

# CODING CHALLENGE AND EFFECTIVE TEAM WORKING

Mark Dawson, Ed Bennett, Michele Mesiti, Colin Sauzé

AIMLAC CDT Induction, 2020-09-22

# WHY ARE WE DOING THIS?

- Gain experience of
  - Remote working and collaboration
  - Applying machine learning / big data to real(ish) problems
  - Technical skills
    - Programming
    - Data processing
    - Machine learning
    - Cloud Computing
  - Software Engineering
    - Version control
    - Testing/Continuous Integration
- Something to show/talk about to prospective employers
- Prepare you for your placements



# WHAT YOU'LL BE DOING

- Project running until May
  - Presented at end of year event in June
- A few hours per week of work
- Regular meetings with a "customer"
- Help sessions with RSEs

# ZOOM THINGS

- Please mute when you're not speaking
- If you're comfortable having a camera on, it's nice to see who we're talking to
- The *Participants* window has options "raise hand", "yes", "no", etc. that we will use
  - Select "Yes" now
- 3 breakout rooms
  - Each one gets an Etherpad:  
<https://etherpad.wikimedia.org/p/aimlac-2020-TEAMNUMBER>
  - We'll let you know when to go and come back



# INTRODUCTIONS

- Name
- PhD topic
- What were you doing this time last year?

# SCENARIO

- You are a small software team specialising in bespoke machine learning products
- You've been hired by an (imaginary) customer
- They wish to investigate automating business decisions with Artificial Intelligence, Machine Learning and/or Advanced Computing



# WHAT MAKES TEAMS CHALLENGING?

- What difficulties can you see with delivering software as a small team, responding to customer needs?
- Discuss in your breakout rooms
- Write thoughts down on the Etherpad
- 10 minutes

# HOW CAN TEAMS?

- Can you imagine some ways of working that might resolve these problems?
- Discuss in your teams
- Agree a series of short bullet points (max 7) describing the process you might use
- 10 minutes



TOOLS!



# slack



1



2



2



3



+

**A1 Marketing**   
 Will Rodrigues

 All threads

 Starred

 # br

 # ev

 m

Channels 

 # accounting-costs

 # **advertising-ops**

1

 # annual-planning

 # design-feedback

 # marketing

 # urgent-issues

Direct messages 

 Slackbot

 Elena Nowak

 Matt Brewer

 Toby Davies

## #annual-planning

  19  0 [Add a topic](#)

**Emily Anderson** 09:47

How are those year-end numbers coming along?

**Paul Leung** 10:18

Year-on-year growth

78 kB PDF



**Sarah Parker** 10:19

Wow, so much great progress! ⚡



Message #annual-planning

Channels are where teams share ideas and make decisions.





# GitHub



SmartTractor > Projects > Soil Data V1

Show menu

+ Add cards

Exit fullscreen



## Backlog 13

Collect satellite data and deliver to farmers

Added by Sam

Launch Plan

Added by Sam

UI for accessing app on tractor screen

Added by Vijay

Fix CSV rendering

Added by Emily

Share farms' soil moisture data

Added by Eddie

File sharing permissions

Added by fahianperaz

## In Progress 6

Crawl tractor engine data (John Deere)

#68801 opened by Melinda

Performance updates for data script

#71011 opened by Sam

User testing with farmers in China

Added by Eddie

Figure out

internationalization

Added by Eddie

New doc editor (@joh

Added by Sophie

## Ready to deploy 1

[Data] Soil data collection scripts

#71001 opened by Vijay

+ Add column

# SAMPLE PROCESS

- Meet the customer
- Hold a planning meeting
  - Agree tasks
  - Add tasks to your task list as "waiting"
  - Estimate and note task length (small/medium/large)
  - Reflect on the last iteration



- Choose a task (waiting or review)
- Self-assign the task
- If "waiting" then move task to "active"
- Complete the task
- Move the task onwards
- Meet with your technical contact
- Have a "daily standup" at agreed intervals





# STANDUP

- Three questions:
  - What have you achieved since last standup?
  - What will you work on for next standup?
  - What issues might you anticipate?

