Djangocon 2015

Cardiff City Hall

Date: 1st June 2015

DANIELE PROCIDA: Good morning everybody, welcome to Cardiff and welcome to Djangocon Europe 2015 it is (APPLAUSE)...

It is honestly a really superb pleasure and honour to welcome you here so I am so glad to see so many friendly familiar faces and new 1s too.

While you are here, if you need any help, speak to any of the conference volunteers we are all wearing blue lanyards. Could the volunteers just stand up for a moment, so you can get an idea of where, okay, there are several of us, as well as some of the volunteers our, the organising committee and I will introduce them so again you can speak to us if you need to.

We have Ola Sitarska and Baptiste, stand up they are our code of conduct liaisons, any issues speak directly to them and their telephone numbers are in the hand-outs you were given on arrival and in the programme booklet in your bag and on the website. Baptiste has done a super job of wrangling files through Python scripts to manage information for this.

Our Cardiff team, David Neil. Most of them downstairs, there is David, Geraint Palmer, Jason Young, Vincent our treasurer and that is me. Even she can't be here, Stephanie is part of our committee, we are all volunteers and committee members all unpaid volunteers, in fact it is worse than that, at Djangocon everybody pays. We operate the traditional Python and Djangocon, speakers organisers we are all in the same boat, we don't get free passes to the event, it helps keep prices low, the only people who get free tickets, are those who apply for financial assistance because they wouldn't be able to afford to attend otherwise.

So, speaking of lanyards we have a photography policy, if you are wearing a black lanyard that means that you are happy to be photographed. If you are wearing a, or if someone is wearing a white lanyard, no photographs, they don't want to be asked why or if they are sure., if you find somebody wearing a white lanyard in the photograph, remove it from your collection, thank you for respecting this policy. If someone is wearing a blue lanyard they are a volunteer and happy to be photographed.

Our event is governed by a code of conduct, it is listed in full in the programme booklet but, the short idea is no-one at the event is to suffer harassment or abuse of any form and no-one should behave in way that distresses anybody else, if there is any kind of problem, speak to a volunteer, you have the code of conduct liaisons.

All the information that you need is on our website, includes the timetable also in your programme booklet. Also in your bag as well as the programme, we are trying to operate a waste free Djangocon, so conferences are notoriously wasteful affairs that is why in your bag you will find a lovely non-disposable indefinitely reusable attractive stainless steel bottle which some of the sponsors have paid for, when you see them, say thank you to them. We won't be providing disposable bottles or cups, that is your drinking vessel, please use the water bottles, water fountains -- your bottle has been washed and rinsed by hand. Now we didn't realise until last week when they were delivered that they were going the need to be washed before use. So, in case you are curious this is what 400 washed bottles look like! (APPLAUSE).

And, this is what a Djangocon Chairman looks like when he is washing 400 bottles.

I think it proves that when you feel glamorous you look glamorous; I hear that the film industry is looking for its next Carry Grant, if you want to talk to me in the break; I am available for a chat.

So lunches and dinners. Our policy on waste is also why we have been very strict about asking you for information about which meals you are coming to. So I don't apologise that we have bombarded you with annoying requests for information about this. We don't want to throw away food at the end of every meal. We don't want to not just because we don't want to pay for it and flow it away, but because it is wrong to throw away food. We have asked you to tell us which meals you are coming for. So you must tell us, if you haven't already done so on your conference ticket told us which meals you are coming to over the next 5 days, find your conference ticket, every conference ticket that you have bought or has been allocated to you, should have been e-mailed to you, probably several times, certainly when you get a message through titto it e-mails all your tickets to you. Find the ticket, check the information you have provided about meals is accurate.

We would appreciate that. It applies also to tonight's conference dinner. There is no extra charge for that dinner but we only have space for the people who told us they were coming. If you were hoping to come to that, but haven't told us, there won't be a place for you and you will have to go on the waiting list, that is how strict we are. We are not apologising for this. We will try and get everyone in, I think we will get everyone in, but now is the time to check your ticket so you don't miss out on a meal at the museum next door. If you are not coming to tonight's dinner or any other dinner you said you were, please amend your ticket accordingly so we can allocate the space to someone else.

Come and see us at the registration desk if there is a problem about this.

One of the other things in your bag is a piece of art work. It is a handmade print from a limited edition, been numbered and signed by the artist. We commissioned these; they are all interpretations of the animal wall at Cardiff castle. Made by the Cardiff print workshop it is one of the loveliest landmarks in the city. As I was coming in this morning, I saw at least a few Djangocon Visitors having a look at the wall. Look at that, you have got a lovely souvenir, the artist wills be here on Wednesday, they will have more of the prints if you want to complete the set.

If for some reason you are not interested in having a print just give it back to us or give it to someone else.

We have a crèche; this is the first Djangocon with a crèche for small children. So we have got a couple of customers for the crèche and the crèche is in Room L so, maybe this will become a standard theme at future Djangocons.

We have a quiet room which is in the Council Chamber in that direction; it is a nice wood panelled room. If you want a bit of quiet space and time to yourself, go in there, if you are in there please respect the quiet.

We have 2 speech to text transcriptors working, there is Sheryll and Hilary they are doing a fantastic job.

(APPLAUSE).

We have 4 screens with their output so, if you would like to sit near a screen because it will help you follow along, just ask someone else to move along or position yourself near it. Those screens are to help people of course with hearing difficulties and also people whose first language isn't English, so, they are doing a very difficult job because not only is 50% of what they are hearing, stuff they have never heard before, some of it is delivered at a much faster speed they are used to working with -- Russell! -- (LAUGHTER) and, some of it is delivered in a wide variety of foreign accents which all adds to the challenge and, as I said to Hilary or Sheryll yesterday, as she was wiping her brow, you have heard nothing yet! You have got 3 more days of this. So, thank you very much. It is fantastic to have this here as well.

