

Things* I wish I knew* before starting an industry job

TODO:explain *

Trevor Sullivan

February 23, 2018

Background in 30 seconds, GO!

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool
- Double major in linguistics/CS

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool
- Double major in linguistics/CS
- Masters program? hell yeah!

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool
- Double major in linguistics/CS
- Masters program? hell yeah!
- Internship with ayfie on fuzzy prefix search

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool
- Double major in linguistics/CS
- Masters program? hell yeah!
- Internship with ayfie on fuzzy prefix search
- Job?

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool
- Double major in linguistics/CS
- Masters program? hell yeah!
- Internship with ayfie on fuzzy prefix search
- Job?
- yes

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool
- Double major in linguistics/CS
- Masters program? hell yeah!
- Internship with ayfie on fuzzy prefix search
- Job?
- yes
- search for lawyers

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool
- Double major in linguistics/CS
- Masters program? hell yeah!
- Internship with ayfie on fuzzy prefix search
- Job?
- yes
- search for lawyers

Background in 30 seconds, GO!

- Came to UofAZ as a music/linguistics student
- Realized music is a dumb thing to spend \$40,000 on
- Hey computers are cool
- Double major in linguistics/CS
- Masters program? hell yeah!
- Internship with ayfie on fuzzy prefix search
- Job?
- yes
- search for lawyers

TIME

Overview

Overview

- I worked as an intern for ayfie for my hlt project on fuzzy prefix search (cf October 2016 Data Science Meetup)

Overview

- I worked as an intern for ayfie for my hlt project on fuzzy prefix search (cf October 2016 Data Science Meetup)
- I was hired afterwards as a Professional Service Engineer, to start in May

Overview

- I worked as an intern for ayfie for my hlt project on fuzzy prefix search (cf October 2016 Data Science Meetup)
- I was hired afterwards as a Professional Service Engineer, to start in May
- I took some notes on things* that I found particularly frustrating to not know* ahead of time

Overview

- I worked as an intern for ayfie for my hlt project on fuzzy prefix search (cf October 2016 Data Science Meetup)
- I was hired afterwards as a Professional Service Engineer, to start in May
- I took some notes on things* that I found particularly frustrating to not know* ahead of time

Overview

- I worked as an intern for ayfie for my hlt project on fuzzy prefix search (cf October 2016 Data Science Meetup)
- I was hired afterwards as a Professional Service Engineer, to start in May
- I took some notes on things* that I found particularly frustrating to not know* ahead of time

Starting any new job comes with a learning phase that lasts weeks or months, but knowing* about some things* ahead of time will help you make the transition more easily

On things and knowing

- Things

On things and knowing

- Things
 - This is just an excuse for me to bring up **impostor syndrome** because it didn't fit anywhere else in the talk

On things and knowing

- Things
 - This is just an excuse for me to bring up **impostor syndrome** because it didn't fit anywhere else in the talk
 - it never goes away

On things and knowing

- Things
 - This is just an excuse for me to bring up **impostor syndrome** because it didn't fit anywhere else in the talk
 - it never goes away
 - **NOW** is the time to practice your confidence and self esteem

On things and knowing

- Things
 - This is just an excuse for me to bring up **impostor syndrome** because it didn't fit anywhere else in the talk
 - it never goes away
 - **NOW** is the time to practice your confidence and self esteem
- Knowing

On things and knowing

- Things
 - This is just an excuse for me to bring up **impostor syndrome** because it didn't fit anywhere else in the talk
 - it never goes away
 - **NOW** is the time to practice your confidence and self esteem
- Knowing
 - knowing *about*

On things and knowing

- Things
 - This is just an excuse for me to bring up **impostor syndrome** because it didn't fit anywhere else in the talk
 - it never goes away
 - **NOW** is the time to practice your confidence and self esteem
- Knowing
 - *knowing about*
 - *knowing knowing*

Table of contents

1. Cluster computing | Functional programming
2. Continuous Integration | Testing
3. Docker, containerization, packaging | Git/Git-flow

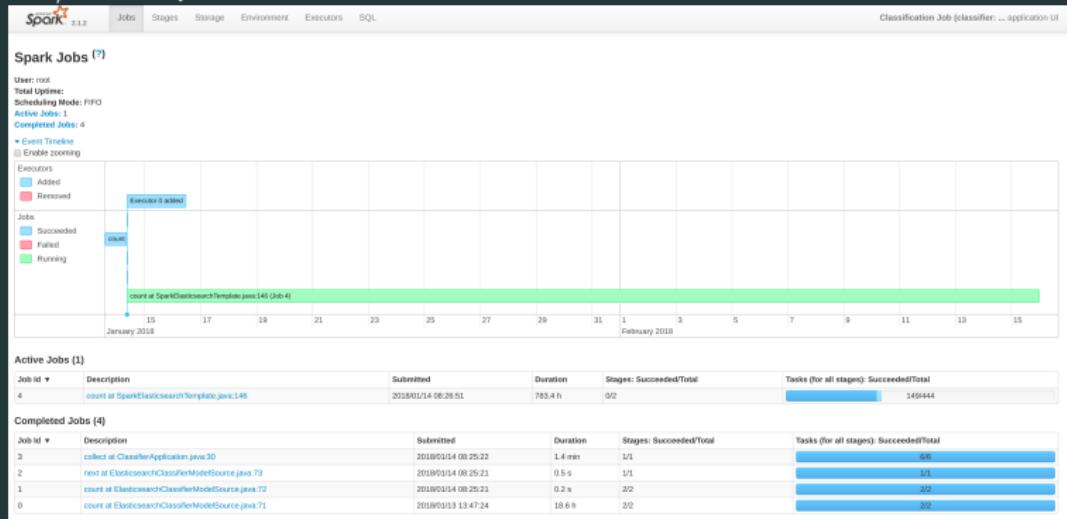
Cluster computing | Functional programming

Be aware of: Cluster computing

Spark/Hadoop

Be aware of: Cluster computing

Spark/Hadoop



A way to distribute data and computation to make your stuff scalable.