# PROCESS DEVELOPMENT

- The ideal process is different for every team and organization
- Make looking critically at your process and developing it part of your process



# EFFECTIVE SOFTWARE DEVELOPMENT

Mark Dawson, Ed Bennett, Michele Mesiti, Colin Sauzé

AIMLAC CDT Induction, 2020-09-23

# WHAT ARE THE CHALLENGES TO WRITING SOFTWARE?

- Writing software is about telling computers what to do
- Why might this be challenging for humans to do effectively?
- Discuss in your teams
- Write thoughts down on the etherpad:  
<https://etherpad.wikimedia.org/p/aimlac-2020-team1>  
<https://etherpad.wikimedia.org/p/aimlac-2020-team2>  
<https://etherpad.wikimedia.org/p/aimlac-2020-team3>
- 10 minutes



# HOW CAN WE MITIGATE THESE DIFFICULTIES?

- What techniques, methods, language features and tools would help mitigate these difficulties?
- Discuss in your teams
- Write thoughts down on the etherpad:  
<https://etherpad.wikimedia.org/p/aimlac-2020-team1>  
<https://etherpad.wikimedia.org/p/aimlac-2020-team2>  
<https://etherpad.wikimedia.org/p/aimlac-2020-team3>
- 15 minutes

# SYNTAX HIGHLIGHTING

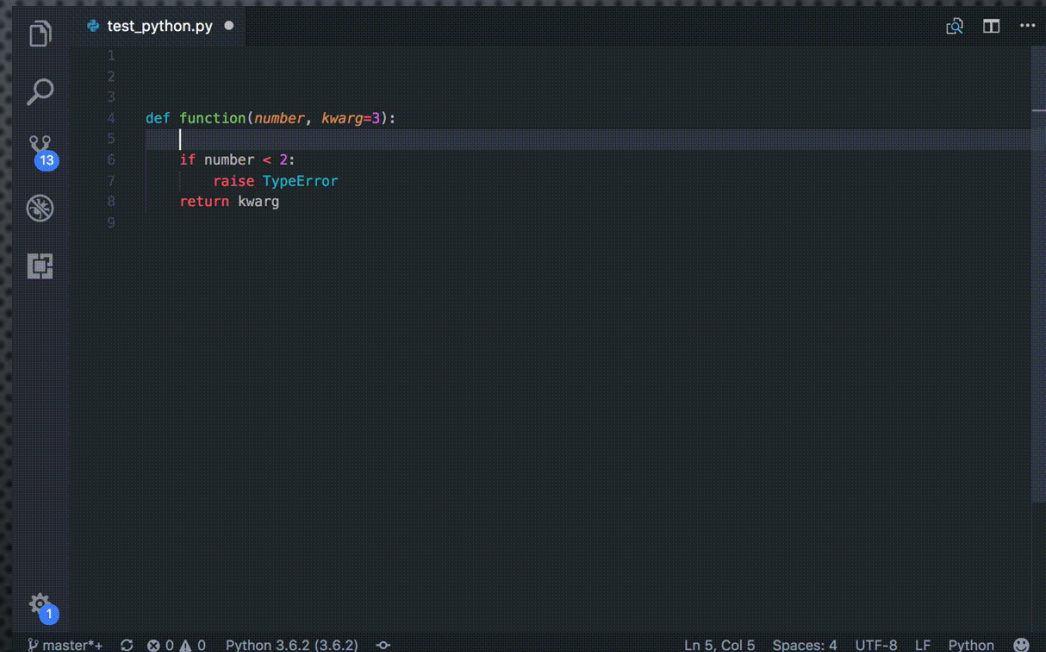
```
1 context.py 2³ utils.py •
12
13 class ContextPopException(Exception):
14     "pop() has been called more times than push()"
15     pass
16
17
18 class ContextDict(dict):
19     def __init__(self, context, *args, **kwargs):
20         super(ContextDict, self).__init__(*args, **kwargs)
21
22         context.dicts.append(self)
23         self.context = context
24
25     def __enter__(self):
26         return self
27
28     def __exit__(self, *args, **kwargs):
29         self.context.pop()
30
context.py [python][unix→utf-8] L17/267:C0 4%
```