I have mentioned our volunteers, all of this, our volunteers are making it happen but the sponsors of the event are making it possible and affordable were it not for the sponsors the tickets for the event would be 2 or possibly 3 times what you are paying for them. They are not just commercial sponsors, they are part of the community, participating, many of them were running workshops yesterday at the open day, they contribute a very great deal, so please take a few moments to thank them with a hopefully, hearing person but also on-line. If you value Djangocon please tell the sponsors you appreciate their supports, even if you are not using the products or whatever they are doing, they are still making the whole thing possible. We would not be here if it were not for them.

So, please thank our sponsors when you speak to them and thank them now. (APPLAUSE).

There is our twitter user name and our hash tag common thing. The IRC channel is Djangocon on free note.net.

Last couple of things before I go, I found a pair of glasses, these were left on the registration desk yesterday at the open day, if someone has lost a pair of glasses speak to a volunteer.

So enjoy your Djangocon, have a lovely time here and in Cardiff, I am going to introduce our first proper speaker of the day. Baptiste Mispelon, 1 of the Django team core developers, it is lovely to have him here, he is at every event helping. He is, it was quite difficult to get him to do this but he is our opening keynote speak r and I think it is ability time. He is going the be talking about his adventures in Djangoland. Please welcome Baptiste.

BAPTISTE MISPELON: (APPLAUSE).

All right. Yep. Hello everyone. My name is Baptiste; you might know me from my on-line nickname. It is not easy to pronounce, as Daniel said I have been a user at Django for about 5 years now, a member of the team for 2 years but my proudest achievement is for the past 6 months the member of the Django maintenance team, more recently 1 of the founder of the cup cakes and emoji, I want to talk to you about the year 2014 and what it meant to me, I spent the year roaming around Djangoland, what is it? Before I explain, I have to tell you a bit about Esperanto. It was a language invented by Mr ... a Polish guy at the end of the 19th century, I encourage you the read the article about it. I want to talk about one specific aspect of it. In the culture, there is the concept of Esperantujo translates as Esperantoland. It is not attached to a single piece of land, it is an international community, so this idea, this concept that the moment two people are speaking to each other in a Esperanto, the answer, it is a virtual country, you can summon by communicating with another human being in a shared language.

I think it is a beautiful concept, kind of sounds familiar though, there is this other international community, you may have heard of them, also ends with an O. Kind of has a virtual place, the mailing list, the channels, these literally exist only because people decide to talk and share about their favourite language or at least a framework. So this is what Djangoland is, it is just a place that comes alive when 2 or more people talk about Django.

So this is what I want to talk about today. This is going the be the story of my year of 2014 on which I travelled around the Djangoland, the people I met, the lessons I learned, before I get to 2014 I want to give you context of how I got there. For this, I will try a little experiment, please bear with me, sit back, relax and enjoy the story of my emoji life. All right I was in born France almost 30 years ago, growing up in interested in science, maths, physics, studied to get to engineering school. Never had formal training for computer science but was interested in computers.

When I was 20, I made what was probably the single biggest decision of my life. I was in love, but the person I loved was in a far off way, in a different country. I was in school, stuck on a path where I didn't feel I belonged to.

So in February 2006 after months of feeling unhappy, I made the decision to drop out of school, I was 20, I went to Hungary I flew the thousand kilometres away and moved in with my girlfriend. No job, no adult supervision for the first time in my life. So for a few months I was doing nothing, was at home, watching TV playing video games.

It was fun but you know, one day, out of the blue and old high school friend contacted me. He had this idea for a business and he needed someone to make him a website. I didn't know how to make websites but he said, if you remember in high school I used to make little text based silly games in basic, that is kind of like a website right? Not really (LAUGHTER), but I can probably figure it out, how hard can it be?

So for the next few years I was learning PHP on my own, building the website as I wept, it wasn't pretty. I made it up as it went along. Soon we started to run the company around the website and growing. At some point I discovered this thing called Django, Python web framework, I fell in love with it. So we slowly switched our hack together to hack together Django, turns out works much better, that is how I stood in 2012 the website coming along nicely. This time it was a very small decision but I think it ended up changing my life.

You see in March I bought a ticket to Djangocon. It was the first time I had ever attended a conference for meet up or anything I had been watching videos of conferences on-line thanks to things like py videos but I didn't know anybody and I definitely had never been to any conference but in June I flew to Zurich in Switzerland and had an awesome time, it was a wonderful experience and I met like so many great people but one of the key decisions I made I decided to stay for these things called sprints and I think that had the biggest impact of my professional and even personal life. I wasn't exactly sure what to expect but during the sprints I met Audrey Greenfelt, you might know her as the author of Django series books. And she didn't know me I didn't really know her but she showed me how you can install this thing called swings then build Djangos computation directly on your computer and that blew my find. I met Eric, a founder member, I met mark who joined the team later, I met a couple from Poland who ended up organising Djangocon the year after and I met these great people and they welcome me. Of course this was sprint so I try my hands at fixing bugs in Django but I didn't really manage to do anything but it didn't really matter. What mattered was I got an understanding of the process of what it's like to contribute to Django. And because of that a few months later I was able to make my first ever Django contribution in December. It typed a one line character in error message I was super proud of it. After that though I didn't really continue I was using Django still at work of course and on personal project but I wasn't just contributing any more until a few month later during my work I came across a bug in a 4 loop template implementation so I made a fix and it got launched quickly and I dug a bit deeper and I saw actually this bug was a in a few other template tags so I investigated got to the bottom and made a patch fixing all of these and it got merged as well so I was gaining familiar with the code base as well as the process of contributing and it just got more and more interesting.

