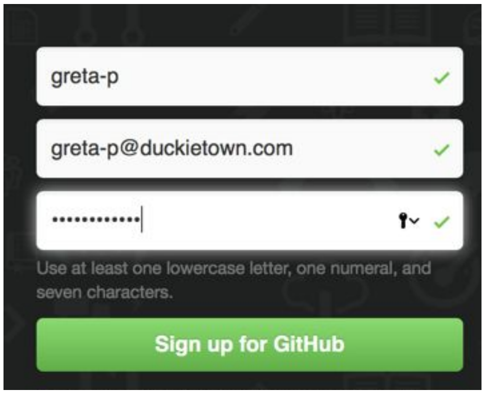
/Step 1.9 - GitHub basics

Owner: Hang Zhao

Part 1. Get an account

Our example account is the following:

/cat-cd Github : greta-p
greta-p@duckietown.com (redirects to Andrea)
greta-p@censi.org



greta-p ✓

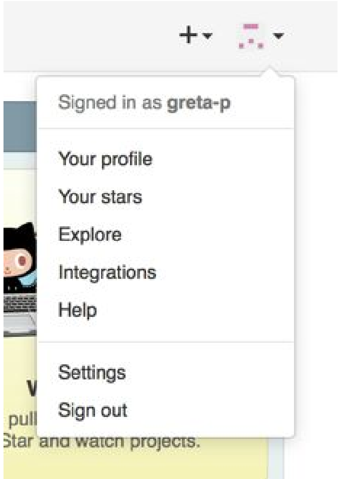
greta-p@duckietown.com ✓

••••••••••••• ✓

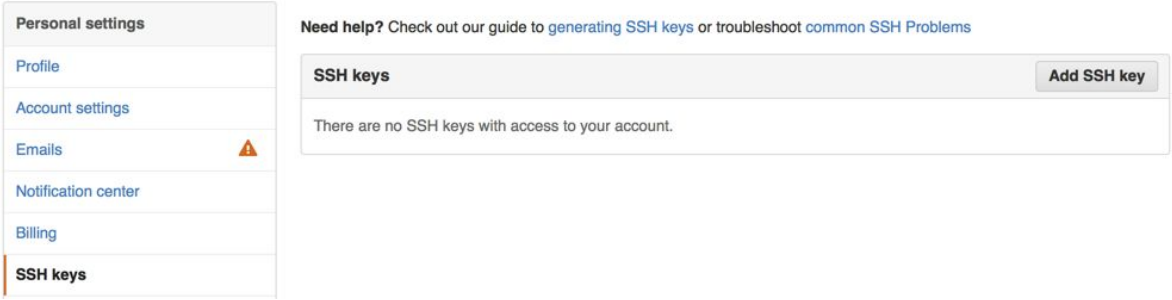
Use at least one lowercase letter, one numeral, and seven characters.

Verify email

Go to settings



Part 2. Set up SSH Key of your machine (car/laptop)



Need help? Check out our guide to generating SSH keys or troubleshoot common SSH Problems

There are no SSH keys with access to your account.