Abstracts away parallelism and data structure so you can focus on functionality

Do it now: Functional programming

Cluster computing relies on functional programming, so you should definitely be familiar with it.

These two code samples do the same thing.

```
1 for i, row in big_everything.iterrows():
2     if row.type == "Listings_City":
3         big_everything[i, 'jsonstuff'] = re.sub("displayValue\":(.*\")", lambda m:
4             case_fixer(m, True), row.jsonstuff)
5     else:
6         big_everything[i, 'jsonstuff'] = re.sub("displayValue\":(.*\")", lambda m:
7             case_fixer(m, False), row.jsonstuff)
```

Do it now: Functional programming

Cluster computing relies on functional programming, so you should definitely be familiar with it.

These two code samples do the same thing.

```
1 for i, row in big_everything.iterrows():
2     if row.type == "Listings_City":
3         big_everything[i, 'jsonstuff'] = re.sub("displayValue\":(.*\")", lambda m:
4             case_fixer(m, True), row.jsonstuff)
5     else:
6         big_everything[i, 'jsonstuff'] = re.sub("displayValue\":(.*\")", lambda m:
7             case_fixer(m, False), row.jsonstuff)
```

10 hours

Do it now: Functional programming

Cluster computing relies on functional programming, so you should definitely be familiar with it.

These two code samples do the same thing.

```
1 for i, row in big_everything.iterrows():
2     if row.type == "Listings_City":
3         big_everything[i, 'jsonstuff'] = re.sub("displayValue\":(.*\")", lambda m: case_fixer(m, True), row.jsonstuff)
4
5     else:
6         big_everything[i, 'jsonstuff'] = re.sub("displayValue\":(.*\")", lambda m: case_fixer(m, False), row.jsonstuff)
```

10 hours

```
1 def json_stuff_fixer(row):
2     if row.type == "Listings_City":
3         return re.sub("displayValue\":(.*\")", lambda m: case_fixer(m, True), row.jsonstuff)
4
5     else:
6         return re.sub("displayValue\":(.*\")", lambda m: case_fixer(m, False), row.jsonstuff)
7
8
9 big_everything['jsonstuff'] = big_everything.apply(json_stuff_fixer, axis=1)
```

Do it now: Functional programming

Cluster computing relies on functional programming, so you should definitely be familiar with it.

These two code samples do the same thing.

```
1 for i, row in big_everything.iterrows():
2     if row.type == "Listings_City":
3         big_everything[i, 'jsonstuff'] = re.sub("displayValue\":(.*\")", lambda m: case_fixer(m, True), row.jsonstuff)
4
5     else:
6         big_everything[i, 'jsonstuff'] = re.sub("displayValue\":(.*\")", lambda m: case_fixer(m, False), row.jsonstuff)
```

10 hours

```
1 def json_stuff_fixer(row):
2     if row.type == "Listings_City":
3         return re.sub("displayValue\":(.*\")", lambda m: case_fixer(m, True), row.jsonstuff)
4
5     else:
6         return re.sub("displayValue\":(.*\")", lambda m: case_fixer(m, False), row.jsonstuff)
7
8
9 big_everything['jsonstuff'] = big_everything.
    apply(json_stuff_fixer, axis=1)
```

4 minutes

Functional Programming

People who write functional/distributed database libraries are smarter than you.

Functional Programming

People who write functional/distributed database libraries are smarter than you.

If a functional idiom is available, **use it**.

Functional Programming

People who write functional/distributed database libraries are smarter than you.

If a functional idiom is available, **use it**.

It requires you to think a different way from the typical OOP paradigm, but there is great benefit and a good mental exercise besides.

Most important things to learn:

- lambda
- foreach
- apply
- map
- filter
- reduce

Continuous Integration | Testing

Be aware of: Continuous Integration

CI is a way of making sure that code you commit or push works.

Be aware of: Continuous Integration

CI is a way of making sure that code you commit or push works.

In commercial applications, you will probably use Jenkins

Be aware of: Continuous Integration

CI is a way of making sure that code you commit or push works.

In commercial applications, you will probably use Jenkins

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Open Blue Ocean', 'Credentials', and 'New View'. Below that is a 'Build Queue' section stating 'No builds in the queue.' Under 'Build Executor Status', it shows two entries: 'master' (with 1 idle, 2 available, 3 offline) and 'devint' (with 1 idle, 1 available, 1 offline). The main area displays a table of builds:

S	W	Name	Last Success	Last Failure	Last Duration	Fav
1	Cloud	Aude Platform	N/A	N/A	N/A	☆
2	Cloud	docker-prune	20 hr - #114	N/A	23 sec	☆
3	Cloud	legacy-searcher	N/A	N/A	N/A	☆
4	Cloud	L1Locate	2 days 8 hr - #50	N/A	4 min 54 sec	☆
5	Cloud	search-plugins	8 min 14 sec - log	N/A	1 sec	☆
6	Cloud	search-plugins-old	1 yr 0 mo - #8	N/A	57 sec	☆
7	Cloud	Ultrakt Backend	10 hr - #82	N/A	1 min 26 sec	☆
8	Cloud	Ultrakt Frontend	9 hr 10 min - #81	2 days 8 hr - #50	4 min 47 sec	☆

At the bottom right, there's a legend for RSS feeds: RSS for all jobs, RSS for failures, and RSS for just latest builds. The footer includes links for 'Page generated: Fri 16, 2018 12:43:25 AM CEST', 'REST API', and 'Jenkins ver. 2.88.3'.