So, I started looking to the bug checker, I was fixing bugs other people had reported, I was not doing anything major just fixing things I knew I was able to handle.

Then came Djangocon again which of course I attended, you may have heard it was 2013 edition also called Django circus because it took place on a horse race track in Poland. To this day it's still one of the best conference I ever attended I met Audrey and Danny again. This time they were selling the brand new first edition of their books on Django. I saw Brendan one of my first speakers give a rock concert in a circus tent and get a bunch of programmers to dance - unforgettable memories really!

Of course I stayed for the sprints and saw Erik again and to my surprise the team says you have not been giving us patches regularly and we really like that he said if you are interested, you keep doing this, you know, maybe you can join the team later. And I did just that a month later. In June I got my commitment and I made my first push to Django's poster. I must have checked 3, 4, probably more times that everything was OK because I was so scared of breaking everything before the final push but I did it, I pushed a one character typo fix in a documentation and I couldn't be more proud! {Laughter}.

Then things started accelerating a bit. In July of 2013 my girlfriend got a job offer to go teach for 6 months in a school in Norway as part of a European programme. That happened really fast and a month later in August we gave back the keys to our Budapest flat packed our lives in a small car and drove 2000 kilometres from Budapest to Burr(?) - you may have heard of it but maybe not, it is 5,000 people city 2 hours from Oslo in the middle of the mountains not the busiest place on earth; I was working remotely from my cabin in the middle of gorgeous nature. Everything happened fast as I said so my team colleagues and I weren't really prepared for doing remote work which is more involved tan just saying oh we're just going to be talking on Skype every day but we pushed through. It was only going to be for a few months until December any way so we could probably figure it out.

Then this was Norway so winter settled in, temperatures started going down, days got shorter, the town I lived in as I said was pretty small. I didn't know anybody there except for a few of my girlfriends colleagues so as a very social programmer I just kept to myself I was working in my cabin all day every day not going out month much.

Wasn't ideal but this was going to be only a few months so what did I do? I just pushed do.

That brings me to 2014. Turns out the 6 month contract got extended by another 6 month which was excellent news for my girlfriend because things were going really, really well at her school. For me though well not so much. You see, my slip schedule started getting really messed up, blame it on Norwegian winter months or something else. I would sleep during the day. Try and work during the night. Didn't really work. I couldn't concentrate on anything as a programmer sometimes you need to hold complex problems in your head, figure ins and outs and I couldn't do that couldn't concentrate. Communication with my team was getting worse and worse. I would disappear from IM and emails days at a time and feel guilty, I disappeared and disappear a bit more - you see where this goes I would spend days without going out. I would not see anybody except my girlfriend when she came back in the evening from work.

Some of you may know this. This is pretty common in our industry. What we call this burn out I didn't really recognise it at the time. Oh I'm just tired it's the winter but no this was burn out I had tried to push through it and let me tell you it doesn't work. Actually makes it worse.

And so I thought what can I do about this? And my first instinct was I felt so isolated, so lonely that I just decided you know what I am just going to start travelling. So I spent all my money on buying conference tickets and flights and going round conferences, meet ups, sprints, workshops what have you, I was to meet the people of Djangoland and as many as I could.

So here it is month by month the travelogue of my year 2014. February. February had 4 weekends and I spent each of them in a different country. The first one I was in Norway in my little wooden cabin in the mountains. For a second one, I went to Cardiff, car ... diff! {Laughter} {applause} I attended the first edition of Django weekend. This was some of you may know you may have been there 2 day conference here at Cardiff at the university hosted by the {inaudible} I saw many great talks on the weekend but one of them specially sticks in mind. Something I had never thought of but is obvious in retrospect. This was a talk by Dr. Hayley Gomez about all the Python libraries that astronomers like her use in their day to day jobs. I had never realised that there were all these people, they use Python, they use Django every day, they do really cool stuff and I had never heard of them, it made me realise how we tend to build our communities in bubbles an not see the outside of it and we miss out on all these really cool things that people do.

That's also where I discovered existence of Welsh cakes. If you haven't had the chance yet, I heard they will be served at basically every coffee break I encourage you to try it.

Then I went home in Norway and then again third weekend of February I flew to Krakow in Poland. I attended a 2 day sprint organised by carconif(?) It was a second sprint as my member of Django team. I have many great memories of their sprints. I remember this guy, he hacked together a script in Python that takes the whole of Django source code like everything, push it through a spell checker, uses some regular expression to filter out false positives and stuff and he found typos everywhere and I mean everywhere; I'm talking documentation obviously, doc strings, and comments and even viable names that had typos so we fixed those but we realise we can't just use this horrible script all the time so he looked into building a speck checker as part of the documentation building process and to day actually this spell checker is run automatically whenever somebody submits a request to Django and that's work from February. It's also the sprints where I realise the importance of windows support for Django. You see, I'm long time Linux user, I don't use users but at the sprints there were these 2 people, they had never contributed to open source but they were really motivated but the thing our contributors guide wasn't really made with windows in mind, it was using commands that don't work on windows, it was using pass with a forward slash instead of a backward slash so it doesn't work on windows, some tests were plain failing on windows and I realised how much of a terrible experience it is for somebody who is new coming and motivated and made me think how many people did we lose who could have been really productive contributors just because we don't care apart their platform?