# AUTO-COMPLETE

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 x = np.arange(-np.pi,np.pi, 0.05)
6 y = np.sin(x)
7 plt.plot(x,y)
8 plt.title('plot with numpy data')
9
10 df = pd.DataFrame(y, index=x)
11 df.plot(title='plot with pandas data')
12
13 np.
14
```

<b>abs</b>	instance
absolute	instance
absolute_import	instance
add	instance
add_docstring	function
add_newdoc	function
add_newdoc_ufunc	function
add_newdocs	module
alen	function
all	function



# INLINE DOCUMENTATION

```
def StormCastle(castleName, castleLocation, totalTroops, weapons):  
    """  
    Storm a castle anywhere in the world.  
    :param string castleName: The name of the castle.  
    :param string castleLocation: The location of the castle.  
    :param int totalTroops: The total number of troops.  
    :param list[string] weapons: A list of weapons to use.  
    :return: True | False  
    """  
    print "Storming: " + castleName + " at " + castleLocation  
    print "Total Troops: " + str(totalTroops)  
    print "Weapons: "  
    for weapon in weapons:  
        print " * " + weapon
```

StormCastle()

Documentation for StormCastle(castleName, castleLocation, totalTroops, w...

← → ↑ ↵ IronPythonTest

```
def  
StormCastle(castleName, castleLocation, totalTroops, weapons)  
Inferred type: (castleName: unicode, castleLocation: unicode,  
totalTroops: int, weapons: list[unicode]) -> None
```

Storm a castle anywhere in the world.



# VERSION CONTROL

File Edit View Repository Branch Help

Current repository: desktop

Current branch: esc-pr #3972

Fetch origin: Last fetched 2 minutes ago

Changes History

Add event handler to dropdown component

iAmWillShepherd and Markus Olsson committed c79e71c 1 changed file

Co-Authored-By: Markus Olsson <niiik@users.noreply.github.com>

app\src\ui\toolbar\dropdown.tsx

Line	Current Branch	Origin
145	145	@@ -145,6 +145,10 @@ export class ToolbarDropdown extends React.Component<
146	146	this.state = { clientRect: null }
147	147	}
148	148	+ private get isOpen() {
149	149	+ return this.props.dropdownState === 'open'
150	150	+ }
151	151	+ }
152	152	private dropdownIcon(state: DropdownState): OcticonSymbol {
153	153	// @TODO: Remake triangle octicon in a 12px version,
154	154	// right now it's scaled badly on normal dpi monitors.
155	155	@@ -249,6 +253,13 @@ export class ToolbarDropdown extends React.Component<
156	156	}
157	157	}
158	158	}
159	159	}
160	160	}
161	161	}
162	162	}
163	163	}
164	164	}
165	165	}
166	166	}
167	167	}
168	168	}
169	169	}
170	170	}
171	171	}
172	172	}
173	173	}
174	174	}
175	175	}
176	176	}
177	177	}
178	178	}
179	179	}
180	180	}
181	181	}
182	182	}
183	183	}
184	184	}
185	185	}
186	186	}
187	187	}
188	188	}
189	189	}
190	190	}
191	191	}
192	192	}
193	193	}
194	194	}
195	195	}
196	196	}
197	197	}
198	198	}
199	199	}
200	200	}
201	201	}
202	202	}
203	203	}
204	204	}
205	205	}
206	206	}
207	207	}
208	208	}
209	209	}
210	210	}
211	211	}
212	212	}
213	213	}
214	214	}
215	215	}
216	216	}
217	217	}
218	218	}
219	219	}
220	220	}
221	221	}
222	222	}
223	223	}
224	224	}
225	225	}
226	226	}
227	227	}
228	228	}
229	229	}
230	230	}
231	231	}
232	232	}
233	233	}
234	234	}
235	235	}
236	236	}
237	237	}
238	238	}
239	239	}
240	240	}
241	241	}
242	242	}
243	243	}
244	244	}
245	245	}
246	246	}
247	247	}
248	248	}
249	249	}
250	250	}
251	251	}
252	252	}
253	253	}
254	254	}