Be aware of: Continuous Integration

CI is a way of making sure that code you commit or push works.

In commercial applications, you will probably use Jenkins

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Open Blue Ocean', 'Credentials', and 'New View'. Below that is a 'Build Queue' section stating 'No builds in the queue.' Under 'Build Executor Status', it shows two items: 'master' (with 1 idle, 2 available, 3 offline) and 'devint' (with 1 idle, 1 available, 1 offline). The main area displays a table of builds:

S	W	Name	Last Success	Last Failure	Last Duration	Fav
---	---	Auto Platform	N/A	N/A	N/A	☆
---	---	docker-prune	20 hr - #114	N/A	23 sec	☆
---	---	legacy-searcher	N/A	N/A	N/A	☆
---	---	L1Locate	2 days 8 hr - #50	N/A	4 min 54 sec	☆
---	---	search-plugins	8 min 14 sec - log	N/A	1 sec	☆
---	---	search-plugins-old	1 yr 0 min - #8	N/A	57 sec	☆
---	---	Ultrikit Backend	10 hr - #82	N/A	1 min 26 sec	☆
---	---	Ultrikit Frontend	9 hr 10 min - #81	2 days 8 hr - #202	4 min 47 sec	☆

At the bottom right, there's a legend for RSS feeds: RSS for success, RSS for failure, and RSS for just latest builds. The footer includes the page generation time ('Page generated: Fri 16, 2018 12:43:25 AM CEST'), the REST API link, and the Jenkins version ('Jenkins ver. 2.8.3').

CI builds your code in a consistent environment and tests it to make sure it works.

Travis

About Us Blog Status Help Trevor Sullivan 

Search all repositories 

My Repositories +

✓ Trevordts/lingbot  build pending

Duration: 9min 45 sec  Finished: a day ago

✗ artois/phonological-imaging  # 44 Duration: 21 min 3 sec  Finished: 2 years ago

Trevordts / lingbot  build pending

Current Branches Build History Pull Requests More options 

Default Branch

Branch	Builds	Passed	Time	Commit	Author	Build Status					
✓ master	27 builds	# 47 passed	■ a day ago	→ ffa416f ↵	Ubuntu						

Active Branches

Branch	Builds	Passed	Time	Commit	Author	Build Status	Build Status				
✓ hotfix/catchall	1 build	# 42 passed	■ 3 months ago	→ 6d9fc1d ↵	trevordts						
✓ develop	6 builds	# 40 passed	■ 4 months ago	→ be320a1 ↵	trevordts						
✓ feature/better-schedule	3 builds	# 38 passed	■ 4 months ago	→ c4bf5a1 ↵	trevordts						

Inactive Branches

Branch	Builds	Passed	Time	Commit	Author	Build Status	Build Status				
✓ feature/yaml	8 builds	# 27 passed	■ 4 months ago	→ 123ef4f ↵	trevordts						

Do it now: Testing

Do it now: Testing

OMG testing is great.

Do it now: Testing

OMG testing is great.

For medium-to-large size projects

1. Find bugs faster

Do it now: Testing

OMG testing is great.

For medium-to-large size projects

1. Find bugs faster
2. Find their causes faster

Do it now: Testing

OMG testing is great.

For medium-to-large size projects

1. Find bugs faster
2. Find their causes faster
3. Prevent problems from reoccurring

Do it now: Testing

OMG testing is great.

For medium-to-large size projects

1. Find bugs faster
2. Find their causes faster
3. Prevent problems from reoccurring

Do it now: Testing

OMG testing is great.

For medium-to-large size projects

1. Find bugs faster
2. Find their causes faster
3. Prevent problems from reoccurring

There are testing systems out there for every language.

Python pytest

Java junit

scala scalatest

And you can set up your environment to run tests on every commit.

How to write tests

There are three kinds of tests.

Unit tests Test a method or function

How to write tests

There are three kinds of tests.

Unit tests Test a method or function

Regression tests Duplicate known bugs

How to write tests

There are three kinds of tests.

Unit tests Test a method or function

Regression tests Duplicate known bugs

Integration tests Test interactions between modules

Unit test example

```
00     return reminder
67
68 ▼def test_check_morning(patch_datetime_morning):
69     assert datetime.datetime.now() == FAKE_TIME_MORNING
70     assert FAKE_TIME_MORNING.date() == patch_datetime_morning.next_nlprg.date.date()
71     assert FAKE_TIME_MORNING.weekday() == 4
72     assert FAKE_TIME_MORNING == FAKE_TIME_MORNING.replace(hour=10, minute=0, second=0)
73 ▼    assert patch_datetime_morning.check() == (("\n" + "Rebecca" + "\n" +
74         "presenting on\n" + "Deep Learning for NLP Best Practices\nLink: "
75         "http://ruder.io/deep-learning-nlp-best-practices/?utm_campaign=Arti
76         ficial%2BIntelligence%2BWeekly&utm_medium=email&utm_source=Arti
77         ficial Intelligence Weekly_66\n" +
78         "\n\nJoin us in Gould-Simpson 906 at 1400\n\n" +
79         "(food and coffee provided)\nSee full schedule here: " +
         "https://github.com/clulab/nlp-reading-group/wiki/FALL" +
         "-2017-Reading-Schedule"), True)
```

