FutureMakers Trainers' Manual

Fintan Nagle
September 2019

Verison 1.0.1

Workshop overview

Part A: basic coding in CodeSkulptor (online platform usable from any computer with zero configuration).

Part B: coding in the Jupyter notebook, using Amazon machine learning services like translation and speech synthesis.

Part C: making a simple interactive web app using the Flask server in Python, and deploying it to Heroku.

Teaching style

Being open about difficulty

Programming is difficult and we musn't give students the impression it's too easy; they *will* get stuck, and that's fine. Show them useful ways to get un-stuck: asking fellow students, searching online (using proper keywords such as the name of the relevant language), searching on Stack Overflow, asking an instructor, posting on Stack Overflow.

Encouraging mistakes and questions

Mistakes are how we learn: there is no such thing as a silly question. Encourage students to ask as many questions as possible.

It's OK to be confused

Students can be worried or disheartened by the amount of things they don't know or the amount of questions there are to ask. It's very important to get across to students that *even professional programmers are always Googling for answers and asking their colleagues*. It really helps students if they know that even people who have been successfully programming for years often end up lost and confused - and that the answer is often only just a question or a web search away.

Setting up the workshop

Translation

Contact AI For Good **before** you start translating the workshop manuals into another language! It may have been done already; otherwise, we can help you with the translation and also translate the images and diagrams for you.

Connectivity

You must be able to access the services we use in the workshop through whatever firewalls or network security systems are present in the venue.

- Heroku
- Pip
- CodeSkulptor
- AWS machine learning services
- Google (so that students can do research)

Internet connectivity is vital as both days need good connections - for CodeSkulptor and AWS machine learning services (Day 1) and for Heroku deployments (Day 2). This means that good wifi is needed and firewall ports may have to be opened to allow services to work. If you run through one exercise from CodeSkulptor, one machine learning test with the Jupyter notebook (translation, etc) and one deploy to Heroku, this is a good way to test that everything works.

Equipment

Laptops must have mice: it is impossible to program effectively with a trackpad, especially if you're unfamiliar with it.

Each student should be given (or should bring) a USB stick **containing all the FutureMakers files including cheat sheets and instructions** so that they can take their work home with them.

Information for students

Students will require

· The FutureMakers workshop manual - *it is absolutely vital to print this. If you only print one thing, print this.* Students work much better when they have the instructions always visible on paper next to them.

· Cheat sheets: HTML, CSS, Python - *these are valuable for students to take home, but if you want to save paper you can print one per table and make sure they get the PDFs on their USB sticks*

Equipment

Students will require

- Pens and paper for making notes and drafting ideas
- A laptop with a mouse
- A USB stick containing the FutureMakers folder

The FutureMakers folder

This is the main folder which students work in.

Always download a fresh version from https://github.com/fusionlove/futuremakers (click **Clone or download** then **Download Zip).** The instructions and code are constantly being updated, so get a fresh version.

Do not modify this folder except to remove instructions in languages that you do not need.

Software to install on laptops

For all command line operations, if on Windows, use the Anaconda Command Prompt, not the normal command prompt. On OS X, just use Terminal as normal.

- · Anaconda Python environment
- Make sure you install the Python 3.7 version
- https://www.anaconda.com/download/
- · Git
- Windows: https://git-scm.com/download/win
- OS X: see here
- https://gist.github.com/derhuerst/1b15ff4652a867391f03
- · Heroku command line interface (CLI)
- https://devcenter.heroku.com/articles/heroku-cli
- · Make sure both Firefox and Chrome are installed

- (so that you can run your web app in one, and do Google searches or look up documentation in the other)
- ·   Sublime Text
- https://www.sublimetext.com/

Example install instructions (these may differ for you)

- Install Anaconda 3.7
- Check that the Anaconda Prompt is working
- Install Git
- Check that git works from the Anaconda Prompt (it may need to be added to the PATH)
- Install Heroku CLI
- Check that heroku command works from the Anaconda Prompt (may also need to be added to PATH)
- Make sure either Firefox or Chrome are available
- Install Sublime Text
- Check that the FutureMakers demo Jupyter notebook is working (https://github.com/fusionlove/futuremakers)
- Check that an application can be deployed to Heroku
- Finally, create a Heroku account for each student (no CC required, just email validation)

Pre-workshop laptop test

It is absolutely vital to perform every part of this test to prevent disruption of the workshop.