It really made me think.

Also the would struck was amazing and started my last story with Poland's greatest invention the pierogi and then I went home and the last weekend of February I flew to Amsterdam. The Dutch Django association was organising its annual sprint and I joined in the fun along with {inaudible}. I was hosted with a member of the Dutch community Nicholas who had a guest room for 2 people so I shared a room with Daniel. It turned out this was going to be the beginning of a long tradition as you see in the following month's also I discovered the existence of stroopwafels a delicious Dutch biscuit with some syrup and worse than that it turned out Amsterdam airport sells those by huge boxes which is a great temptation every time I fly through!

Then came March. In March I flew to Warsaw Poland. I attended a conference called mako(?)land first edition it was a conference about 3 D printing, hardware, just like making cool stuff real cool stuff not like software {laughter} this is only my word like I said I am software developer I don't really know anything about hardware, I'm not really good with my hands but this conference was organised by some of the people behind the Django conference so I thought what could go wrong? And nothing went wrong it was a great experience; I got to meet all these people that again had I not stepped outside of my comfort zone of my Django bubble I would never have met.

After makoland I was invited to speak at the local Python user group. I spoke about the Django contribution process and I got to meet Shamac(?) the organiser and later became good friends. Later {inaudible} his company's office so I didn't have to work from my hotel room which was tiny and I got to participate in the company football weekly tournament where I did very well I let you know I actually beat them and since none are here today I can say whatever I want and.

Then came April, April was pretty intense as you will see. Started off easy. I went back to Budapest to see my colleagues, enjoy the food, thermal bath's I attended a conference called redidocs(?) You may have heard of it where you can host your {inaudible} for free and it's built on Django and it was hosted by Eric {inaudible} so Eric and Troy and Ruth organised redidocs and it is a conference for people who write documentation; you see I write a bit of documentation when I have to as part of my work on Django and the conference was literally minutes from my office so there was no reason for me not to attend. It ended up again me getting out of my bubble and meeting great people, people I don't usually meet, word press programmers, droopel programmers technical writers, translators, all this cross-section of people that you tend not to meet at our conference. Our met friends from the Django team, Yannis from Berlin and Zedan from Israel who I met for the first time and I think the important bit was making friendships, exchanging ideas outside of the Django bubble.

From Budapest I flew to Montreal. It was the first time I ever travelled out sad of Europe and I was pretty excited about it I was there to attend the big py con, in the US, in Canada for some reason, ended up a huge experience for me a big family reunion, before I made a list of all the people I wanted to list, stars, people I admire, work with every day, but on-line, never met in person. Huge sensory overload, left me completely exhausted physically from all the germs that tend to float around with international conferences with AC blasting but mentally seeing people all the time in a short amount of time it was really draining. But out of everything that happened at this conference all the great people the great thing that happened there is one thing I want to share that marked me deeply and I think I never forget. You see of course I stay for the sprints at py con as I do but on the second day of the sprints I got a message from my father telling me my grandmother had just passed away unexpectedly. I was not sure what to do. I stepped out of the room found a place to sit then it dawned on me. I was completely alone I was thousands of kilometres away from anyone in my family. I wasn't able to do anything I was lost. I started to break down. Then, I saw Diana Clerk the conference chair who was pretty busy organising the conference but I stopped her and I asked her if there was may be a room where I could get some privacy and sort myself out? Without a question not a word she said yes. She started taking me to a room where I could get some private moment. On our way there, I started explaining to her the situation and the moment that I let out the words that my grandmother had passed a way I started crying, breaking down. She literally offered me her shoulder to cry on, she calmed me down, brought me to that room, then I was able to get on a phone call with my family, my siblings, and I regained some composure. Diana checked up on me making sure I was OK and every day after that at the conference she would see me asking if I was OK? I was OK. After this phone call with my family I felt better, I managed to move my flight one day early and was able to attend my grandmother's funeral and be with my family like I wanted to.

You see I thought I was alone I thought I had no friends and the family but I was wrong. I was among my Python family, I was among friends. This episode understandably I think has marked me deeply and I think this kind of behaviour for me has set a bar when it comes to empathy and this is what I strive for in my life.

So that was pretty intense so after such an intense month it was time for relaxing vacation. Luckily, it was a time for Django Island the 2014 edition of Djangocon Europe that was set on an island in the south of France. There, I met as usual many great people. I met Shy for the first time. He is a member of the Django team from Israel. I remember we talked about the algorithm that allows switching between right to left and let to right languages inside text and how much messages like everything with computers I met clawed a member of the Django team from Switzerland and he works a lot of the translation machinery for Django and he translates all of Django's documentation into French which beats me but most impressively I saw Jacob do a release of Django over the 3 G from someone else's phone because the conference Wi-Fi was fine, can release Django from anywhere, next step is a plane.

Of course I stayed for the sprints and I did try something new at the sprint, I tried paper(?) gramming, triaging, I met a French contributor and together we went to bug checker and looking at tickets seeing old tickets if they were valid and it was the first experience for me and it ended up been very, very productive. It turns out 2 people working together go much faster twice the speed where you go on your own so if you stay for the sprints and I encourage you to may be give it a try.

I went home and end of May I went to Serbia to visit some family. I was in the country so made a visit to Belgrade to visit Peta. I met him first time completely by accident and I was getting lunch and was looking for a place to sit I saw him sitting on his own and I decided to have lunch with him. We talked and he told me he had brought his copy of Audrey and Danny's script of Django except his copy was the one he made himself by printing out the PDF so it was a huge stack of paper but he still got Danny to sign and Danny I think had a good laugh about it, ended up meeting him again and again at several conferences and he always had the most outlandish story so sometimes all you need to make a friend is strike a conversation over lunch with a stranger so may be try it.