Unit test example

```
00     return reminder
67
68 ▼def test_check_morning(patch_datetime_morning):
69     assert datetime.datetime.now() == FAKE_TIME_MORNING
70     assert FAKE_TIME_MORNING.date() == patch_datetime_morning.next_nlprg.date.date()
71     assert FAKE_TIME_MORNING.weekday() == 4
72     assert FAKE_TIME_MORNING == FAKE_TIME_MORNING.replace(hour=10, minute=0, second=0)
73 ▼    assert patch_datetime_morning.check() == ((("NLP Reading Group today!\n" + "Rebecca" +
74         "\n" + "presenting on\n" + "Deep Learning for NLP Best Practices\nLink:\n"
75         "http://ruder.io/deep-learning-nlp-best-practices/?utm_campaign=Arti
76         ficial%2BIntelligence%2BWeekly&utm_medium=email&utm_source=Arti
77         ficial Intelligence Weekly_66\n" +
78         "\n\nJoin us in Gould-Simpson 906 at 1400\n\n" +
79         "(food and coffee provided)\nSee full schedule here: " +
         "https://github.com/clulab/nlp-reading-group/wiki/FALL" +
         "-2017-Reading-Schedule"), True)
```

Each unit test should look at one method and exemplify one edge case.

Unit test example

```
00     return reminder
67
68 ▼def test_check_morning(patch_datetime_morning):
69     assert datetime.datetime.now() == FAKE_TIME_MORNING
70     assert FAKE_TIME_MORNING.date() == patch_datetime_morning.next_nlprg.date.date()
71     assert FAKE_TIME_MORNING.weekday() == 4
72     assert FAKE_TIME_MORNING == FAKE_TIME_MORNING.replace(hour=10, minute=0, second=0)
73 ▼    assert patch_datetime_morning.check() == ((("NLP Reading Group today!\n" + "Rebecca" +
74         "\n" + "presenting on\n" + "Deep Learning for NLP Best Practices\nLink:\n"
75         "http://ruder.io/deep-learning-nlp-best-practices/?utm_campaign=Arti
76         ficial%2BIntelligence%2BWeekly&utm_medium=email&utm_source=Arti
77         ficial Intelligence Weekly 66\n" +
78         "\n\nJoin us in Gould-Simpson 906 at 1400\n\n" +
79         "(food and coffee provided)\n\nSee full schedule here: " +
         "https://github.com/clulab/nlp-reading-group/wiki/FALL" +
         "-2017-Reading-Schedule"), True)
```

Each unit test should look at one method and exemplify one edge case.

Forces you to think about how things could go wrong, and reveals them before they do.

Docker, containerization, packaging | Git/Git-flow

Be aware of: Docker, containerization, packaging

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables
- software versions

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables
- software versions
- port

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables
- software versions
- port
- data locations

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables
- software versions
- port
- data locations
- including what to store and what to keep ephemeral

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables
- software versions
- port
- data locations
- including what to store and what to keep ephemeral
- custom binaries to inject

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables
- software versions
- port
- data locations
- including what to store and what to keep ephemeral
- custom binaries to inject
- memory and cpu limits

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables
- software versions
- port
- data locations
- including what to store and what to keep ephemeral
- custom binaries to inject
- memory and cpu limits
- system integration

Be aware of: Docker, containerization, packaging

Docker is a system for dedicated containers, mini VMs, that run just one application accessed over the network.

Docker lets you manage

- environment variables
- software versions
- port
- data locations
- including what to store and what to keep ephemeral
- custom binaries to inject
- memory and cpu limits
- system integration
- all kinds of cool stuff

Using Docker

Settings go into a docker-compose.yml

```
x 1 version: '2.1'
x 2 services:
x 3   api:
x 4     image: quay.io/ayfile/platform:v1.9.0
x 5     mem_limit: ${AYFILE MEM LIMIT:-40G}
x 6     memswap_limit: ${AYFILE MEM_LIMIT:-40G}
x 7     restart: unless-stopped
x 8     stop_grace_period: 2m
x 9     environment:
x10       SPARK_WORKER_MEMORY=${PROCESSING MEM LIMIT:-24G}
x11       SPARK_EXECUTOR_MEMORY=${PROCESSING MEM_LIMIT:-24G}
x12     links:
x13       - extraction:extraction
x14       - suggest:suggest
x15       - elasticsearch:elasticsearch
x16     volumes:
x17       - ${DATA_VOLUME:-data}:/data
x18       - /application-custom.yml:/home/dev/restapp/application-custom.yml
x19   elasticsearch:
x20     image: quay.io/ayfile/elasticsearch:v5.6.3.1
x21     mem_limit: ${ELASTICSEARCH MEM LIMIT:-16G}
x22     memswap_limit: ${ELASTICSEARCH_MEM_LIMIT:-16G}
x23     ulimits:
x24       - nproc: 1024
x25       - soft:-1
x26       - hard:-1
x27     environment:
x28       - soft: 65536
x29       - hard: 65536
x30     volumes:
x31       - ${DATA_VOLUME:-data}:/data
x32   messaging:
x33     image: rabbitmq:3.6.14-management-alpine
x34     hostname: queue
x35     environment:
x36       - RABBITMQ_DEFAULT_USER=ayfile
x37       - RABBITMQ_DEFAULT_PASS=platform
x38     alerting:
x39       image: quay.io/ayfile/alerting:v1.9.0
x40     links:
x41       - messaging:messaging
x42       - elasticsearch:elasticsearch
x43       - api:api
x44     extraction:
```