- Check that all previous FutureMakers folders have been deleted from the desktop.
- Download a fresh FutureMakers folder, unzip it and place it on the desktop.
- Open an Anaconda notebook and run some code.
- Open the FutureMakers demo notebook (where we test the AL/ML services) and run *all* the cells to check they work. Make sure the speech synthesis call actually plays the sound.
- Open the Anaconda prompt and check that Heroku and Git are accessible.
- Check that Firefox and Chrome are installed and working.
- Check that Sublime Text is installed and working.

- Make sure students have mice with two buttons and a scroll wheel.
  If using Mac OS X, make sure the mice are set up to be able to right-click (in System Preferences).
- Check wifi connectivity: If you run through one exercise from CodeSkulptor, one machine learning test with the Jupyter notebook (translation, etc) and one deploy to Heroku, this is a good way to test that everything works.

Other setup

· Each student should have a number in the workshop. This should be filled in on the inside front cover of the workshop manual.
· The path to the FutureMakers folder (for example C:\Users\Code\Desktop\FutureMakers) should be filled in on the inside front cover of the workshop manual.
· Each student must have a Heroku account (username and password) for which the email address has been validated. These should be filled in on the inside front cover of the workshop manual.

Final setup

1. Preparing for the session - make sure:
   - There are enough laptops and mice for each laptop;
   - The room has enough sockets, perhaps bring some extension leads. Assume laptops have to be plugged in at all times;
   - All the necessary software is installed on learners' laptops (Anaconda, Git, Firefox, Git, Sublime Text, Heroku);
   - There are enough USB sticks for every learner to save and take home what they've been working on during the day;
   - The room layout allows collaboration in groups and that there's a station for trainers;
   - There are enough pens and paper for learners to take notes;
   - The workshop manual (Git > Instructions), the Python cheat sheet, the personalised sheet (only for day 2) are printed;
   - The breakfast and lunch food is organised at least a day in advance.
   - Connectivity has been checked: If you run through one exercise from CodeSkulptor, one machine learning test with the Jupyter notebook (translation, etc) and one deploy to Heroku, this is a good way to test that everything works.

2. Technical content - be always ready to explain:
    - A: Indentation and spaces
    - A: The concept of loops
    - A: What a bug is and how to debug
    - A: Syntax errors
    - B: Jupiter notebook environment (show rather than explain)
    - B: Cells, function and variables
    - B: What a library is
    - B: finding a photo - tell them not to spend too much time on it
    - B: emphasise the examples of combining different AI services (e.g. the output of the object recognition can be used as an input for voice synthesis, etc.)
3. Non-technical part:
    - SDG's: focus on the subset of goals
    - Ethics: make sure you illustrate it using the youtube example
    - HCD: emphasise the process and why it matters
    - Ideation: don't forget to give them a list of questions for good storytelling

Example day

It is very important to make sure that students have enough time to work on the programming parts of the workshop (Part A, Part B, Part C) as well as the design/soft skills parts.

It is ideal to have 2.5 hours for Part A and 2.5 hours for Part B.

On day 2, we should aim to have at least 5 hours dedicated to Part C.

This is how the two days of a FutureMakers workshop might look:

8:45 - 9:00	Students arrive; 15 mins breakfast
9:00 - 9:15	Icebreaker and introduction
9:15 - 9:30	UN SDG's video
9:30 - 12:00	Workshop session 1
12:00 - 12:45	Lunch
12:45 - 13:00	AI Ethics Game & AI for Good Examples video
13:00 - 15:00	Workshop session 2
15:00 - 15:15	Briefing on talks and design session *Inform students to copy their work onto a USB stick*
15:15 - 1600	Design session & students prepare talks
1600 - 17:00	Student presentations

Note: it's very important to ask students to copy their work onto a USB stick, write down the address of their Heroku app, and save their Heroku login details *before* the group work and presentations which close the session. Once students start working together and presenting, they don't want to go back to the computers or copy files on to a USB stick, so make sure this is done at the end of the programming workshops when everyone is still relatively calm.

Key checklist

These items must be checked and verified working at least one week before the course is due to run. If software is properly installed, fixing issues can take several days.

We require a positive YES response to each one of these questions at least one week before programme delivery. This must be true for all the classroom computers.

- Has the Anaconda Python distribution been installed?
- Has Git been installed?
- Has Heroku been installed?
- Can you run Git from the Anaconda Prompt?
  git --version
- Can you run Heroku from the Anaconda Prompt?
  heroku login
- Has the passwords.py file been placed in the FutureMakers folder next to futuremakers.py (in python-notebooks)?
- Has the FutureMakers folder been placed on the desktop?
- Has every cell in the FutureMakers API demo notebook been run without error?
- Has the sample app been launched locally using the run file (Mac) or run.bat file (Windows)? Instructions are in the course manual.
- Has the sample app running locally been visited in a browser (see manual) and confirmed to be working?
- Has a separate Heroku account been created for every student, and the email confirmed?
- Has the sample app been deployed to Heroku? (See manual.)
- Has the sample app been visited on the Web and confirmed to be working?

