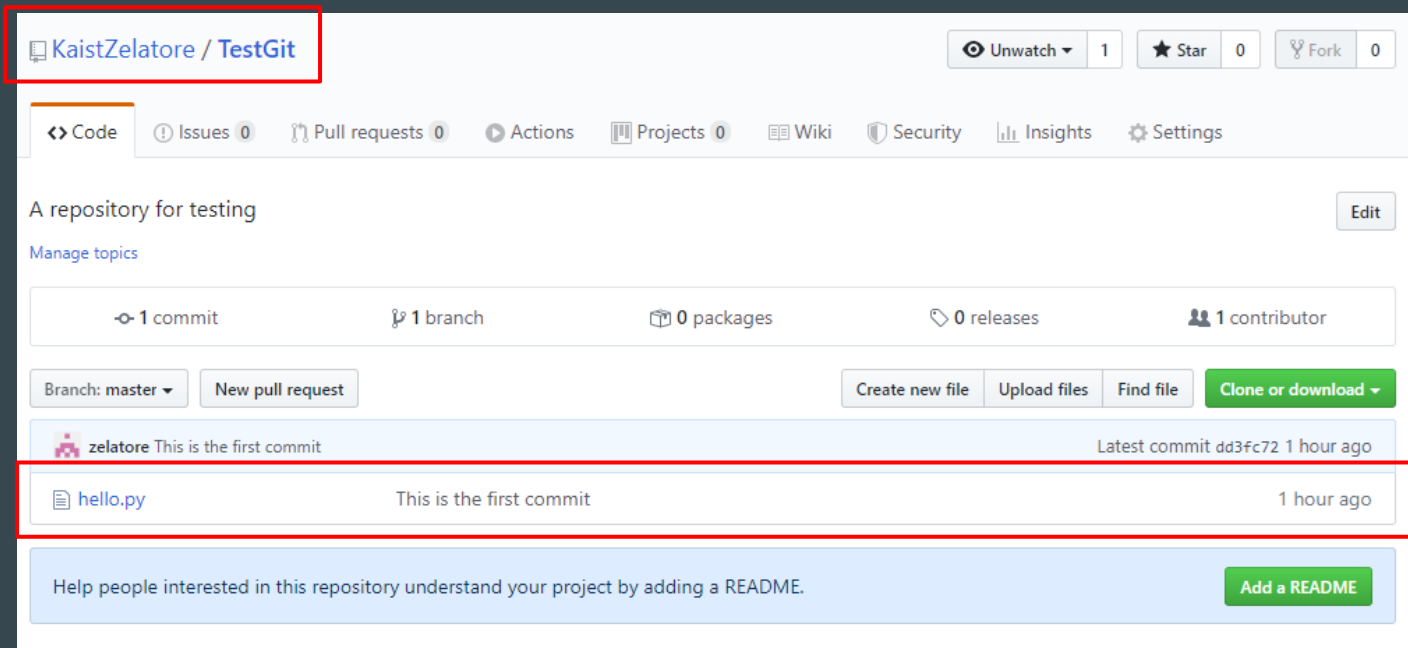
```
C:\Users\zelatore\Desktop\flaskexample-class\ch01>git push -f origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 635 bytes | 635.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KaistZelatore/TestGit
+ 8dBff7e...dd3fc72 master -> master (forced update)

C:\Users\zelatore\Desktop\flaskexample-class\ch01>
```

After login credentials,
Archiving process will be completed

Save files to remote server (Local → Remote)

- Step 5: Check and see the uploaded files



Get project from remote server (Remote → Local)

- Using 'git clone' instructions

```
C:\Users\zelatore\Desktop\cloneDir>git clone https://github.com/KaistZelatore/TestGit.git
```

git clone '<remote git path>'

```
Cloning into 'TestGit'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

```
C:\Users\zelatore\Desktop\cloneDir>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 0E12-FEF2
```

```
C:\Users\zelatore\Desktop\cloneDir 디렉터리
```

```
2020-04-01 오전 11:55 <DIR> .
```

```
2020-04-01 오전 11:55 <DIR> ..
```

```
2020-04-01 오전 11:55 <DIR> TestGit
```

```
0개 파일 0 바이트
3개 디렉터리 26,225,934,336 바이트 남음
```

Remote repository was cloned to the local directory

```
C:\Users\zelatore\Desktop\cloneDir>cd TestGit
```

```
C:\Users\zelatore\Desktop\cloneDir\TestGit>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 0E12-FEF2
```

```
C:\Users\zelatore\Desktop\cloneDir\TestGit 디렉터리
```

```
2020-04-01 오전 11:55 <DIR> .
```

```
2020-04-01 오전 11:55 <DIR> ..
```

```
2020-04-01 오전 11:55 891 hello.py
```

```
1개 파일 891 바이트
```

```
2개 디렉터리 26,225,930,240 바이트 남음
```

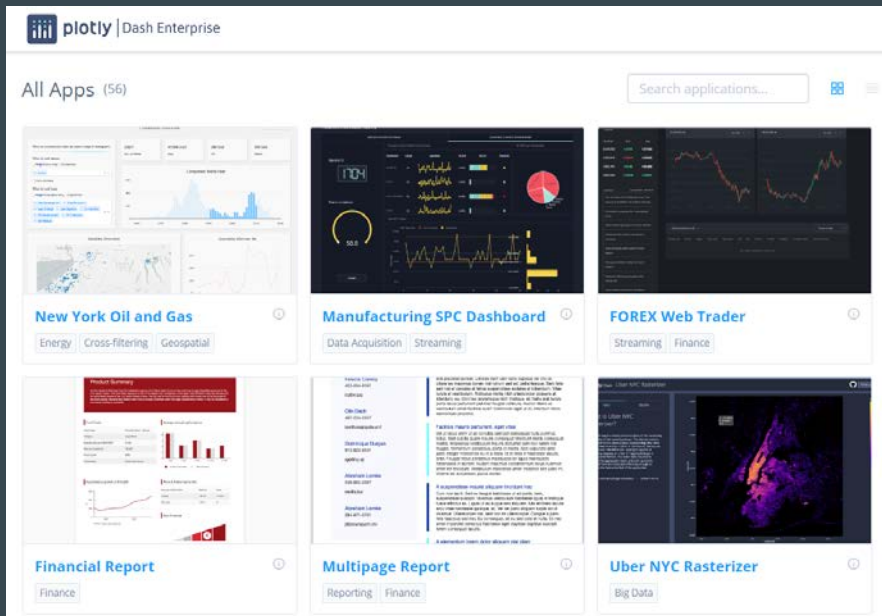
Go to the directory and see the file