And starting the system as a background service is as easy as docker-compose up

Maybe consider: yaml for configuration

Way easier than putting stuff at the top of each file, and more understandable than environment variables

```
1 # This is the default config. If you want to change something for a
2 # particular instance, make a config-custom.yml in the same folder.
3 version: 1.0
4
5 ▼features:
6 ▼▼ passive:
7 ▼▼▼ hackyHourReminder:
8   active: True
9   location: Pasco
10  time: 10
11 ▼▼ nlpgrReminder:
12   active: True
13   url: "https://raw.githubusercontent.com/wiki/clulab/nlp.reading-group/
14     Fall-2017-Reading-Schedule.md"
15   time: 10
16 ▼▼ genericReminder:
17   active: True
18   hoursBefore: 6
19   filename: "generic_schedule.csv"
20 ▼▼ active:
21   meetingInfo:
22     active: True
23     command: "next"
24     filename: "generic_schedule.csv"
25 ▼▼ addEvent:
26   active: True
27   command: "add event"
28   filename: "generic_schedule.csv"
29
30 # TODO Figure out how to get default behavior, esp for channels.
31 # make a default yaml that is common to all versions, and an unversioned
32 # yaml that overrides the defaults
33 ▼channels:
34 ▼▼ restricted: True
35 ▼▼▼ allowed:
36   - \#bottest
37   - C25Nw0Wn7
```

Maybe consider: yaml for configuration

Way easier than putting stuff at the top of each file, and more understandable than environment variables

```
1 # This is the default config. If you want to change something for a
2 # particular instance, make a config-custom.yml in the same folder.
3 version: 1.0
4
5 ▼features:
6 ▼▼ passive:
7 ▼▼▼ hackyHourReminder:
8   active: True
9   location: Pasco
10  time: 10
11 ▼▼▼ nlpgrReminder:
12   active: True
13   url: "https://raw.githubusercontent.com/wiki/clulab/nlp.reading-group/
14     Fall-2017-Reading-Schedule.md"
15   time: 10
16 ▼▼ genericReminder:
17   active: True
18   hoursBefore: 6
19   filename: "generic_schedule.csv"
20 ▼▼ active:
21   meetingInfo:
22     active: True
23     command: "next"
24     filename: "generic_schedule.csv"
25 ▼▼ addevent:
26   active: True
27   command: "add event"
28   filename: "generic_schedule.csv"
29
30 # TODO Figure out how to get default behavior, esp for channels.
31 # make a default yml that is common to all versions, and an unversioned
32 # yml that overrides the defaults
33
34 ▼channels:
35   restricted: True
36 ▼▼ allowed:
37   - \#bottest
38   - C25NW0Wn7
```

```
30
31 ▼try:
32   ...with open("config.yml", "r") as yamlfile:
33     ...cfg = yaml.load(yamlfile)
34   ▼except FileNotFoundError:
35     ...with open("default-config.yml", "r") as yamlfile:
36       ...cfg = yaml.load(yamlfile)
37
```

Do it now: Git everything!

- We are all introduced to git for collaboration

Do it now: Git everything!

- We are all introduced to git for collaboration

Do it now: Git everything!

- We are all introduced to git for collaboration
THAT'S NOT WHAT IT'S FOR
- Version Control

Do it now: Git everything!

- We are all introduced to git for collaboration
THAT'S NOT WHAT IT'S FOR
- Version Control
- Even if you don't have a repository

Do it now: Git everything!

- We are all introduced to git for collaboration
THAT'S NOT WHAT IT'S FOR
- Version Control
- Even if you don't have a repository
- Github, Gitlab, Bitbucket are nice for publishing and sharing, but you should be using git even if you don't want to share.

Do it now: Git everything!

- We are all introduced to git for collaboration
THAT'S NOT WHAT IT'S FOR
- Version Control
- Even if you don't have a repository
- Github, Gitlab, Bitbucket are nice for publishing and sharing, but you should be using git even if you don't want to share.
- `cd thing_im_working_on ; git init`

Do it now: Git Flow

For semester projects, theses, things you expect to be working on more than a week

Do it now: Git Flow

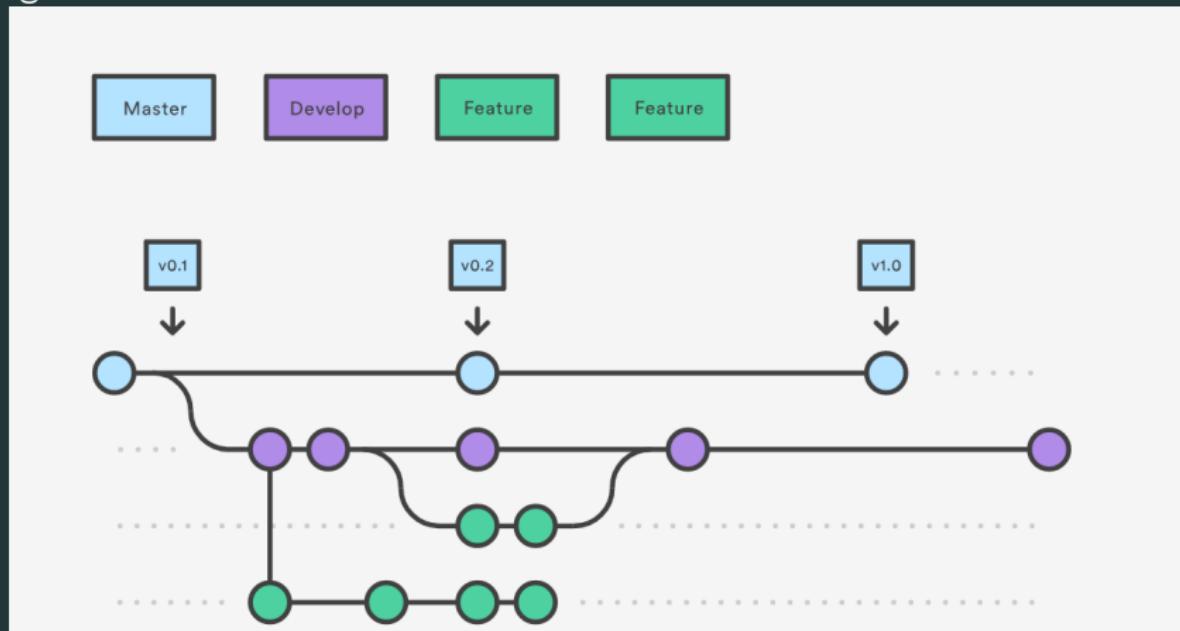
For semester projects, theses, things you expect to be working on more than a week

```
git flow init
```