So yes June started the month where I travelled the most, 3 weeks I was away from home. I started in Serbia, then took a bus to Budapest. From there a train back to Krakow. I met with my friend, gave a talk, then about my favourite model of the Python standard library, might be weird, sit weird to have a favourite Python module. If you are stuck, try it, you might learn a cool 1.

Then, took a train to Warsaw, where I again was invited to speak at the Python meet up. There I spoke about little twist of my previous talk about my favourite bits and bobs of the Python library, it was the second time, I was really impress it had way they ran it. It is completely volunteer run and they have been doing it for 4 years, every month and they haven't missed a single month to me that was impressive, in fact they are so successful that they recently held the Python summit in Warsaw, which I heard went great.

From Warsaw I flew to Rome, a beautiful city on the top of a cliff. There I was attended the Django village it was really amazing, I mean the venue was this old medieval building with really beautiful architecture, the food was this being Italy, even the worst food was delicious most of all I think I learned that every single conference should have a quiet room where they play classical music it was a wonderful experience. This time I shared a room. We had many great discussions at the conference, I think 1 of them was more important than the others, back on the way back to Rome and Mark and I were talking about our experience contributing to Django, sometimes a contributor submits a patch for requests and we have to review it to make sure it is acceptable but they are trying to change a part of the code we don't know how it works. Very few people know how it works, very little time to handle all the requests. Could we maybe help spread the knowledge inside the team and the community. Of course our solution was, let's organise a conference. But if we could get a conference whose sole purpose was to get the Django team in the room and spread the knowledge internally, so help drive features forward and help make contributions easily. Mark took out his laptop. Some spread sheets, flight costs, hotel costs we realised we have never organised a conference before between the 2 of us, it might be doable. That is how we started the Django conference, happened a few months later as you will see from Rome I flew to Prague. I met Honza, he let me stay at his apartment. In 2013 I attended a meet up in Berlin, every time there is a Django meet up I get an e-mail asking if I am going the be here. Usually a few hundred kilometres away, this time, I took a look at it. Trains from Prague to Berlin, doable for me to do a round trip for the night. So I did that, took a train to Berlin.

I attended the Django meet up. Met with the German community, summer meet up. Informal, met in a beer garden, talk about Python and stuff. Again, wonderful discussions I remember 1 specifically, was talking with Janis and ... they were trying to bring a code of conduct, they were struggling, a few months ago, I saw that there this is this thing, the Berlin code of conduct, ... decided to pledge together and pledge the 1 code of conducts, turned out the Django meet up was part of the list, interesting to see how people working, can make progress and go in the right direction.

From Berlin to the train back to Prague, I attended my first Prague Python meet up. Interesting in Prague the way they do things at the beginning of every meet up. Every single person in the room just stands up, tells their name and tells what they have been doing with Python recently, I thought was a great twist for a meet up. July came, in July, sadly my girlfriend's contract ended and we were both sad about leaving Norway, packed our lives in a small car and drove back to Hungary. We drove 1958-kilometers through Denmark, Germany, ... finally Hungary, another time we stopped at the border, this is the privilege of living in Django space. In Budapest I was staying at a Budapest I was staying at a friends, looking for an apartment. Soon time to go back to Berlin, took a train there and attended Europe Python, I had not had the occasion to attend so was pretty excited about it. As good as the conference was, for me the best part of it was what happened on the first day, you see a few months back I was in Krakow. She told me about an idea, wanted to organise a Django workshop with a lot of hard work, so improvisation, a bit of luck and managed to pull them off., Django girls, a free workshop for women, learn to build a website with Django, I signed up as a coach, a fun adventure not knowing what to expect. In Berlin, assigned 3 women on my team. A coaching guide for what to do, a tutorial to follow, but that was it. The workshop was big improvisation but a huge success and after you could see the attendees of the workshop attending the conference they were in the hall ways attending talks speaking with sponsors just making friends and they were complete and integral part of our community.

But even more inspiring were the sprints so as you have seen I have been to a few sprints but this time the attendees of the workshops, stayed for the sprint, got on the tutorial they have used a few days prior. I have never seen so much motivation and energy at a sprint, the most awe inspiring thing I have seen.

Also at euro Python, the Django team, we had a crap html website we were going to work on the CMS. She took pity on us, Mark and me, in a couple of hours made a gorgeous website we ended up using, just in time for announcing the conference at the lightening talks, maybe a couple of dozen people to sign up. In a day or 2, hundreds of people signing up for the conference, maybe we are on to something.

Then went back in August, to Budapest, I was spending the month, looking for a flat half-heartedly, staying at my friend's guest room slash gym. At the weekend travel to see my girlfriend at her families. I was bummed out., in Norway trying to figure out how to settle back to a normal life and normal working environment.