```
C:\Users\zelatore\Desktop\cloneDir\TestGit>
```

Dash

What is Dash?

- Open source python library for creating 'reactive' web application
- Integrating with Flask for building a variety of data-driven web application
- Usually used with Plotly library for data analytics app (e.g., data visualization dashboard application)



A variety of examples of Dashboard application using Dash with Plotly

<https://dash-gallery.plotly.host/Portal/>

<https://medium.com/plotly/introducing-dash-5ecf7191b503>

Contents

- Git/Github

- Install and configure Git
- Create a Github account
- Guideline for using Git and Github

- Dash

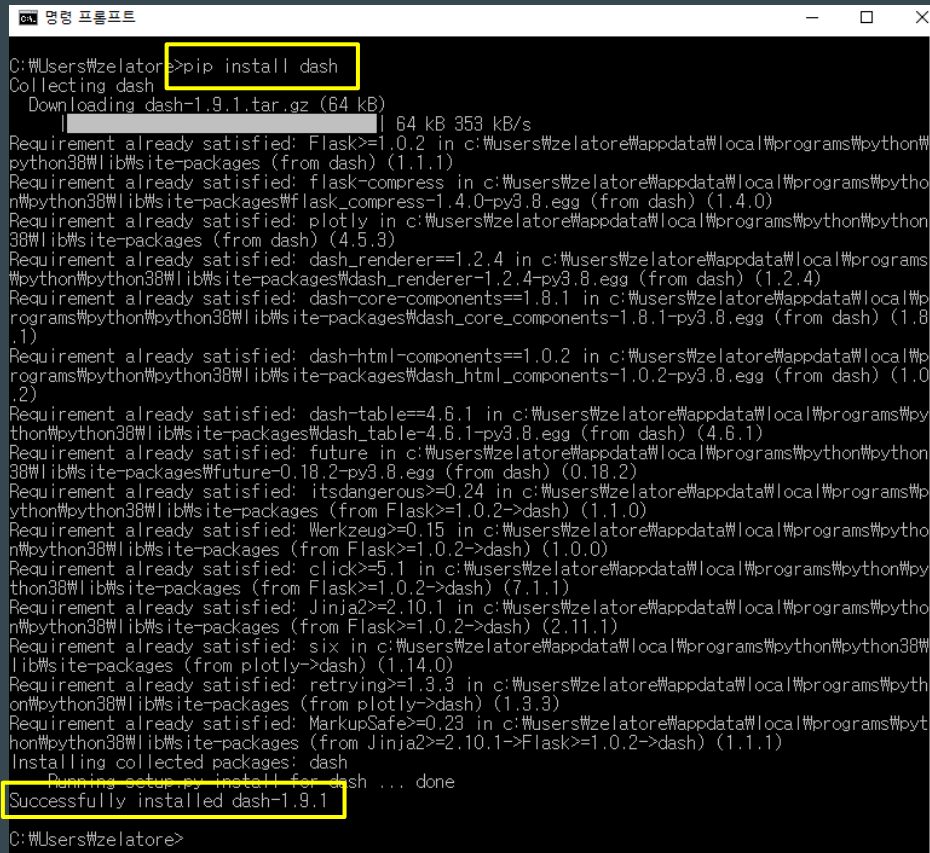
- Install and configure Dash
- Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)

- Heroku

- Install Heroku and create an account
- A sample example of web hosting using Heroku
- Guideline for hosting web services using Python with Heroku

Install Dash

- Just typing the text below on the command line
- 'pip install dash'