255	255	}
256	256	}
257	257	}
258	258	}
259	259	}
260	260	}
261	261	}
262	262	}
263	263	}
264	264	}
265	265	}
266	266	}
267	267	}
268	268	}
269	269	}
270	270	}
271	271	}
272	272	}
273	273	}
274	274	}
275	275	}
276	276	}
277	277	}
278	278	}
279	279	}
280	280	}
281	281	}
282	282	}
283	283	}
284	284	}
285	285	}
286	286	}
287	287	}
288	288	}
289	289	}
290	290	}
291	291	}
292	292	}
293	293	}
294	294	}
295	295	}
296	296	}
297	297	}
298	298	}
299	299	}
300	300	}
301	301	}
302	302	}
303	303	}
304	304	}
305	305	}
306	306	}
307	307	}
308	308	}
309	309	}
310	310	}
311	311	}
312	312	}
313	313	}
314	314	}
315	315	}
316	316	}
317	317	}
318	318	}
319	319	}
320	320	}
321	321	}
322	322	}
323	323	}
324	324	}
325	325	}
326	326	}
327	327	}
328	328	}
329	329	}
330	330	}
331	331	}
332	332	}
333	333	}
334	334	}
335	335	}
336	336	}
337	337	}
338	338	}
339	339	}
340	340	}
341	341	}
342	342	}
343	343	}
344	344	}
345	345	}
346	346	}
347	347	}
348	348	}
349	349	}
350	350	}
351	351	}
352	352	}
353	353	}
354	354	}
355	355	}
356	356	}
357	357	}
358	358	}
359	359	}
360	360	}
361	361	}
362	362	}
363	363	}
364	364	}
365	365	}
366	366	}
367	367	}
368	368	}
369	369	}
370	370	}
371	371	}
372	372	}
373	373	}
374	374	}
375	375	}
376	376	}
377	377	}
378	378	}
379	379	}
380	380	}
381	381	}
382	382	}
383	383	}
384	384	}
385	385	}
386	386	}
387	387	}
388	388	}
389	389	}
390	390	}
391	391	}
392	392	}
393	393	}
394	394	}
395	395	}
396	396	}
397	397	}
398	398	}
399	399	}
400	400	}
401	401	}
402	402	}
403	403	}
404	404	}
405	405	}
406	406	}
407	407	}
408	408	}
409	409	}
410	410	}
411	411	}
412	412	}
413	413	}
414	414	}
415	415	}
416	416	}
417	417	}
418	418	}
419	419	}
420	420	}
421	421	}
422	422	}
423	423	}
424	424	}
425	425	}
426	426	}
427	427	}
428	428	}
429	429	}
430	430	}
431	431	}
432	432	}
433	433	}
434	434	}
435	435	}
436	436	}
437	437	}
438	438	}
439	439	}
440	440	}
441	441	}
442	442	}
443	443	}
444	444	}
445	445	}
446	446	}
447	447	}
448	448	}
449	449	}
450	450	}
451	451	}
452	452	}
453	453	}
454	454	}
455	455	}
456	456	}
457	457	}
458	458	}
459	459	}
460	460	}
461	461	}
462	462	}
463	463	}
464	464	}
465	465	}
466	466	}
467	467	}
468	468	}
469	469	}
470	470	}
471	471	}
472	472	}
473	473	}
474	474	}
475	475	}
476	476	}
477	477	}
478	478	}
479	479	}
480	480	}
481	481	}
482	482	}
483	483	}
484	484	}
485	485	}
486	486	}
487	487	}
488	488	}
489	489	}
490	490	}
491	491	}
492	492	}
493	493	}
494	494	}
495	495	}
496	496	}
497	497	}
498	498	}
499	499	}
500	500	}
501	501	}
502	502	}
503	503	}
504	504	}
505	505	}
506	506	}
507	507	}
508	508	}
509	509	}
510	510	}
511	511	}
512	512	}
513	513	}
514	514	}
515	515	}
516	516	}
517	517	}
518	518	}
519	519	}
520	520	}
521	521	}
522	522	}
523	523	}
524	524	}
525	525	}
526	526	}
527	527	}
528	528	}
529	529	}
530	530	}
531	531	}
532	532	}
533	533	}
534	534	}
535	535	}
536	536	}
537	537	}
538	538	}
539	539	}
540	540	}
541	541	}
542	542	}
543	543	}
544	544	}
545	545	}
546	546	}
547	547	}
548	548	}
549	549	}
550	550	}
551	551	}
552	552	}
553	553	}
554	554	}
555	555	}
556	556	}
557	557	}
558	558	}
559	559	}
560	560	}
561	561	}
562	562	}
563	563	}
564	564	}
565	565	}
566	566	}
567	567	}
568	568	}
569	569	}
570	570	}
571	571	}
572	572	}
573	573	}
574	574	}
575	575	}
576	576	}
577	577	}
578	578	}
579	579	}
580	580	}
581	581	}
582	582	}
583	583	}
584	584	}
585	585	}
586	586	}
587	587	

# AUTOMATED TESTING

