GIT and GitHub

Agenda

- What is Git
- How to install Git
- Basic Git usage
- Intro to GitHub

Sometimes when you work on some coding project, you program some new improvements. But later you discover these improvements were not that good so you'd like to go back to the old code. Unfortunately, the "undo" button doesn't work anymore and your good old code is lost forever...

Wouldn't it be great if you could have a historic archive of your code?

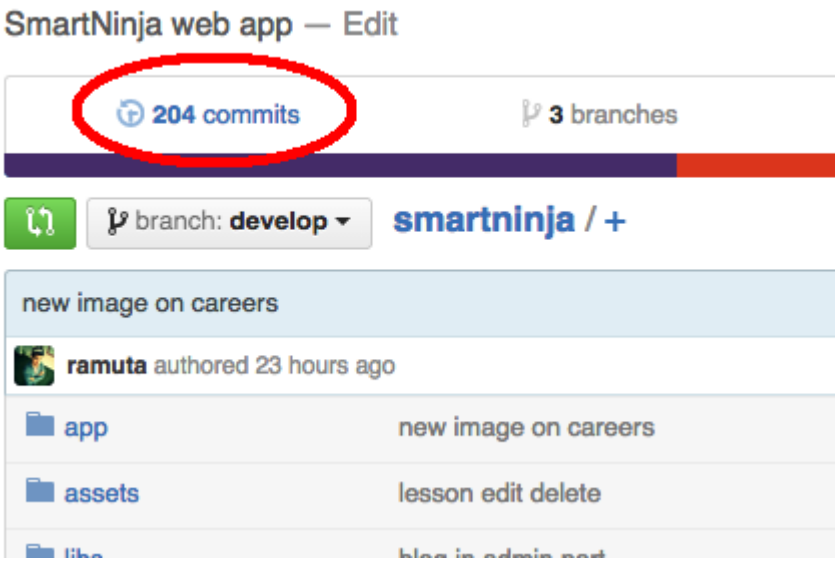
Well... you can! Not just *can*, every seroius programmer **should** have it. And it's called Git.

How does Git look like?

There's a website called GitHub where you can upload and save your code. Let's take a look for example a code for an ex SmartNinja forum: https://github.com/smartninja/forum. If you click for example on `main.py` you're able to see some of the forum code.

This is the latest version of the code. So where can we find the previous versions? After all, we said that Git provides you with a historic code archive (in Git parlance it's called a *repository*).

Go back to https://github.com/smartninja/forum and click on "commits":



Now you can see a list of all the changes ever made in the project. If you click on one, you can see what exactly was changed - what was added (marked green) and what was deleted (marked red).

Besides being a historic code archive, there are also other benefits with using Git:

- multiple programmers at once can work on a single project,
- GitHub serves as a code storage for you - in case your computer disk breaks and you loose all the data on it,
- the projects you have on GitHub serve as sort of a programming CV for you. Having a GitHub account with many projects on it can increase your chances of getting a coding job :)

Exercise 4.1: create your own Git repository

1. Create a user account on GitHub.
2. In the right side of the navigation bar (near your username), click on + (plus) and select "New repository". This is basically a code repository for some coding project of yours.
3. As a project name put `testproject` (don't use any space in the name).
4. You can add some description if you want, but leave the rest as it is.
5. Click on "Create repository".

Voila! Your first GitHub repository is created :) (Btw, leave that web page still open, because you'll need it later).

But the repository is empty... How can I upload some code to it?

For that, you'll need to first install Git on your computer.

How to install Git on your computer

1. Go to http://git-scm.com/downloads and download & install Git (just click next when installing - use the default options).
2. When Git is installed, open Git Bash - you can find it in the Start menu (Windows users only). If you use Mac or Linux, open the Terminal.
3. In Git Bash or Terminal you'll enter your name and email (so that Git can know who was the person that was changing the code). Enter these two commands (one after another) into Bash/Terminal (use your real name and email):

```
git config --global user.name "John Doe"

git config --global user.email johndoe@example.com
```

7. Create a new project folder on your computer.
8. Open the folder and right-click in it and select "Git Bash here" (if you use Windows). If you use Terminal you'll have to navigate to the folder (instructor will help you or ask on SmartNinja forum).
9. In Bash/Terminal execute the following commands (you have these commands on the web page when you created a new repository). IMPORTANT: The URL next to `git remote add origin` will be **different** in your case than in the example below. Use your own repository URL, not the one in the example!

```
echo "# test" >> README.md

git init

git add README.md

git commit -m "first commit"
```

17.
18. `git remote add origin https://github.com/ramuta/test.git`
19.
20. `git push -u origin master`
21. Go to GitHub and open your testproject repository. If everything went well, you should see a README.md file over there.

Let's repeat

These Git commands are the most important:

- `git init`: creates or initialises a Git repository (you execute this command only ONCE for the project. Later you don't execute it anymore).
- `git remote add origin https://github.com/ramuta/test.git`: connects a Git repository on your computer with the one on GitHub (you also execute this only ONCE, after you do git init command. And use a different URL of course).
- `git add .`: adds all new changes. Dot at the end is important - it means ALL changed files. You do this for every task that you finish.
- `git commit -m "some text message"`: commits the changes and adds a short message describing the changes that you made. You do this for every task that you finish and you can do it only if you do the "git add" command before.
- `git push origin master`: Push the changes you've made to GitHub.

Exercise 4.2: Upload Boogle to GitHub

- Go to GitHub and create a new repository
- Name it Boogle
- Upload Boogle code (from the previous lesson) in this repository. Use the knowledge that you've learned during this lesson.

Paste a link to the Boogle repository on the Forum.

If the exercise is to difficult, do the GitHub tutorial (4.3) first.

Home exercise 4.3: GitHub tutorial

Go through try.github.io tutorial: https://try.github.io/

Home exercise 4.4: preparation for the next lesson

In the next lesson you'll learn about Bootstrap. To prepare for the lesson, go to getbootstrap.com and download it.

Additional exercises (if you have time)

- Atlassian tutorial: https://www.atlassian.com/git/tutorials/
- Learn how to navigate in the Terminal: https://www.codecademy.com/en/courses/learn-the-command-line
- Git YouTube tutorials: https://www.youtube.com/playlist?list=PL6gx4Cwl9DGAKWClAD_iKpNC0bGHxGhcx

- Command Line Crash Course: http://cli.learncodethehardway.org/book/

Bonus 1: a faster way of setting up a Git repository (using clone command)

There's a faster way of creating a Git repository (both on GitHub and on your computer). Follow these steps:

1. Click on the + (plus) in the navigation bar and select "New repository".
2. Add a name for a project.
3. Important: check *Initialize this repository with a README*.
4. Click on "Create repository".
5. In the right sidebar, under *HTTPS clone URL*, copy that URL.
6. Open Git Shell or Terminal on your computer and type in `git clone` + the URL that you copied. It should look something like this: `git clone https://github.com/ramuta/test.git`.
7. As you can see, a new project folder was created. This folder is already initialized into Git and you can also see the readme.md file (the same as on GitHub. Now you can change something in this folder and use commands `git add .`, `commit` and `push` to upload the changes to GitHub.

Bonus 2: SourceTree

Instead of using Git through a command line (terminal/bash/shell), you can use it via graphic user interface. For this, you need SourceTree. Read a short tutorial here.

Bonus 3: GitKraken

Another Git GUI tool is GitKraken. You can download and try it out here: https://www.gitkraken.com/.

You can also take a look at this GitKraken tutorial: https://www.youtube.com/watch?v=f0y_xCeM1Rk.