```
C:\Users\zelatore>pip install dash
Collecting dash
  Downloading dash-1.9.1.tar.gz (64 kB)
    64 kB 353 kB/s
Requirement already satisfied: Flask>=1.0.2 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from dash) (1.1.1)
Requirement already satisfied: flask-compress in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages\flask_compress-1.4.0-py3.8.egg (from dash) (1.4.0)
Requirement already satisfied: plotly in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from dash) (4.5.3)
Requirement already satisfied: dash-renderer==1.2.4 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages\dash_renderer-1.2.4-py3.8.egg (from dash) (1.2.4)
Requirement already satisfied: dash-core-components==1.8.1 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages\dash_core_components-1.8.1-py3.8.egg (from dash) (1.8.1)
Requirement already satisfied: dash-html-components==1.0.2 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages\dash_html_components-1.0.2-py3.8.egg (from dash) (1.0.2)
Requirement already satisfied: dash-table==4.6.1 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages\dash_table-4.6.1-py3.8.egg (from dash) (4.6.1)
Requirement already satisfied: future in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages\future-0.18.2-py3.8.egg (from dash) (0.18.2)
Requirement already satisfied: itsdangerous>=0.24 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from Flask>=1.0.2->dash) (1.1.0)
Requirement already satisfied: Werkzeug>=0.15 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from Flask>=1.0.2->dash) (1.0.0)
Requirement already satisfied: click>=5.1 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from Flask>=1.0.2->dash) (7.1.1)
Requirement already satisfied: Jinja2>=2.10.1 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from Flask>=1.0.2->dash) (2.11.1)
Requirement already satisfied: six in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from plotly->dash) (1.14.0)
Requirement already satisfied: retrying>=1.3.3 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from plotly->dash) (1.3.3)
Requirement already satisfied: MarkupSafe>=0.23 in c:\Users\zelatore\AppData\Local\Programs\Python\Python38\lib\site-packages (from Jinja2>=2.10.1->Flask>=1.0.2->dash) (1.1.1)
Installing collected packages: dash
  Running setup.py install for dash ... done
Successfully installed dash-1.9.1
C:\Users\zelatore>
```

Contents

- Git/Github

- Install and configure Git
- Create a Github account
- Guideline for using Git and Github

- Dash

- Install and configure Dash
- Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)

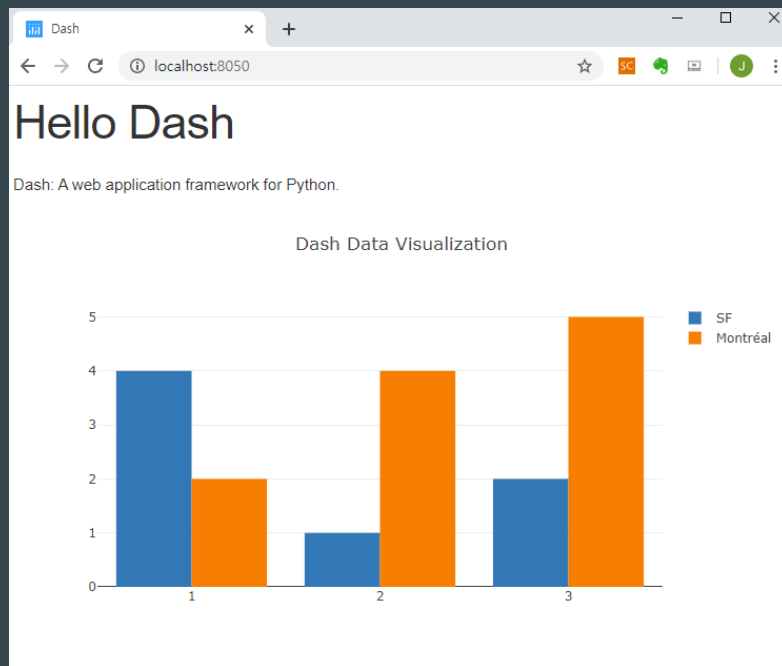
- Heroku

- Install Heroku and create an account
- A sample example of web hosting using Heroku
- Guideline for hosting web services using Python with Heroku

An example of Dash application

- Using 'dash_core_components' to include and use Dash components (e.g., Graph) in a web page
- Using 'dash_html_components' to provide page layout information
- You may use the source code [here](https://dash.plotly.com/layout)

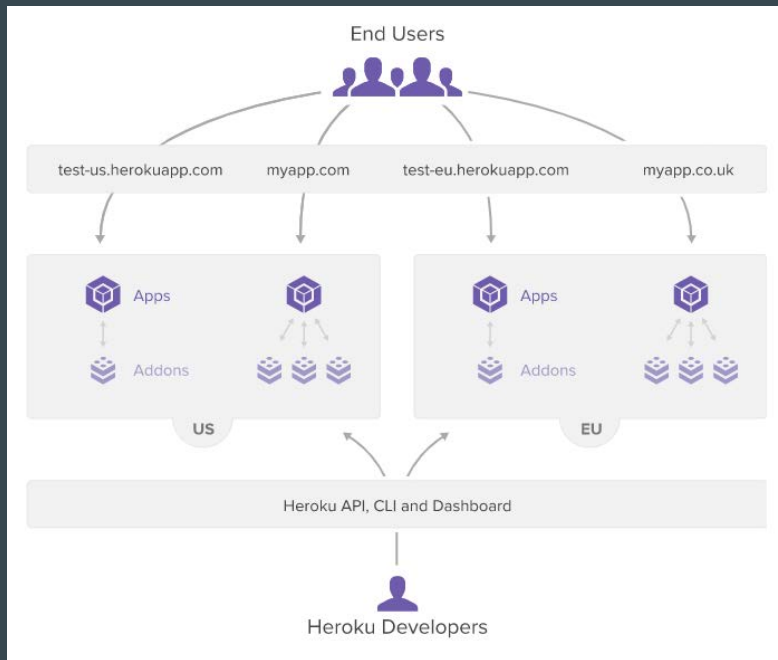
```
1 # -*- coding: utf-8 -*-
2 import dash
3 import dash_core_components as dcc
4 import dash_html_components as html
5
6 external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
7