```
$ py.test -v --tb=no

===== test session starts =====
platform darwin -- Python 2.7.2 -- py-1.4.20 -- pytest-2.5.2 -- /Users/John/.pyt
honbrew/python/Python-2.7.2/bin/python
collected 27 items

ctest/test:0: test1 PASSED
ctest/test:12: test2 FAILED
ctest/test:16: test1 FAILED
ctest/test:0: test3 PASSED
ctest/test:0: test1 PASSED
ctest/test:0: test3 SKIPPED
ctest/test:53: test2 FAILED
ctest/test:61: test1 FAILED
ctest/test:0: test1 PASSED
ctest/test:0: test2 PASSED
ctest/test:0: test1 PASSED
ctest/test:98: test_assert_str FAILED
ctest/test:103: test_assert_equal FAILED
ctest/test:108: test_assert_not_equal FAILED
ctest/test:114: test_assert_null FAILED
ctest/test:0: test_assert_not_null_const PASSED
ctest/test:123: test_assert_not_null FAILED
ctest/test:128: test_assert_true FAILED
ctest/test:133: test_assert_false FAILED
ctest/test:0: test_skip SKIPPED
ctest/test:141: test_assert_fail FAILED
ctest/test:0: test_null_null PASSED
ctest/test:150: test_null_string FAILED
ctest/test:154: test_string_null FAILED
ctest/test:0: test_string_diff_ptrs PASSED
ctest/test:0: test_large_numbers PASSED
ctest/test:0: test_ctest_err FAILED