Do it now: Git Flow

For semester projects, theses, things you expect to be working on more than a week

git flow init



Check time!

Team workflows | Personal task management

Be aware of: Scrum and Team Workflows

- Every company does this a different way

Be aware of: Scrum and Team Workflows

- Every company does this a different way
- scrum master

Be aware of: Scrum and Team Workflows

- Every company does this a different way
- scrum master
- sprint review

Be aware of: Scrum and Team Workflows

- Every company does this a different way
- scrum master
- sprint review
- board

Boards

Professional Services

Kanban board

Release  

TO DO 2 of 23 BLOCKED 5 of 12 IN PROGRESS 2 of 9 DONE 4 of 6

Bahr POC 5 issues

Enable deeplink functionality
                                 <img alt="grey info icon" data-bbox="

Boards

CRUNCH

AP Sprint 65

Stabilize for iConnect

3 days remaining Complete sprint

Quick filters Assignee

TO DO	IN PROGRESS	DONE
<p>✓ <input checked="" type="checkbox"/> AP-3034 DONE 2 sub-tasks Basic Spark Telemetry</p> <div><p>Fix it</p><p>None</p><p> AP-3130</p></div>		<p>Use metrics library</p> <p>3 hours, 30 minutes</p> <p> AP-3140</p> <p>Fix metric type bug</p> <p>None</p> <p> AP-3159</p>
<p>✗ <input type="checkbox"/> AP-2693 TO DO 2 sub-tasks Find out why some emails are missing thread meta.</p> <div><p>Fix it</p><p>None</p><p> AP-3130</p></div>		
<p>✓ <input type="checkbox"/> AP-3133 DONE 2 sub-tasks Branch points for forwarded emails in group 1144595606 on paul.</p> <div><p>Changelog</p><p>None</p><p> AP-3137</p></div>		
		<p>Fix it in ultrakit-frontend</p> <p>None</p> <p> AP-3131</p> <p>Investigate</p> <p>None</p> <p> AP-3163</p>

Do it now: Task Management

- People tend to overestimate their own ability to remember things

Do it now: Task Management

- People tend to overestimate their own ability to remember things
- There is a limited amount of space in your working memory, and to commit something to long term memory, you need to call it back to working memory several times

Do it now: Task Management

- People tend to overestimate their own ability to remember things
- There is a limited amount of space in your working memory, and to commit something to long term memory, you need to call it back to working memory several times
- This is a waste of brain power. Everything you have to remember distracts you.

Do it now: Task Management

- People tend to overestimate their own ability to remember things
- There is a limited amount of space in your working memory, and to commit something to long term memory, you need to call it back to working memory several times
- This is a waste of brain power. Everything you have to remember distracts you.
- Strive therefore to get things out of your head as soon as possible

Apps: Asana

The screenshot shows the Asana application interface with two main windows open.

Left Window: My Tasks

- My Favorites:** A sidebar with pinned items: lingbot, Eymachine, blog, move, asyle, and a link to "Show Recently and more..."
- Reports:** A sidebar with pinned items: Incubation, lingbot, blog, Eymachine, move, asyle, 352:UNIX & C, API, CPR, CSC Computer Organization, CompLing, CompLing, Episode 1 linkedin, Episode 2 done, Episode 3 done, Eymachine, German, History, Laptop purchased, Minimalist Machine, Moon and Planets, Persian, and Podcast.
- Task List:** A list of tasks under "Today" and "Upcoming".
 - Today tasks:
 - ✓ setup up application.yml that does email threading and reduplication and measure performance
 - ✓ set up same app on a server and minimize resource requirements
 - ✓ Look at [Dene yenisan](#) evidence
 - ✓ pay off credit card
 - ✓ pay cox
 - ✓ pay lep
 - ✓ contacts:
 - Upcoming tasks:
 - ✓ setup two amazon instances
 - ✓ ingest and process emnion dataset, measure time
 - ✓ do several runs to stabilize numbers
 - ✓ find maximum amount to data ingestible using apache email dataset
 - ✓ setup a demo based on [wikipedia](#) content
 - ✓ fill out check writing thing for cash management account
 - ✓ Look into active learning for Eymachine
 - ✓ [Xkcd smiley parenthetical](#)
 - ✓ Make new survey data into .ped format
 - ✓ make [gus's](#) changes
 - ✓ data normalization: why it's important to have input values in a known range
 - ✓ respond to this
 - ✓ make twitch streaming [settings](#) in obs
 - ✓ make twitch
 - ✓ [hackyhour](#) share photo
 - ✓ Understanding LSTM Networks -- colah's blog
 - ✓ [np-read] Pvd: Augur: Mining Human Behaviors from Fiction to Power Interactive Systems
 - ✓ Skinner box games
 - ✓ The generation fraction for the simple world

Apps: Todoist

The screenshot shows the Todoist application interface. On the left, there is a sidebar with navigation links: Inbox (5), Today (2), and Next 7 days (2). Below these are sections for Projects, Labels, and Filters. The Projects section lists several projects with their names and counts: packing list (1), Personal (1), ayfile (1), Elymachine (1), Shopping Errands (1), blog (1), Lingbot (1), move (1), Errands (1), Movies to watch (1), and Incubation (1). There is also a collapsed project for The Greats To Date (12) and a link to Add Project. At the bottom of the sidebar, it says "Archived projects".