8 app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
9
10 app.layout = html.Div(children=[
11     html.H1(children='Hello Dash'),
12
13     html.Div(children='''
14         Dash: A web application framework for Python.
15     '''),
16
17     dcc.Graph(
18         id='example-graph',
19         figure={
20             'data': [
21                 {'x': [1, 2, 3], 'y': [4, 1, 2], 'type': 'bar', 'name': 'SF'},
22                 {'x': [1, 2, 3], 'y': [2, 4, 5], 'type': 'bar', 'name': 'Montréal'},
23             ],
24             'layout': {
25                 'title': 'Dash Data Visualization'
26             }
27         }
28     )
29 ])
30
31 if __name__ == '__main__':
32     app.run_server(debug=True)
```



Heroku

What is Heroku?

- A cloud platform that helps users to deploy your app
- Support a free cloud web hosting services to run your web application (e.g., Flask app)



<https://devcenter.heroku.com/categories/reference>

<https://blog.heroku.com/europe-region>

Contents

- Git/Github

- Install and configure Git
- Create a Github account
- Guideline for using Git and Github

- Dash

- Install and configure Dash
- Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)

- Heroku

- Install Heroku and create an account
- A sample example of web hosting using Heroku
- Guideline for hosting web services using Python with Heroku

Install Heroku and create an account

- Go to the Heroku homepage: [link](#)
- Click the sign up button (Fig 1.)
- Fill in your information (Fig 2.)

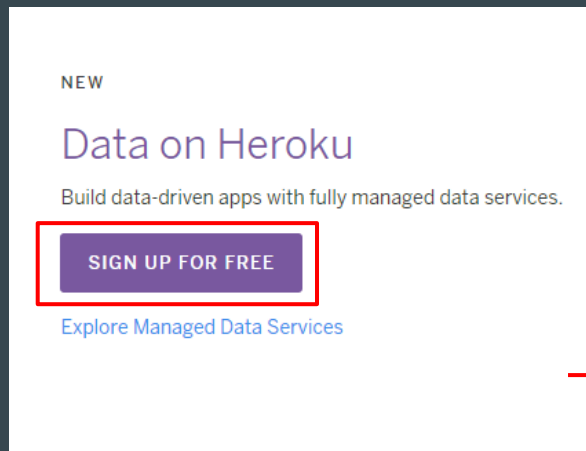


Fig 1. Click sign up button

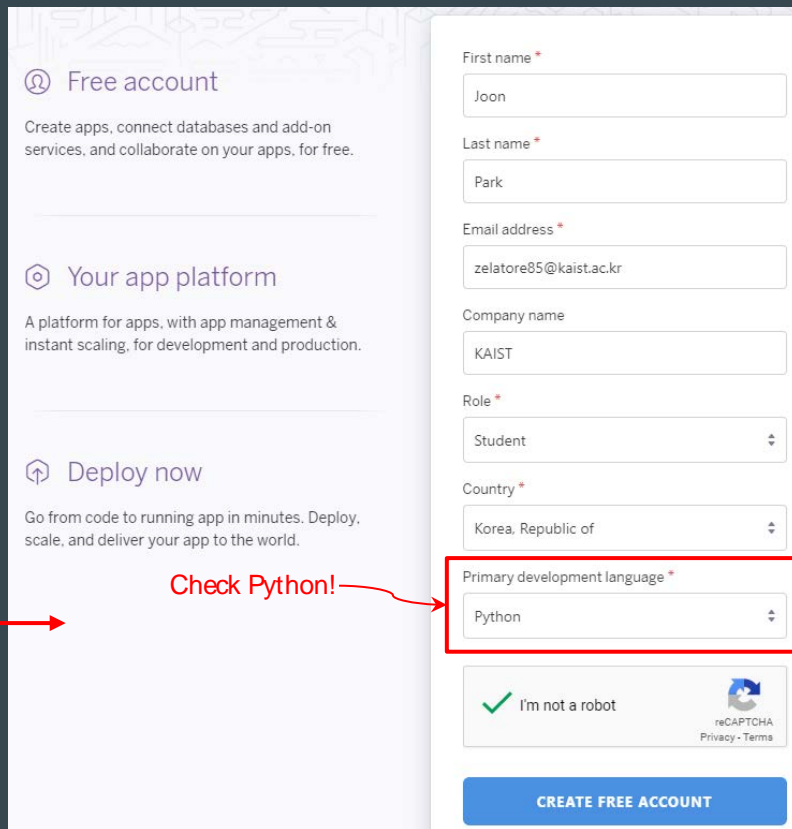
A screenshot of the Heroku account creation form. The form is titled 'Free account' and 'Your app platform'. It contains several input fields: 'First name' (Joon), 'Last name' (Park), 'Email address' (zelatore85@kaist.ac.kr), 'Company name' (KAIST), 'Role' (Student), and 'Country' (Korea, Republic of). The 'Primary development language' dropdown menu is highlighted with a red rectangular box, and a red arrow points to it with the text 'Check Python!'. Below the dropdown is a reCAPTCHA checkbox labeled 'I'm not a robot'. At the bottom of the form is a blue button labeled 'CREATE FREE ACCOUNT'.

Fig 2. Create your account

Install Heroku and create an account

- Verify your account by email (Fig 3.) and finalize setting up your account (Fig 4.)

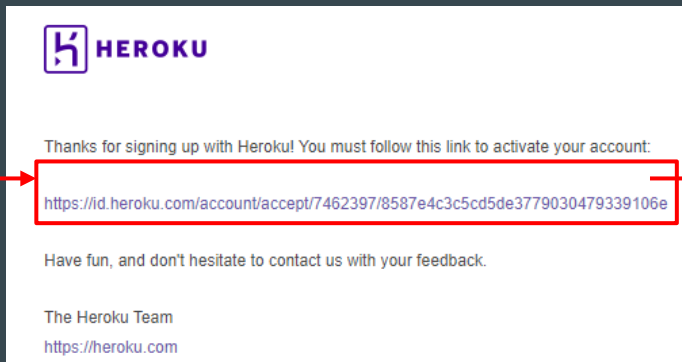
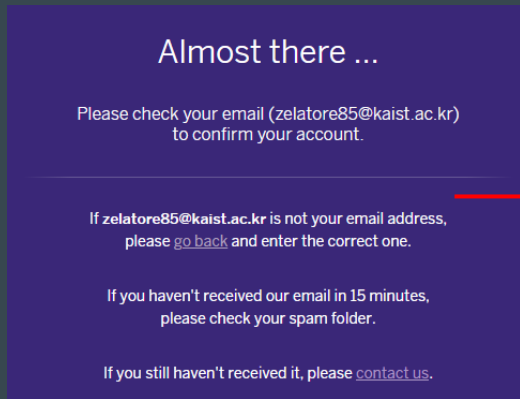


Fig 3. Confirm your account in your email

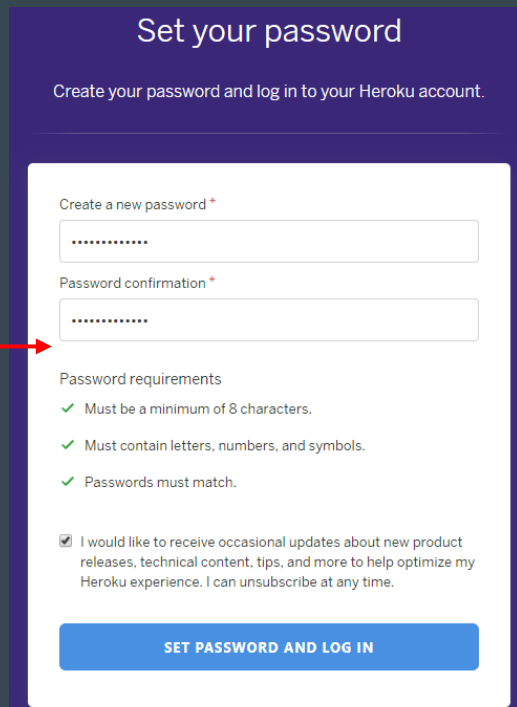



Fig 4. Finalize your account by creating your password

Install Heroku and create an account

- After logging in, install the Heroku (Go to the [download page](#))
- In the download page, choose your OS version and download the executable file (Fig 5.)


Set up

 The Heroku CLI requires Git, the popular version control system. If you don't already have Git installed, complete the following before proceeding:

- [Git installation](#)
- [First-time Git setup](#)

In this step you'll install the Heroku Command Line Interface (CLI). You use the CLI to manage and scale your applications, provision add-ons, view your application logs, and run your application locally.

Download and run the installer for your platform:

 **macOS**
[Download the installer](#)
Also available via Homebrew:

```
$ brew install heroku/brew/heroku
```


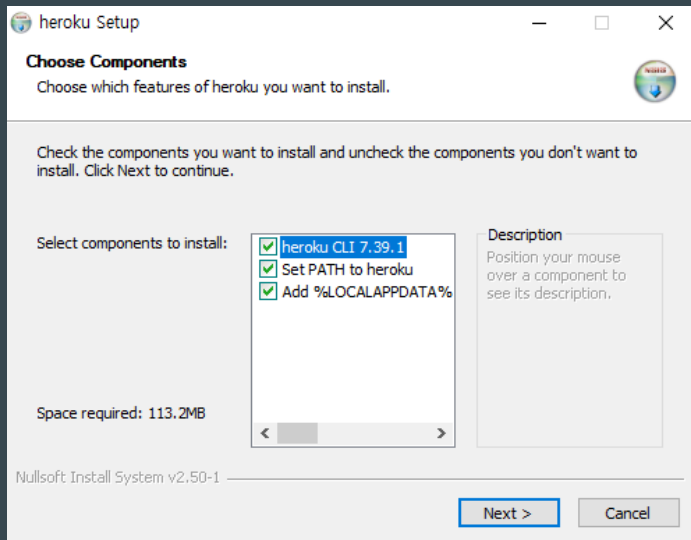
 **Windows**
Download the appropriate installer for your Windows installation:
[64-bit installer](#)
[32-bit installer](#)

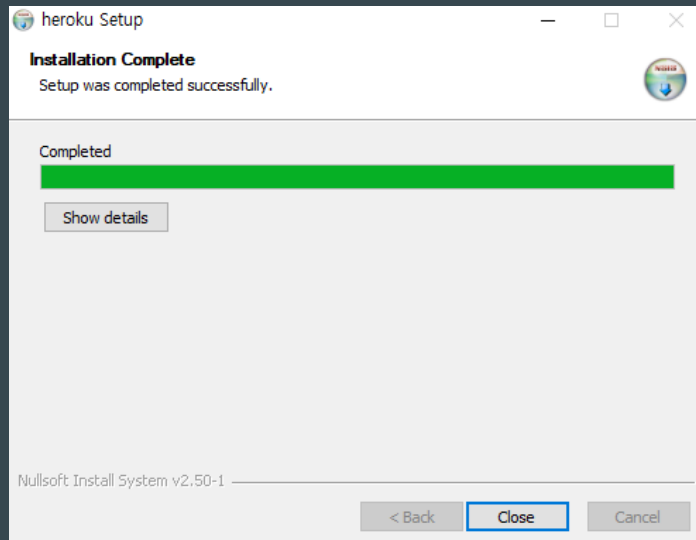
Fig 5. Choose the executable file based on your OS version

Install Heroku and create an account

- Iteratively click the 'Next' button, and complete your installation



...
(Some steps)



Install Heroku and create an account

- Check your installation result
- Type 'heroku' in a command line (Fig 6.)