September came, found a flat, that didn't stop me from travelling, I took a bus to Croatia, I met Josie, was then a 13 year old, she attended the workshop, she liked it so much, later at the conference she did a lightening talk in front of a thousand people, a plan aimed at, workshops aimed at teenage girls and teach them programming and teach them how to do cool stuff. I was in Zagreb, I was in the initial pixy dust workshop., take the Django, tweak it a little bit and have it for half the workshop. The other half of the day, they were been (INAUDIBLE) we had to cut down. I meant Anna, together we prepared the material and the workshop came. I was really, really stressed the night before I didn't sleep much. Was completely out of my depth. It was a different universe, if you think you know something, try teaching to a room of teenage girls, it was a huge humbling experience for me. I went to Budapest, I flew to Dublin in Ireland and attended the Irish Python conference was expecting a small local conference and I was pretty surprised when more I think 300 people, close to it showed up. Met the organisers and 1 of them who told me the trick he had during the conference, he would not attend talks, just go around, see someone sitting by themselves, sit down and strike a conversation with them, where they are from and what their favourite modules. There is always something to learn and interesting story. This resonated with me, only a couple of years ago I was attending a conference by myself, this can make a difference, try it, walk around the conference and if you see someone by themselves strike a conversation.

Also, I shared a room with Daniele. November came, in November, to Warsaw again, meet with 2 Django team members, staying together for 2 weeks, staying at ... if you have contributed to Django in the past, a year or maybe more, probably interacted with Tim, Django fellow, reviews all the requests and he is helpful. I talked to him on-line before but never met him in person, from Philadelphia, or from the US, I didn't know him very well. After meeting him for the past 2 weeks spending time with him, something changed in the way we communicated. I think after you meet someone in real life the way you talk with them on IRC and e-mail changes. When I would talk with him, I could picture him in my head, the way he talked, his hand gestures, I knew him better so it made communication better.

I knew never to again trust him with a map, he could not read them and always get lost (LAUGHTER), sorry Tim. So we were coaching at the Django Girls Workshop in Warsaw. This is the first time as a Meta-coach, I didn't have a group. I was going around the room and helping and debugging, we were fixing the tutorial live. So Warsaw, we took a train back to Krakow, time for another workshop, on my coach application. There was dietary requirements field. I hadn't been filling out those a lot. I wrote down that I only eat pierogi. The organiser, thank you dinner for the coaches, I was brought a plate of pierogi. I am not sure still to this day who had the last laugh because I like pierogi's, so.

So then we flew to Amsterdam, the time for Django ... despite the weather that some people complained about, Amsterdam in November, I don't know what you were expecting. The conference went well, the unanimous feedback was just make it bigger, we want more of this. There is something to be done here, also the time by accident we locked everybody in the room for 10 minutes, so nobody realised. It was the biggest gathering of the Django core team ever and it was a pleasure to meet them like Arnsi from Finland been on the team for a while, nobody met him in person, we didn't know what he looked like, his picture is a robot but he turned up to everybody's surprise and gave a great talk, I think he is here at the conference giving a talk.

Django Amsterdam, Dodi, attendee also from Budapest, just got hired as a junior Python programmer and I was graduating from attending to coach at the workshop, it was really inspiring by seeing the team grow, grow on its own wings.

Then December came. A Django Girls workshop in Budapest, very successful it was amazing to see that inside this budding Django girls community there were from the original team in Berlin of the attendees, some attendees became organisers, coaches, became friends was really building and growing the community, was sharing eagerly what they had learned and was spreading the love of Django girls further and further.

I also went to Krakow, for a month or so, we had been working Ola, to make on the redesign of the Django website. This project had been restarted from scratch many, many times, this time a talented designer, Andrew, who came up with a good design, thanks of the efforts for Janis, looked like we were able to ship on time. A few things that didn't look right, formatting errors or CSS positioning wonky on the page, or track looking more terrible than it usually does.

I will go to Krakow for a weekend, do an improvised sprint and hack the last details. I went, we ate food, had a great time together, ... we managed to code. We shipped the redesign successfully a week or so before New Year, was really proud of the work we done and happy that we managed to do this euro projects and ship it. Not everybody likes changes, we got negative feeding, like the guy on mailing lists, months of unpaid work, a stupid mess, and the worst website he had seen for a long time. I have bookmarked his e-mail in my inbox, he is comments discouraged me a lot. I took a step back as a result from contributing because I just couldn't get over it. To this day I am angry with this., maybe think about this. The next time you are about to be an ass hole on the internet, behind the code there is a person that wrote it. In our community for better or worse, probably doing it in their free time. Insults are never okay, creates a toxic atmosphere that drive people away, that is unacceptable.

On that happy note, that is how the 2014 ended. I went 9 conferences. I learned that by stepping outside, breaking the bubble that the magic happens. I went to 10 Django sprints, my biggest take away, sprints are horrible, from the name we expect people who show up to know what to do. People who show up and never come before, don't know what to do. That is where the biggest room for improvement. I am interested to see what Daniele is doing this year, we are trying to make it a newcomer friendly atmosphere.

Made to 6 Django meet ups, I love to see the meet up format and every local community does a twist around the format. Motivating me to ... the Budapest meet up. That was dying at this point. Six workshops, Django girls, pixy dust, through this I gained respect for teachers.

I went to 13 countries in the end, learned the privilege of living in the ... area, I didn't need my passport for most of the travelling. I also shared the room 3 times with Daniele, he said he probably spent more time with me than his girlfriend, Carol sorry.

I learned that travelling didn't help against burn out. But meeting friends tanned people talking to them, got me to a position where I could work on this, through honest open communication with my team and time. Managed to start healing and get through there. I am not there yet, but I'm doing better.

If you are feeling something similar talk with your team, your friends, maybe your family, or maybe even a professional, at this conference there will be a wellbeing service where you can anonymously if you wish, so, talk to a professional psychologist, if you have the issues, it is a big problem in our communities and I am grateful for this opportunity that Daniele gave us.