Generate the ssh key by:

```
$ ssh-keygen -h
```

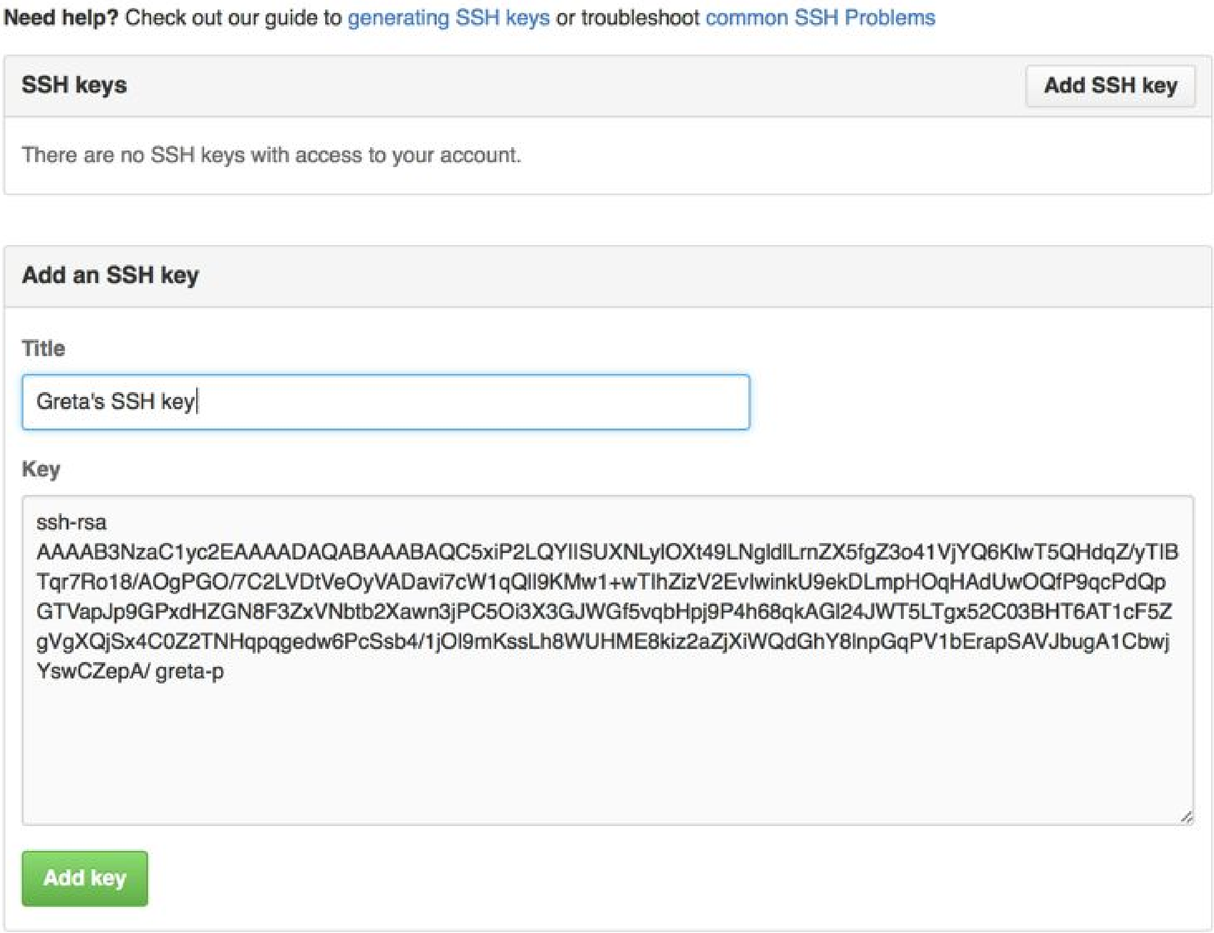
You will be asked to name the ssh key file:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): /home/ubuntu/.ssh/<your name>@<car>
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/<your name>@<car>
Your public key has been saved in /home/ubuntu/.ssh/<your name>@<car>.pub
The key fingerprint is:
4c:f8:c4:72:57:ba:ce:f1:e6:97:13:0d:27:39:43:ef ubuntu@<car>
The key's randomart image is:
+--[ RSA 2048]----+
|            .    |
|      o   o  .   |
|     o = o  . o  |
|      B . .  * o |
|       S o    O  |
|        o o  . E |
|         o o  o  |
|          o  +   |
|           .. .  |
+-----------------+
```

copy the public key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC5xiP2LQYIISUXNLylOXt49LNgldILrnZX5fg
Z3o41VjYQ6KIwT5QHdqZ/yTIBTqr7Ro18/AOgPGO/7C2LVDtVeOyVADavi7cW1qQIl9K
Mw1+wTIhZizV2EvIwinkU9ekDLmpHOqHAdUwOQfP9qcPdQpGTVapJp9GPxdHZGN8F
3ZxVNbtb2Xawn3jPC5Oi3X3GJWGf5vqbHpj9P4h68qkAGI24JWT5LTgx52C03BHT6AT
1cF5ZgVgXQjSx4C0Z2TNHqpqgedw6PcSsb4/1jOI9mKssLh8WUHME8kiz2aZjXiWQdG
hY8InpGqPV1bErapSAVJbugA1CbwjYswCZepA/ greta-p
```

Copy and paste the keys to GitHub SSH keys setup page:



Need help? Check out our guide to generating SSH keys or troubleshoot common SSH Problems

SSH keys

There are no SSH keys with access to your account.