```
C:\Users\#zelatore>heroku
```

```
CLI to interact with Heroku
```

```
VERSION
```

```
heroku/7.39.1 win32-x64 node-v12.13.0
```

```
USAGE
```

```
$ heroku [COMMAND]
```

```
COMMANDS
```

access	manage user access to apps
addons	tools and services for developing, extending, and operating your app
apps	manage apps on Heroku
auth	check 2fa status
authorizations	OAuth authorizations
autocomplete	display autocomplete installation instructions
base	
buildpacks	scripts used to compile apps
certs	a topic for the ssl plugin
ci	run an application test suite on Heroku
clients	OAuth clients on the platform
config	environment variables of apps
container	Use containers to build and deploy Heroku apps
domains	custom domains for apps
drains	forward logs to syslog or HTTPS
features	add/remove app features
git	manage local git repository for app
help	display help for heroku
keys	add/remove account ssh keys
labs	add/remove experimental features
local	run Heroku app locally
logs	display recent log output
maintenance	enable/disable access to app
members	manage organization members
notifications	display notifications
orgs	manage organizations
pg	manage postgresql databases

Contents

- Git/Github

- Install and configure Git
- Create a Github account
- Guideline for using Git and Github

- Dash

- Install and configure Dash
- Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)

- Heroku

- Install Heroku and create an account
- A sample example of web hosting using Heroku
- Guideline for hosting web services using Python with Heroku

Configure web hosting for your app using Heroku

- If you complete creating account, you can see the Fig 1.
- Otherwise, you can see the same page (Fig 2.) by logging in at the [login page](#)

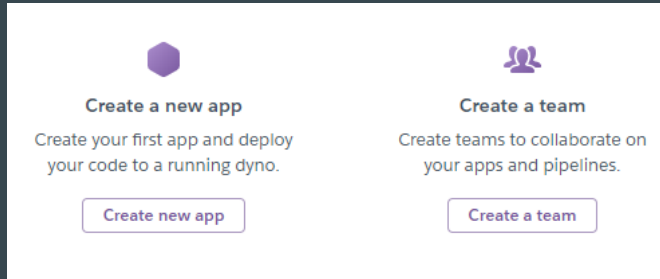


Fig 1. A web page for creating a new app

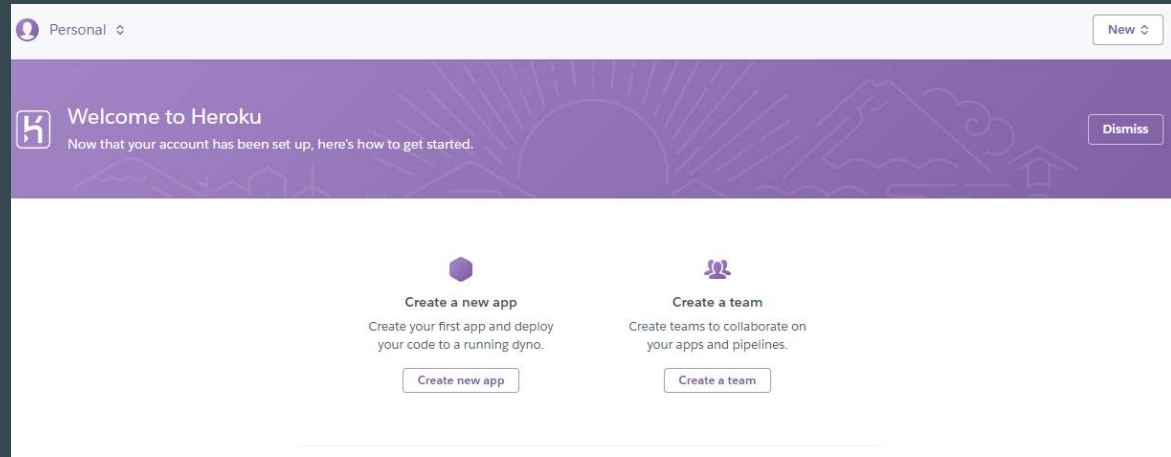


Fig 2. The same as the Fig 1. (Accessed by log in)

Configure web hosting for your app using Heroku

- Click the 'create new app' button (Fig 1), then fill in your app for your web hosting
- You are not able to change your region, just keep choosing the default option and click 'Create App' button

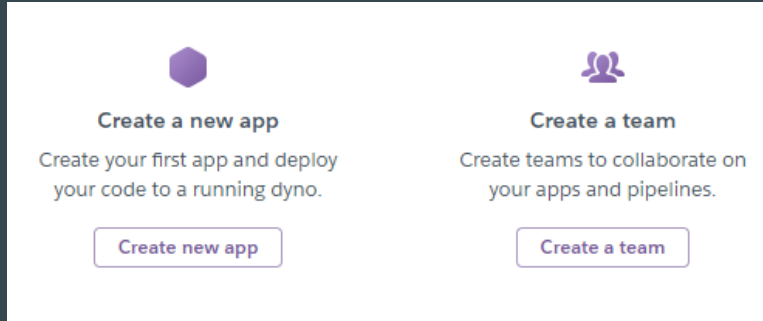