I also learned that behind every line of code written, every meet up or conference organised there are people, volunteers most of the time. Today, our communications are mostly digital, I mean we have to. But it tends to be easy to forget that there is a person on the end of the communication, my advice is practice empathy 1 of the most useful skill you can have.

I also learn that the Django community, Djangoland is wide and full of surprises, to me this strong community is Django's single best feature, it is Djangoland's national treasure.

Ola once said or wrote, for Django you come for the generated admin but you stay for the community. I encourage you, cherish the community, nurture it and find a way to make it grow.

We do this by welcoming newcomers, letting them find a space among us. We do this by building a diverse community because diversity is what makes us strong. We do this by practising empathy and caring for 1 another. With these words I hope you enjoy the conference, I know I will. Thank you very much.

(APPLAUSE).

We are running late so I will not be taking questions but I will be here all week and will take questions but only in the form of emoji. {laughter}.

DANIELE PROCIDA: Thank you Baptiste that was moving, honest, insightful both about yourself, and ourselves as a community, I'm very glad we've opened the conference with you. Thank you very much. {Applause}.

While Hanna is setting up as a reminder we found a pair of classes yesterday so they're in the office. Speaker to a volunteer about that.

We have some attendees who would like to borrow the leads that go into a charger. You know those sockets with 3 - the small ones with 3 pins. I don't know what they're called. They're smaller than kettle leads, like a miniature kettle lead. If anyone has one of those we can borrow we'd be very grateful.

So we're running about - only running about 10 minutes late.

OK we're just having a bit of a fight with a projector which is a conference ritual.

Whilst we're waiting to let you know at 7 o'clock tomorrow morning Vincent is leading a 5 kilometre fun run through Bute Park. I won't be going on that because the only time I run is when I am being chased by my enemies. But if you are the sporting kind you can meet on the steps outside City Hall for a lovely run up along the river through the woods. It is listed on the web-site, so see Vincent, he is one of the volunteers and he'll be leading you. S.

Please welcome Hanna Kollo, she will be tall kind pout how to avoid monoliths in your Django projects {applause}.

HANNA KOLLO: Thank you. Thank you very much and welcome everybody, this is the first time I am speaking at a conference, so please bear with me if I'm a bit nervous. So, today I would like to share my ideas about overcoming one of the biggest anti patterns in software design and that is avoiding building monolithic applications. So I will tell you what I would define as a monolith and I will tell you a story about project I did in the last year. And in that project how we overcame this problem.

A little bit about me. My name is Hanna Kollo. I am a Hungarian. I am a software engineer for professionally like 6 years. We can find some more information about me on those web-sites and I would like to adhere as a follow up on the previous talk that Baptiste was the core developer who contribute my first ever request on Django on one of the Django spills he mentioned in this presentation. I work at spill games a company in the Netherlands, it's a casual gaming company; we have a lot of users and we make our money by showing advertisements to them, we have a couple of web-sites, one of the most represented - representative one is gamesgames.com, they all look pretty similar but are tailored to different audiences and different organisations but basically published.

We use a lot of Python and Django at spill games so our web-sites have service oriented architecture behind them which is built in house. That is mostly done in Erlang the load sensitive parts of it but some of the components are done in Python and the non-user facing internal tools used only by people working at spill games they are almost all of them done in Django because that is just the most convenient for us.

These internal tools, this is where I work, so these are business driven applications which means the users are internal so sitting where the developers are, communicating easily with each other, and this is presenting special challenges in software design. Why is that? Because it's different from working based on a contract where the requirements are clear and it's also different from working towards the millions of internet users where you don't have a contract at all. Here the requirements are very fluid. They are not clearly defined and changing all the time. And when people want something they want it for yesterday. So, it's all very hectic and a bit chaotic and because of this it's hard to keep the code clean.

And you tend to slip into bad practices and you tend to just always make small changes to your code and don't do refactorings and proper structurings and while your code is growing with these small editions you will just end up having one big monolithic code, just everything connected to everything.

So, this is the problem of business applications but for other applications too - ending up with monoliths.

So, what is a monolith and how does it look in general?

In my terms a monolith is a Django project with one single app or one single app which is not a third party. You can still have third party apps but your business logic is just in one app and it's just too big. That's what I call a monolith.

So basically it means there is no modularity on the app level in Django because Django has this very, very handy concept of app S which is the main tool of structuring your code so an app should just be like as a best practice focusing on one single aspect of your code and that is your driving force for modularisation to divide them into different apps ideally re-useable in different projects.

I would like to mention here that in literature monolith is defined a bit broader. It is defined as not having layers in the code saw not having a models layer and a views layer and other presentation layers. Here it is - this is not what I mean, I just mean putting your code in one app, still having layers inside that app I would still call that a monolith.

Why are monoliths bad? It's opposite of modularity right? It's bad because it fits not a simplistic very small app then it is just hard to understand, hard to comprehend, it's just too big to put in your memory, your head I mean, and because of that it's hard to make changes and it's hard to decide what impacts what and it's hard to re-use so all these are slowing you down as a developer. It might be quick at first because you can quick and easily add changes, but on the long-term it is slowing you down.

So, we want to avoid them. And from now on I would like to tell you the story of the publishing project a project which we did - there were 2 iterations of the project, one iteration was half year, the other was full year, 3 to 5 people working on it so you can have a grasp on the size of the project.

The publishing project is about publishing games on our web-site. It's a CMS which is really business specific, we build it from scratch just for our business needs. So theirs the publishing project. The first iteration of this publishing project we really wanted to avoid building a monolith and there was a strong willed architect who designed modular structure upfront, really enforcing that we have many different well-defined apps in the project, and trying to guess upfront how many apps we need in the end goal of this project saw trying to picture how it will look like when it's ready.