===== 15 failed, 10 passed, 2 skipped in 0.09 seconds =====
```

62 %  
SW Test Academy  
Reporting with pytest and Testults  
Mon 9 Apr 1:11:30AM

<Module 'results\_pytest\_example/test\_suite\_a.py'>

test3<Module 'results\_pytest\_example/test\_suite\_a.py'>✓

test2<Module 'results\_pytest\_example/test\_suite\_a.py'>✓

test1<Module 'results\_pytest\_example/test\_suite\_a.py'>✗

<Module 'results\_pytest\_example/test\_suite\_b.py'>

test6test\_input = 6\*9, expected = 42<Module 'results\_pytest\_example/test\_suite\_b.py'>✗

test6test\_input = 2+4, expected = 6<Module 'results\_pytest\_example/test\_suite\_b.py'>✓

test6test\_input = 3+5, expected = 8<Module 'results\_pytest\_example/test\_suite\_b.py'>✓

test5<Module 'results\_pytest\_example/test\_suite\_b.py'>✓

test4<Module 'results\_pytest\_example/test\_suite\_b.py'>✗

VIEW: RESULTS

PROJECT  
SW Test Academy

TARGET  
Reporting with pytest and Testults

DATE/TIME  
Mon 9 Apr 1:11:30AM

SORT BY  
Suite

AUTO REFRESH

Direct links for current selection:

PROJECT  
<https://www.testults.com/results/rsp/View/results/project/b290840a-68a0-4c50-b962-0997cd32066a>

TARGET  
<https://www.testults.com/results/rsp/View/results/target/b290840a-68a0-4c50-b962-0997cd32066a>



# CONTINUOUS INTEGRATION

The screenshot displays the Travis CI web interface. At the top, the header includes the Travis CI logo, navigation links (About Us, Blog, Status, Help), and the user profile 'Sam Iamm'. A search bar for repositories is located on the left. The main content area is for the repository 'green-eggs / ham', showing a 'build passing' status. The 'Current' tab is active, displaying build details for the 'master' branch, including commit 'abc123' by 'Sven Fuchs'. A 'Restart build' button is visible. On the left sidebar, a list of recent builds is shown for various repositories. At the bottom, a 'Job log' section displays the build output, which includes worker information and system details.

Travis CI - Test and Deploy with Confidence

Travis CI About Us Blog Status Help Sam Iamm

Search all repositories

My Repositories +

- ✓ one-fish/two-fish # 2686  
Duration: 33 min 46 sec  
Finished: 30 minutes ago
- ✓ hop-on/pop # 7001  
Duration: 22 min 54 sec  
Finished: about an hour ago
- ✓ horton-hears/awho # 209  
Duration: 53 sec  
Finished: about 2 hours ago
- ✓ green-eggs/ham # 209  
Duration: 53 sec  
Finished: about 2 hours ago
- ✓ ohthe/places-youllgo # 778

green-eggs / ham build passing

Current Branches Build History Pull Requests More options

✓ master adding in Oh the places you'll go! #209 passed Restart build

You'll be on your way up!  
You'll be seeing great sights!

⌚ Ran for 53 sec  
📅 about 2 hours ago

🔗 Commit abc123  
🔗 Branch master

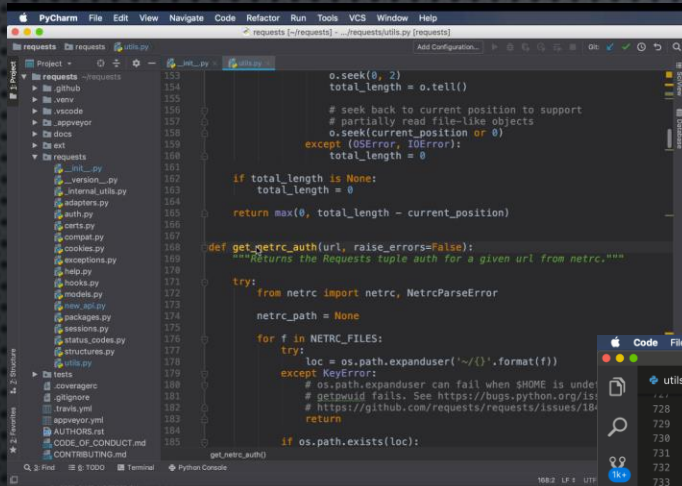
👤 Sven Fuchs authored and committed

Job log View config

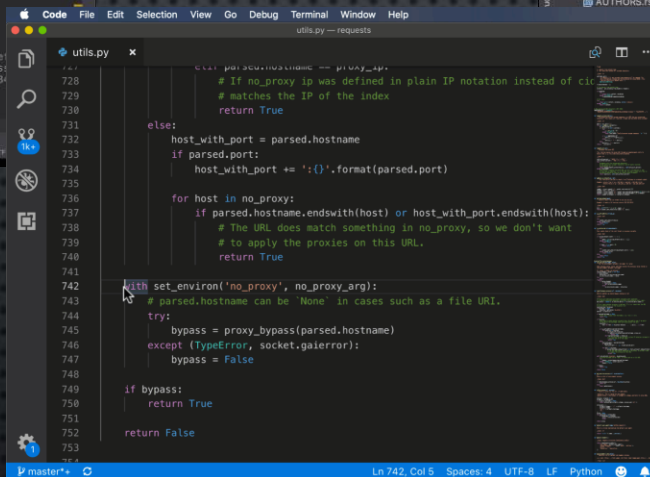
Remove log Raw log