Common bugs and problems

· Can't access Git?

Make sure it's on the PATH (set in Windows control panel or OS X/Linux .bash_profile) and
open a new command window

·   Can't access Heroku?
Make sure it's on the PATH (set in Windows control panel or OS X/Linux .bash_profile) and
open a new command window. The PATH is something like "C:\Program
Files\Heroku\bin" - check where Heroku is installed on your machine.

·   Python module not installed?
Open an Anaconda command window and use **conda install module** or **pip install module**

·   Can't access conda or pip?
Make sure the Anaconda3 folder, and the Scripts folder within it, are on the PATH.

Setting up Boto:

```
import boto3
client = boto3.client(
    's3',
    aws_access_key_id=ACCESS_KEY,
    aws_secret_access_key=SECRET_KEY,
    aws_session_token=SESSION_TOKEN,
)
```

Setting up a bucket

We use an Amazon Web Services Simple Storage Service (S3) bucket to upload images so that
AWS can use machine learning services to process them.

To set up the bucket, log into the AWS Console and make a new bucket. Note down the
bucket name and the region (pick a region close to you for extra speed). Be sure to untick the
box marked "restrict all public access to this bucket".

The AWS region codes are as follows:

US East (Ohio) us-east-2   US West (Oregon) us-west-2
US East (N. Virginia) us-east-1   Asia Pacific (Hong Kong) ap-east-1
US West (N. California) us-west-1   Asia Pacific (Mumbai) ap-south-1

Asia Pacific (Osaka-Local) ap-northeast-3

Asia Pacific (Seoul) ap-northeast-2

Asia Pacific (Singapore) ap-southeast-1

Asia Pacific (Sydney) ap-southeast-2

Asia Pacific (Tokyo) ap-northeast-1

Canada (Central) ca-central-1

China (Beijing) cn-north-1

China (Ningxia) cn-northwest-1

EU (Frankfurt) eu-central-1

EU (Ireland) eu-west-1

EU (London) eu-west-2

EU (Paris) eu-west-3

EU (Stockholm) eu-north-1

Middle East (Bahrain) me-south-1

South America (Sao Paulo) sa-east-1

You will need to make an IAM (Identity and Access Management) user in the IAM console and set its permissions as follows. Then copy its access key and secret access key (you only get one chance to look at the latter.

Necessary permissions:
- AmazonS3FullAccess
- AmazonRekognitionReadOnlyAccess
- TranslateReadOnly
- AmazonPollyReadOnlyAccess
- S3FullAccess

Now put the two keys, the bucket name and the bucket region code into the passwords.py file **and place this in two places in the Future Makers folder as described in the readme.** An example passwords.py file looks like this:

```
access_key = 'AKIHHFGFHJHJFU6H6'
secret_access_key = 'lV4hjjfbghfgvbhfghbj'
bucket_name = 'futuremakers-2716'
s3_region = 'eu-west-2'
```

Making Heroku accounts

Each student will need a Heroku account. These are free to set up; trainers should make them all in advance and then give students the passwords. Heroku will complain if you make too many accounts too quickly or from similar email addresses, so do this a few days in advance. If you have Gmail you can use the plus operator to general more email addresses - for example, james+heroku1@gmail.com and james+heroku2@gmail.com will both go to james@gmail.com.

Selecting students for Phase 2

· Self-starting
· Asking questions
· Making mistakes
· Focussing on the material

Deployment steps for FutureMakers HQ (don't need to be done by organisers or trainers)

These steps must be followed before commiting the project to the FutureMakers Git repo. The correct Git repo is https://github.com/fusionlove/futuremakers. There is another out-of-date repo under the Sage organisation: don't use this one.

- Make sure the .git folder is deleted from the sample-web-app folder before it's released to students. This folder is put in for Heroku testing, but cannot be present when the project is committed (Git will refuse to commit that folder).
- Make sure there are NO PASSWORDS IN THE PROJECT before you commit. Any files called passwords.py will NOT be committed, but if you have pasted passwords anywhere else **this is a really serious problem as they may be stolen** (and also AWS will detect that they have been leaked and ask us to immediately delete ALL our access keys).