The main area is titled "Today" and shows two tasks:

- Overdue: set up nlp on tulka (Due 29 2017, Inbox)
- call bobs (Due 29 2017, Inbox)

Below the tasks, there is a "Today" section showing 15 items and a "Add Task" button.

Apps: HabitRPG

The screenshot shows the Habitica app interface, a productivity tool that integrates RPG elements. At the top, there's a navigation bar with links for Tasks, Inventory, Shops, Party, Guilds, Group, Challenges, and Help. The top right corner displays player stats: Level 76 Mage, 11 experience, 679.64 gold, and icons for gear, a map, and user profile.

Habits: A list of negative habits to avoid. Each habit has a green plus icon, a name, a progress bar, and a red minus icon. A note at the bottom says: "Use this as a miscellaneous negative habit. Also use it to punish yourself for cheating, or for specific times in D&D challenges."

Dailies: A list of daily tasks. Each task has a blue plus icon, a name, a progress bar, and a red minus icon. Sub-tasks under "time commute" include "start timer" and "save in note".

To-Dos: A list of to-do items. Each item has a red plus icon, a name, a progress bar, and a red minus icon. Sub-tasks under "roll twice on con" include "roll twice on con" (checked), "roll twice on con", and "roll twice on con".

Rewards: A list of rewards. Each reward has a yellow plus icon, a name, a progress bar, and a yellow minus icon. Rewards include "bag of holding", "gloves of dex", "armlet of health", "Perisop of wisdom", "headband of int", "Hire Mercenary" (disabled), "Mana Potion" (disabled), and "Bark".

Skills: A row of skill icons with levels: Burst of Flames (level 10), Ethereal Surge (level 20), Earthquake (level 25), and Chilling Frost (level 40).

Apps: Trello

Hannibal/Scipio2 ☆ | The Greats to Date Free | Team Visible

Hannibal/Scipio II

- Put both files on same machine
- Line up and do meticulous primary edits
- listen and make notes for secondary edits
- listen at super speed and make notes for tertiary edits
- Upload to media host
- Update Blog
- publish to RSS
- Add a card...

in progress

- Add a card...

done

- Read relevant book
⌚ Jul 11, 2016
- Make notes for show
⌚ Jul 12, 2016 ⚖ 1
- Do show
⌚ Jul 21, 2016
- Do show
- Add a card...

Add a list...

Apps: todo.txt

```
todo.todo x
1
2
3 ▼Stuff to do now:
4   o do CV test with syllables
5   o add language names to stopwords list
6   o add code to submission
7   o add readme to submission (python notebook)
8
9   o run through naive bayes
10  o run through k-nearest neighbors
11  o add some neural network stuff to this list o run through naive bayes
12  o run through k-nearest neighbors
13  o add some neural network stuff to this list o run through naive bayes
14  o run through k-nearest neighbors
15  o add some neural network stuff to this list
16
17
18 ▼features:
19  ✓ bag of words @done (16-11-21 13:25)
20  ✓ characters present @done (16-11-23 12:30)
21  o syllables
22  o bigrams (over characters and/or syllables)
23  ✓ attested year (or none) @done (16-11-28 16:49)
24  o normalize over centuries, or continuous value
25  o
26
27 ▼Machine learning categorization:
28  ✓ vectorize definitions (features below) @done (16-11-23 12:30)
29  ✓ SVM @today done (16-11-28 14:49)
30  ✓ test different kernels @done (16-11-28 19:44)
31  ✓ Make a confusion matrix visualization @done (16-11-29 08:46)
32
33  o run through naive bayes
34  o run through k-nearest neighbors
35  o add some neural network stuff to this list
36
37
38 ▼After project is done:
```

Apps: Bulletjournal

15th

- ▲ pick up library book on hold
- Amy for lunch @ 11:30 am
- ☛ make summer bucket list
- started watching Vikings finally