Add an SSH key

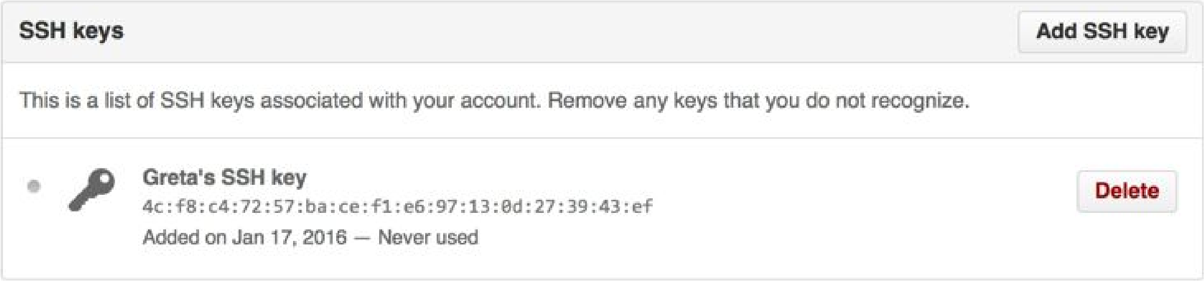
Title

Greta's SSH key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC5xiP2LQYIISUXNLylOXt49LNgldILrnZX5fgZ3o41VjYQ6KIwT5QHdqZ/yTIB
Tqr7Ro18/AOgPGO/7C2LVDtVeOyVADavi7cW1qQIl9KMw1+wTIhZizV2EvIwinkU9ekDLmpHOqHAdUwOQfP9qcPdQp
GTVapJp9GPxdHZGN8F3ZxVNbtb2Xawn3jPC5Oi3X3GJWGf5vqbHpj9P4h68qkAGI24JWT5LTgx52C03BHT6AT1cF5Z
gVgXQjSx4C0Z2TNHqpqgedw6PcSsb4/1jOI9mKssLh8WUHME8kiz2aZjXiWQdGhY8InpGqPV1bErapSAVJbugA1Cbwj
YswCZepA/ greta-p
```

A new key is added:



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Greta's SSH key
4c:f8:c4:72:57:ba:ce:f1:e6:97:13:0d:27:39:43:ef
Added on Jan 17, 2016 — Never used

Creation of SSH config file for your machine

First, create the file ~/.ssh/config.

```
$ touch ~/.ssh/config
```

Then add a line “IdentityFile <path>” using “vim”:

```
$ vim ~/.ssh/config
```

At the end this is the expected result:

save

```
$ cat ~/.ssh/config
IdentityFile ~/.ssh/FILENAME_YOU_USED_BEFORE
```

Now repeat what you just did for your duckiebot on the laptop **as well,** changing <car> to <your laptop>.

Now check that this works:

```
$ ssh -T git@github.com
Warning: Permanently added the RSA host key for IP address '192.30.252.128' to the list of known hosts.
Hi greta-p! You've successfully authenticated, but GitHub does not provide shell access.
```

*** If you don’t see this, stop.

Part 3. Setting up global configurations for Git

On your laptop, set up git, with the following commands:

```
$ git config --global user.email "<email>"
$ git config --global user.name "<name>"
```

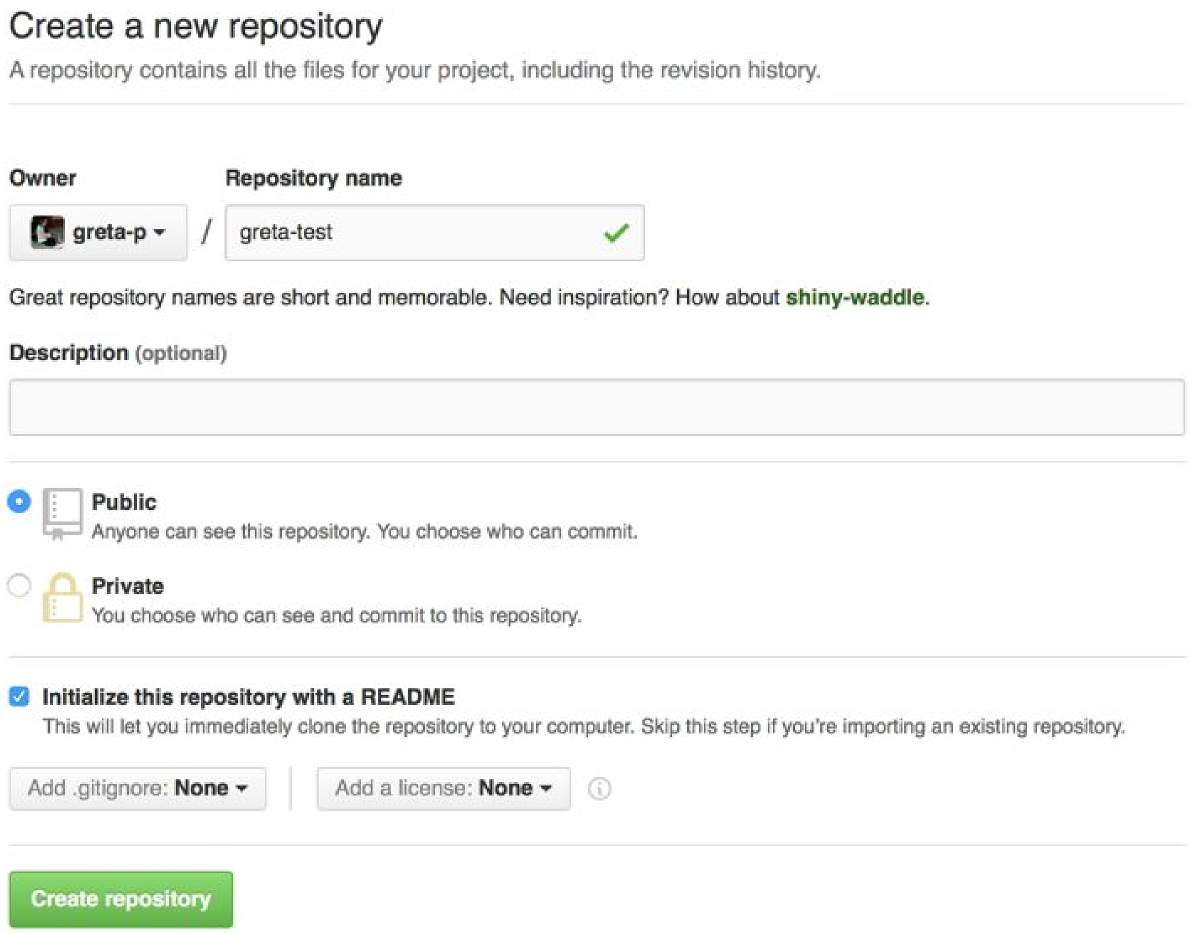
The email will be public in our repository's history. You should use either the @mit.edu email or <nickname>@duckietown.com.

Also do this, and it doesn't matter if you don't know what it is:

```
$ git config --global push.default simple
```

Part 5. Creating and committing to a new repository

Here we talk about how to create a test repository.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: greta-p / Repository name: greta-test ✓

Great repository names are short and memorable. Need inspiration? How about **shiny-waddle**.

Description (optional)

● **Public**
Anyone can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

☑ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None**

```
$ git clone git@github.com:greta-p/greta-test.git greta-test
```

```
Cloning into ...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.ls
Checking connectivity... done.
```

Then you can add a file, commit, and push.

Do not proceed if at any point you had to enter a password.

Part 4. First commit to our repository

Now you will make change to files and commit for the first time.

Please modify the scuderia.yaml file that you can find in the Duckietown/Software repository.

Remember to clone using the git address:

```
$ git clone git@github.com:duckietown/Software.git duckietown
```

These modifications can be done in the “master” branch.

Add a line into the “scuderia.yaml”, which looks like this:

```
<robot>:
    name: <nickname>
```

To check Git is aware of the file change:

```
$ git status
```

e

Then add the file:

```
$ git add scuderia.yaml
```

Commit your change with message (-m, required):

```
$ git commit -m "<robot> was setup"
```

Note: Be sure to NOT copy and paste - the quotes should be standard quotes and not directional quotes. I know that people will copy and past anyway, but in general copy and paste will lead to suffering and despair.gif

Then push the change:

```
$ git push
```

Congratulations, you have made the first commit to the remote repository on GitHub!

Appendix. Troubleshooting

Problem 1: https instead of ssh:

Symptom:

```
$ git push
Username for 'https://github.com':
```

Diagnosis: the “remote” is not correct

If you do git remote you get:

```
$ git remote -v
origin  https://github.com/duckietown/Software.git (fetch)
origin  https://github.com/duckietown/Software.git (push)
```

Solution:

```
$ git remote remove origin
$ git remote add origin git@github.com:duckietown/Software.git
```

Expectation:

```
$ git remote -v
origin  git@github.com:duckietown/Software.git (fetch)
origin  git@github.com:duckietown/Software.git (push)
```

Problem 2: git push complains about upstream

Symptomp:
fatal: The current branch devel has no upstream branch.

Solution:

```
$ git push --set-upstream origin <current branch>
```

ignore greyed out

Update the scuderia.yaml file

Find the MAC address of the interface and add it to the Scuderia file:

```
$ ifconfig | grep wlan0
```

Note the string after “HWaddr”: that is the MAC address.