Fig 1. A web page for creating a new app

This image shows the 'Create app' form on Heroku. It includes an 'App name' field with the value 'ie481-testtheroku' and a green checkmark icon. Below the field, it says 'ie481-testtheroku is available'. There is a 'Choose a region' dropdown menu showing 'United States' with a US flag icon. At the bottom, there is a purple button labeled 'Create app' and a link 'Add to pipeline...' with a hexagon icon.

Fig 3. Finalize your web hosting by clicking the button above

Configure web hosting for your app using Heroku

- Now, you can see a dashboard web page configuring your hosting app (Fig 4.)
- By clicking 'Open app' button in Fig 4., you can actually see and confirm your hosted app
 - In the Heroku, all of the hosted app can be accessed by url like '[your app name].herokuapp.com'
- In heroku, you may create your own app up to five
 - For creating the 6th app, you must remove an existing app)

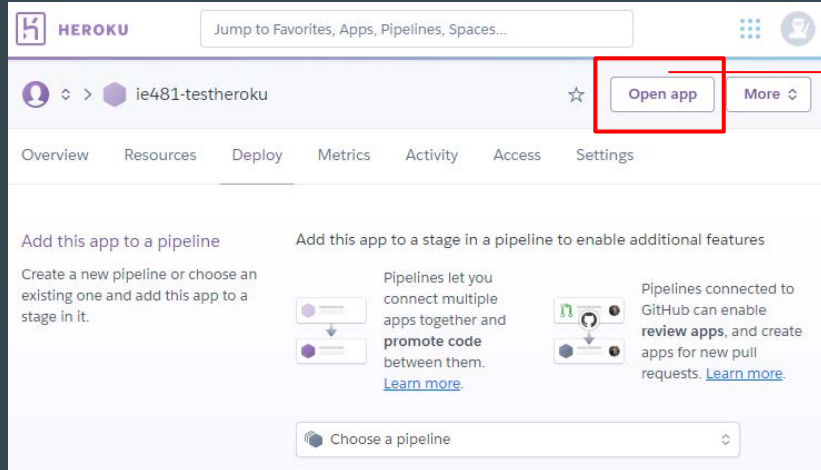


Fig 4. Dashboard page for configuring hosting app

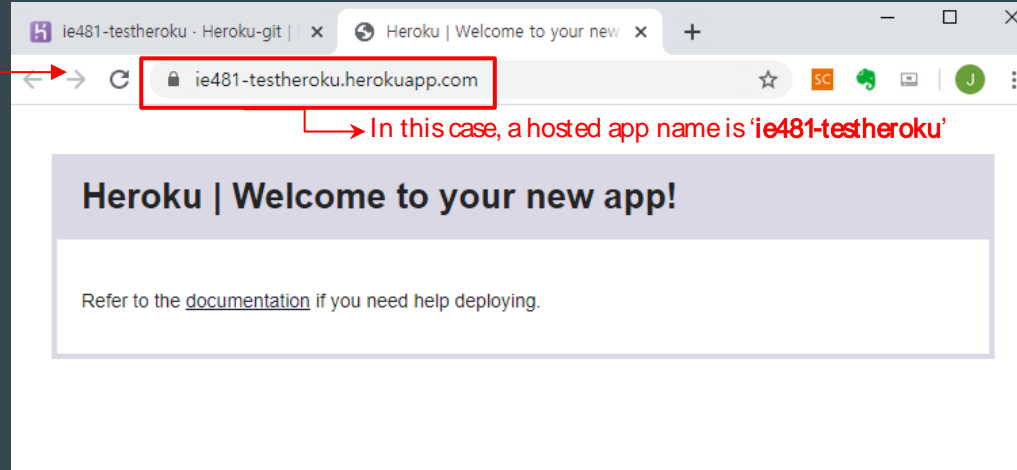


Fig 5. Hosted app service

Contents

- Git/Github

- Install and configure Git
- Create a Github account
- Guideline for using Git and Github

- Dash

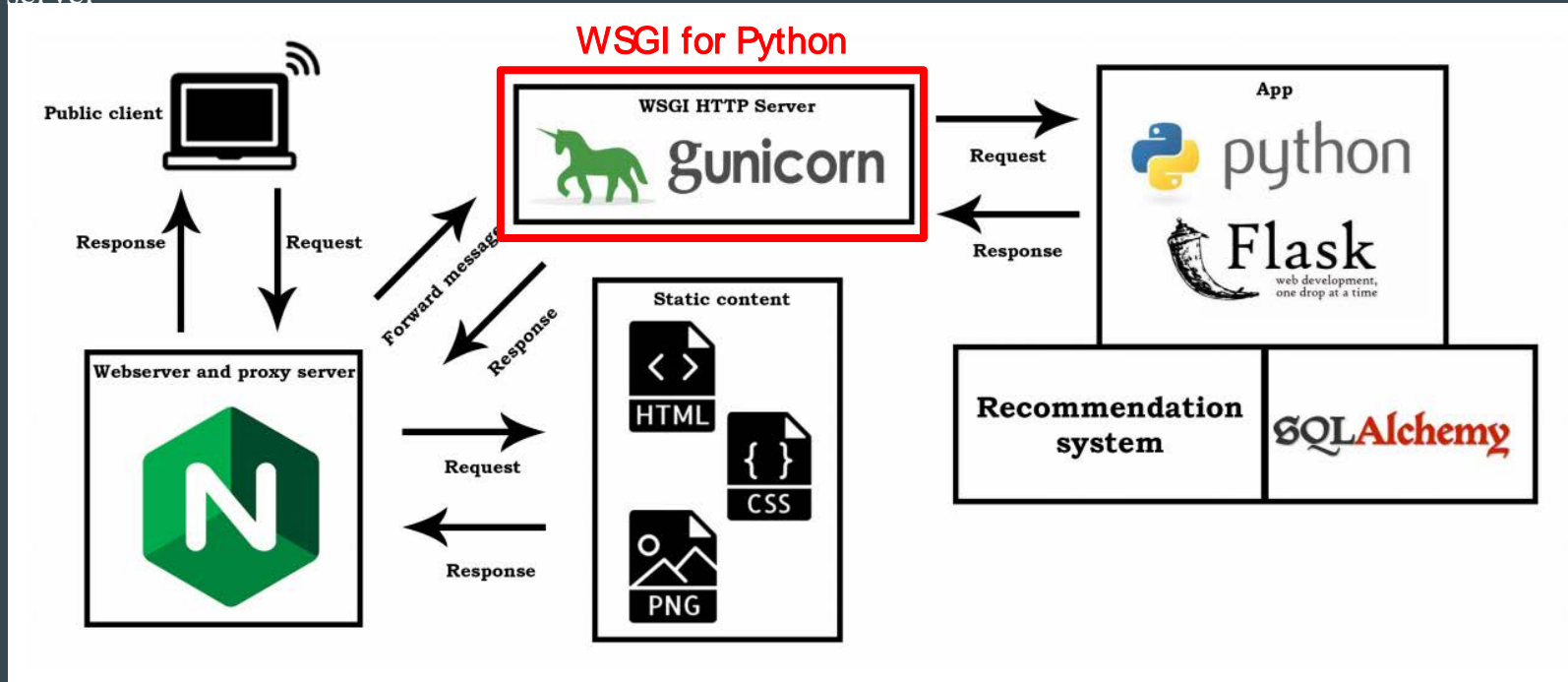
- Install and configure Dash
- Test a basic example of Dash to understand how it works (prerequisite learning for Lab #3)

- Heroku

- Install Heroku and create an account
- A sample example of web hosting using Heroku
- Guideline for hosting web services using Python with Heroku

Process of python application web hosting

- Needs to have Web Server Gateway Interface(WSGI) to communicate between web server and python app
- One kind of WSGI for python is 'unicorn' → Interpret the python script to communicate with web server



For python web application hosting, We need to have 3 modules

Web App

(Python code)

WSGI

(e.g., gunicorn)

Hosting Server

(e.g., Heroku)

Guideline for hosting web services using Python with Heroku

- Step 1: Install gunicorn modules → 'pip install gunicorn' in command line
- Step 2: Create a python code executed on the server (Here, we use a Dash example [code](#))