16th

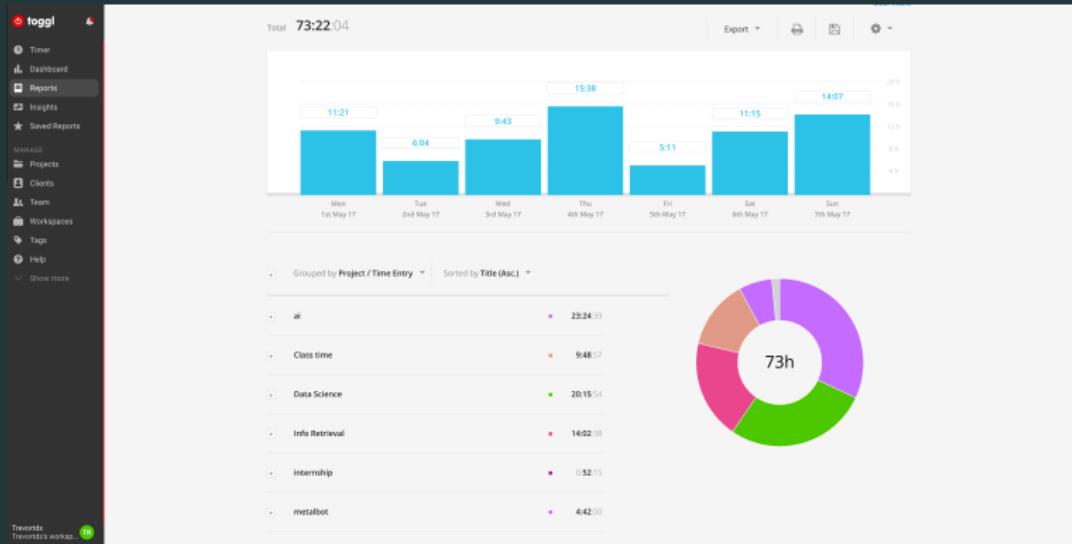
- look for UNC tickets

17th

- ▲ OT @ 3pm
- ▲ dinner at Mellow w/ Carter's @ 5:30PM
- finish Friday's blog post
- Sam said, "Mom, I can't be quiet! I'm a talker!"

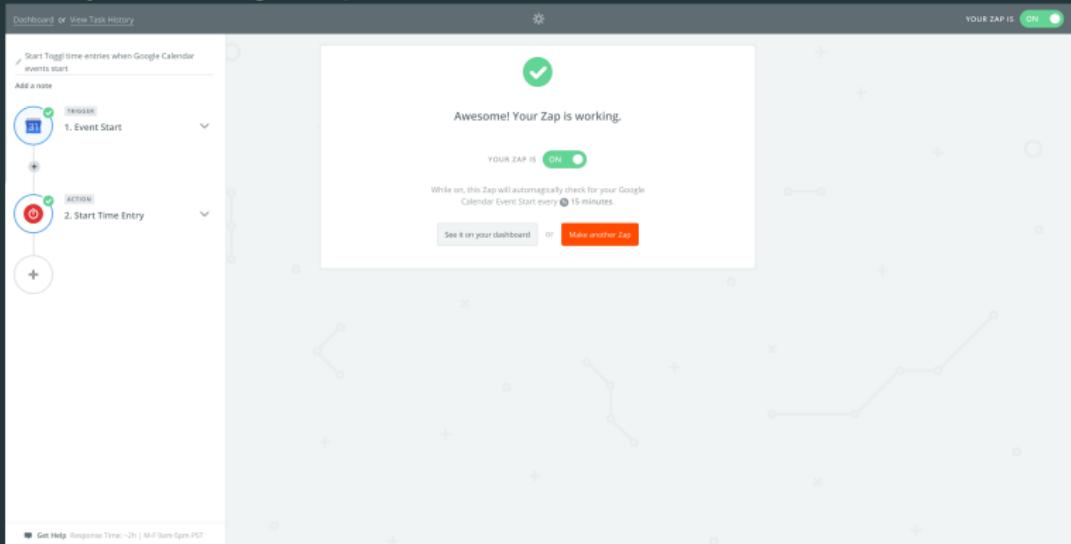
Advanced: Time tracking

Not essential, but if you already have a good system for keeping track of your tasks.



Super-Advanced: Automation

Many of the aforementioned apps have APIs that can be manipulated manually, or through Zapier



System Administration | The Command Line

Be aware of: Finer points of sysadmin

Be aware of: Finer points of sysadmin

When you're working on a server. you will need to learn some particular skills and methods to manage things

Managing volumes df -h, fdisk, mkfs

Be aware of: Finer points of sysadmin

When you're working on a server. you will need to learn some particular skills and methods to manage things

Managing volumes df -h, fdisk, mkfs

ssh tunnels and UDP ssh -L 18080:127.0.0.1:18085 -R
18086:127.0.0.1:18086 user@server

Be aware of: Finer points of sysadmin

When you're working on a server. you will need to learn some particular skills and methods to manage things

Managing volumes df -h, fdisk, mkfs

ssh tunnels and UDP ssh -L 18080:127.0.0.1:18085 -R
18086:127.0.0.1:18086 user@server

web access control nginx

Do it now: The Command Line

- Navigating a *nix filesystem

Do it now: The Command Line

- Navigating a *nix filesystem
- text editor (vim, emacs, nano)

Do it now: The Command Line

- Navigating a *nix filesystem
- text editor (vim, emacs, nano)
- tmux or screen

They may seem dumb at first, but you will soon find you can't live without it

Like browsing without tabs

tmux

File Edit View Search Terminal Help

```
"clustersPerLevel": 20,
"documentsPerCluster": 1000,
"gramianSingularVectors": 50,
"gramianThreshold": 0.14,
"maxDfFraction": 0.25,
"maxRecursionDepth": 2,
"minDf": 3
    }
},
"type": "CLUSTERING"
},
"details": {
    "applicationId": "app-20180215165851-0005",
    "description": "Execution completed successfully",
    "timestamp": "2018-02-15T17:05:54.0368Z"
},
"id": "1727374435",
"state": "SUCCEEDED"
}
]
},
"_links": {
    "self": {
        "href": "http://localhost:93/ayfie/v1/jobs"
    }
}
```

dev@docker2:/home/docker-demos/compositions/ayfie/new-wiki**ipedia\$**
[trevor] 0:ssh 1:bahr-monitor 2:bash* 3:bash-

"docker2" 23:19 15-Feb-18

That's all!