```
1 Worker information
6 mode of '/usr/local/clang-5.0.0/bin' changed from 0777 (rwxrwxrwx) to 0775 (rwxrwxr-x)
7 Build system information
8 Build language: node_js
9 Build group: stable
10 Build dist: trusty
11 Build id: 345296935
12 Job id: 345296935
```

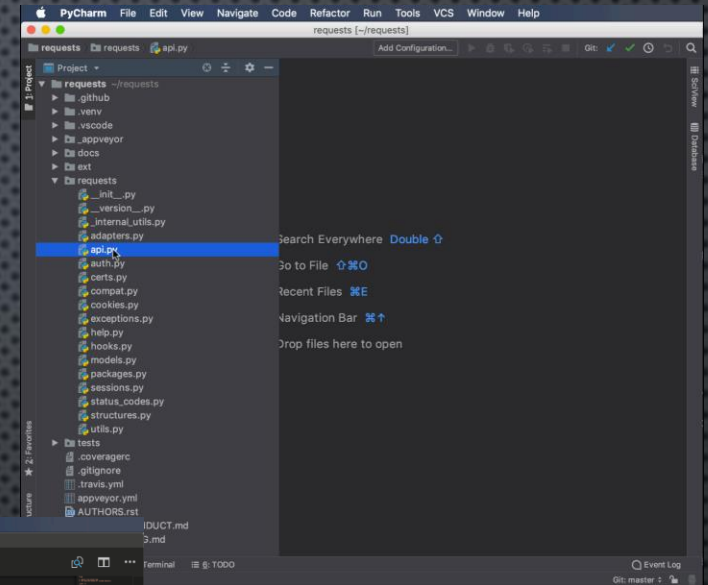
# AUTO-REFACTORING



Change signature



Extract function



Rename file/module



# SYNTAX CHECKERS

```
#!/usr/bin/env python
# encoding: utf-8
import datetime
from south.db import db
from south.v2 import DataMigration
from django.db import models

class Migration(DataMigration):

    def forwards(self, orm):
        "Write your forwards methods here."
        pass
        pass

    def backwards(self, orm):
        "Write your backwards methods here."