```
C:\Users\zelatore>pip install gunicorn
Collecting gunicorn
  Using cached gunicorn-20.0.4-py2.py3-none-any.whl (77 kB)
Requirement already satisfied: setuptools>=3.0 in c:\users\zelatore\appdata\local\programs\python\python38\lib\site-packages (from gunicorn) (41.2.0)
Installing collected packages: gunicorn
Successfully installed gunicorn-20.0.4

C:\Users\zelatore>
```

```
1 # -*- coding: utf-8 -*-
2
3 import os
4 import flask
5 import dash
6 import dash_core_components as dcc
7 import dash_html_components as html
8
9 external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
10
11 server = flask.Flask(__name__)
12 server.secret_key = os.environ.get('secret_key', str(randint(0, 1000000)))
13 app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
14
15 app.layout = html.Div(children=[
16     html.H1(children='Hello Dash'),
17
18     html.Div(children='''
19         Dash: A web application framework for Python.
20     '''),
21
22     dcc.Graph(
23         id='example-graph',
24         figure={
25             'data': [
26                 {'x': [1, 2, 3], 'y': [4, 1, 2], 'type': 'bar', 'name': 'SF'},
27                 {'x': [1, 2, 3], 'y': [2, 4, 5], 'type': 'bar', 'name': 'u'Montréal'},
28             ],
29             'layout': {
30                 'title': 'Dash Data Visualization'
31             }
32         }
33     )
34 ])
35
36 if __name__ == '__main__':
37     app.run_server(debug=True)
```

Guideline for hosting web services using Python with Heroku

- Step 3: In command line, go to the directory including python source code that you created (Fig 1.)
- Step 4: To host python application using Heroku, we need to have 2 files
 - **'Procfile'**: To let Heroku know the sequence of instructions for executing python code
 - **'requirements.txt'**: To let Heroku know which python packages are needed to run the application

The screenshot shows a Windows command prompt with the following commands and outputs:

```
C:\Users\zelatore\Desktop\HerokuApp1>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: CE12-FEF2

C:\Users\zelatore\Desktop\HerokuApp1 디렉터리

2020-03-29 오전 03:40 <DIR>          .
2020-03-29 오전 03:40 <DIR>          ..
2020-03-29 오전 01:45          1,008 app.py
                       1개 파일      1,008 바이트
                       2개 디렉터리 26,113,499,136 바이트 남음

C:\Users\zelatore\Desktop\HerokuApp1>echo web:unicorn app:server > Procfile
C:\Users\zelatore\Desktop\HerokuApp1>pip freeze > requirements.txt
C:\Users\zelatore\Desktop\HerokuApp1>
```

Annotations on the screenshot:

- A yellow bracket on the right side of the directory listing is labeled "Directory path Including Python code".
- A yellow box highlights the command `echo web:unicorn app:server > Procfile`, with an arrow pointing to the label "Creating Procfile".
- A yellow box highlights the command `pip freeze > requirements.txt`, with an arrow pointing to the label "Creating requirements.txt (automatically)".

On the right side of the image, there is a list of Python dependencies from a `requirements.txt` file:

```
Flask==1.1.1
flask-assets==2.0
flask-compress==1.4.0
future==0.18.2
gunicorn==20.0.4
itsdangerous==1.1.0
Jinja2==2.11.1
MarkupSafe==1.1.1
numpy==1.18.1
pandas==1.0.1
pathlib==1.0.1
plotly==4.5.4
plotly-flask-tutorial==0.0.1
python-dateutil==2.8.1
pytz==2019.3
retrying==1.3.3
six==1.14.0
```

Fig 1. Guideline for creating 2 dependency files

An example of requirements.txt

Guideline for hosting web services using Python with Heroku

- Step 5: Initialize, add, and commit the python source code using Git
- Step 6: Create your hosting app
- Step 6: Push all resource files (Procfile, requirements.txt, and python code) to a Heroku remote github

```
C:\Users\zelatore\Desktop\HerokuApp1>git init
Initialized empty Git repository in C:/Users/zelatore/Desktop/HerokuApp1/.git/

C:\Users\zelatore\Desktop\HerokuApp1>git add .

C:\Users\zelatore\Desktop\HerokuApp1>git commit -m "The first dash app commit"
[master (root-commit) e9c46c4] The first dash app commit
3 files changed, 63 insertions(+)
create mode 100644 Procfile
create mode 100644 app.py
create mode 100644 requirements.txt
```

Archiving all files to repository
using Git

```
C:\Users\zelatore\Desktop\HerokuApp1>heroku create ie481-dashapptest
Creating ● ie481-dashapptest... done
https://ie481-dashapptest.herokuapp.com/ | https://git.heroku.com/ie481-dashapptest.git
```

Creating an Heroku app (hosting name)

```
C:\Users\zelatore\Desktop\HerokuApp1>heroku git:remote -a ie481-dashapptest
set git remote heroku to https://git.heroku.com/ie481-dashapptest.git
```

Creating remote repository for Heroku app

```
C:\Users\zelatore\Desktop\HerokuApp1>git push heroku master
```

Pushing all files to remote repository

Guideline for hosting web services using Python with Heroku

- Step 6: Push all resource files (Procfile, requirements.txt, and python code) to a Heroku remote github
- Step 7: Test your web hosting result through web browser (E.g., '[your app name].herokuapp.com')

Deploying process to the Heroku

```
remote:      Stored in directory: /tmp/pip-ephem-wheel-cache-bf1tsc1k/wh
eels/ac/cb/8a/b27bf6323e2f4c462dcbf77d70b7c5e7868a7f6e12871770cf
remote:      Successfully built dash dash-core-components dash-html-compon
ents dash-renderer dash-table flask-compress future pathlib retrying
remote:      Installing collected packages: click, Werkzeug, MarkupSafe, J
inja2, itsdangerous, Flask, flask-compress, six, retrying, plotly, dash-rend
erer, dash-core-components, dash-html-components, dash-table, future, dash,
webassets, flask-assets, gunicorn, numpy, python-dateutil, pytz, pandas, pat
hlib
remote:      Successfully installed Flask-1.1.1 Jinja2-2.11.1 MarkupSafe-1
.1.1 Werkzeug-1.0.0 click-7.1.1 dash-1.9.1 dash-core-components-1.8.1 dash-h
tml-components-1.0.2 dash-renderer-1.2.4 dash-table-4.6.1 flask-assets-2.0 f
lask-compress-1.4.0 future-0.18.2 gunicorn-20.0.4 itsdangerous-1.1.0 numpy-1
.18.1 pandas-1.0.1 pathlib-1.0.1 plotly-4.5.4 python-dateutil-2.8.1 pytz-201
9.3 retrying-1.3.3 six-1.14.0 webassets-2.0
remote: -----> Discovering process types
remote:
remote: -----> Compressing...
remote:      Done: 97M
remote: -----> Launching...
remote:      Released v3
remote:      https://ie481-dashapptest.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/ie481-dashapptest.git
* [new branch]      master -> master
```

C:\Users\zelatore\Desktop\HerokuApp1>