    models = {
        'foobar.stuffstuff': {
            'foo': {'object_name': 'bananas!'},
            'foobaz': ('django.db.models.fields.CharField', [], {'max_length': 100}),
            'foobaz': ('django.db.models.fields.AutoField', [], {'primary_key': True}),
            'foobaz': ('django.db.models.fields.CharField', [], {'max_length': 200}),
            'foobaz': ('django.db.models.fields.BooleanField', [], {'default': False}),
        },
    }
```

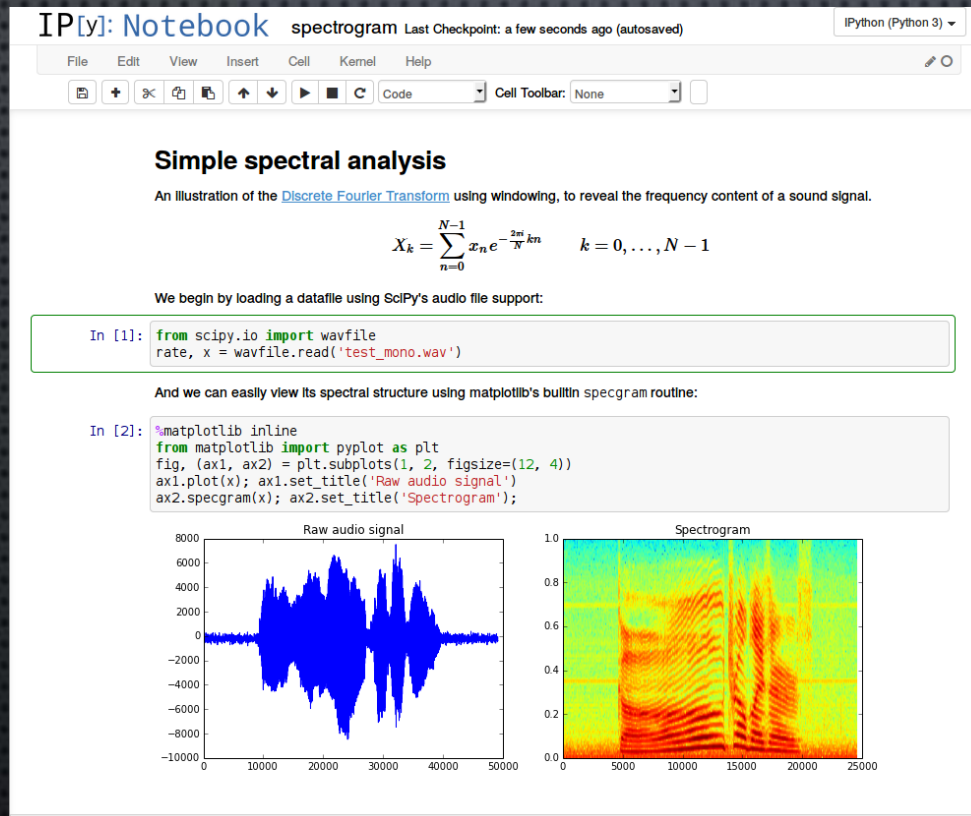
# AUTO-FORMATTERS

example.py

```
1 x = { 'a':37,'b':42,  
2  
3 'c':927}  
4  
5 y = 'hello 'world'  
6 z = 'hello '+'world'  
7 a = 'hello {}'.format('world')  
8 class foo ( object ):  
9     def f (self ):  
10         return 37*--+2  
11     def g(self, x,y=42):  
12         return y  
13 def f ( a ) :  
14     return 37+--+a[42-x : y**3]  
15
```



# LITERATE PROGRAMMING



# WHAT DOES GOOD CODE LOOK LIKE?

- Each example has many positives and negatives
- Discuss in your teams. What are the good points? What are the bad ones?
- Why?
- 15 minutes



# CODE REVIEW

- Why?
  - Learn
  - Share
  - Make code better

# CODE REVIEW EXERCISE

- What?
  - Is it clear what the code is doing?
  - Would it be easy to reuse parts of the program? Could this fail?
  - Is it easy to test?
  - Are there any bugs?
  - Are function/variable names descriptive?
- How?
  - Be positive and constructive
  - Point out things you like
  - Clarify reasoning where code doesn't match your expectations
  - Discuss your opinions
  - Is it worth changing? Sometimes it's OK to comment but take no action