So we ended up having many small apps and only just a few models in one app. I mention the models here because I do think that - well, models and apps are related in a sense that models are also a way of modularising an app in code.

These are how they look like, these are apps and they do relate to each other mainly by the models in one of the apps referencing the models in the other app. So this is not a monolith but it actually has similar problems. It's just everything referencing everything. It's hard to comprehend. You really need to know all the apps or most of them to be able to comprehend how your code is structured and how to make changes.

This turned out to be over killing of modularising.

So we were so lucky to get a second chance and we started the project again. There were also scope changes but I don't want to bore you with that. Let's just assume that we had a chance to re-do the whole project from the start.

We took a different approach. We started with one app. And we really fought back creature creep and provided minimum working product as first and then me started adding features to it but we were really then strict about wanting time to do it right. So not just quick and dirtily add features to this existing app but really consider if it makes sense to split it up into a new app and if it did that we also pulled some other stuff with it from the existing app so it was an incremental approach.

And we ended up having few larger apps and in having many models in one app and whenever we created a new one it usually referenced the old one from which it was taken out from so it ended up in a tree like structure of this.

So, if I would like to picture how the transition would be from the first phase to the second phase, these are our apps referencing each other through models in the first iteration and then we just grouped them and we ended up having fewer apps and fewer references within those - sorry between those apps.

So, we found out that this is better because it just splits up the comprehending of the project in 2 levels. You first have to comprehend only the role of the apps and the relations between them and there are not so many so you can still know about all of them and keep all that information in your head and then you can zoom in and it is - if you want to create just one new feature, you will most likely be able to do that by touching just one single app.

So, why is it better? It is not a monolith. And it facilitates growth. I describe that tree like structure of apps and when the requirements are clear you really need some way to facilitate growth because the requirements will come on the way.

So, the main structural difference between the 2 approaches is that having few models in many apps or many models in few apps and I found out that the second solution is better to this kind of problems where we - what we had.

The driving force of placing code is the business logic. You want to implement a new feature of business logic inside one app or inside a contained space in your code. And this is why the second iteration worked for us better because for each feature you have to add you needed to touch only one app in like 90 per cent of cases.

You might think that yes but there is a problem with size here because the apps then grow bigger if you want to keep the number of apps under control then the apps will be too big themselves and then you will just shift the problem of understanding from the structure on the high level to the structure on the lower level and you might be right with that and we also faced this problem so we solved it by introducing some structure inside the app.

And this is again about where to put the business logic. And our solution is somewhat different from the general best practice in Django, general best practice is to create thin views and fat models so put your business logic into model managers or code which is closely related to the models. Our best practice is to introduce a new service layer, I found this out in, an article and we expanded on the idea to have one extra element next to models and views and templates and that is the services layer which is binding the views to the models and all the business logic should be in this services layer.

Why we don't put this into the model managers? Because our business logic is not only about models; not only about data, there are some examples which don't touch models at all or not directly or not bound to just one main model. For example exporting some data through an API to another place or sending e-mails or opening the doors this is why, this is why we introduced the services layer and it served us better than the model managers, I found it is easier to do it this way.

So I am just showing you a little bit of code of how we use the services, this is an example of the model. The service library and one view a simplistic one. Our coding standards say that you can use the models directly for read only purposes like the first line showing. You can use the models through the service for reading them if that reading contains some business logic.

If you want to update them you really have to use the models through the services layer. This worked very well, very nicely with us.

So my summary is, we had to iterations, two options to fight back monoliths one is create many small apps and the second one is create fewer but larger apps and we found out that actually, the one listed up here first, the few apps with many models is better. Because it is, facilitates growth because you can grow a tree like structure of apps into your project.

We have introduced the services layer which is not our own idea but works very well in this context.

This is what I wanted to say. (APPLAUSE).

VINCE: Thank you very much, if you have questions, please come the the microphone.

FROM THE FLOOR: Intrigued with moving ... the small number of apps ... is that moving deck chairs on the titanic, it doesn't change the way the models interact or do you have another level of, I don't know discipline or sort of rules about how one model is allowed to ... when they are allowed to move applications. Where did you see the architectural advantage from reducing in number of apps beyond having less apps in the system.

HANNA KOLLO: I will try to understand the question. You need to understand much smaller, you really need to deeply understand much part of the code base to be productive, is that an architectural advantage?

RUSSELL KEITH-MAGEE: I suppose it is, what I am looking at, is there, was it increasing cohesion inside just that model so that model is almost entirely self-contained as oppose to be there being.

HANNA KOLLO: Increased the cohesion within the app, we didn't formally specify any guidelines to do that it just happened because of this, this growing nature of the project.

RUSSELL KEITH-MAGEE: Informally, ended up, no cause inside the app but, you could have enforced that if you needed to.

HANNA KOLLO: We didn't enforce it, we need to be pragmatic but by the nature of growing the project this way, naturally supported us in having, in increasing the cohesion.

FROM THE FLOOR: Hi thank you for that. I am interested by your views of service layer I have seen lots of people doing this kind of stuff in forums, like you said in managers. Have you found any good patterns to have the services?

By extension, do you think there is anything that Django could do to make this more obvious and make our, all of us architectures better?

HANNA KOLLO: Yes. Well, I only found out that it is just, I am always designing when I design new features, I start with what services, business logic services I need. That is what I develop first and that is what I unit test first before writing any views or html code for that, that is our best practice.

As for what my suggestion for the Django structure would be, I think not as experienced to make a suggestion there. However, I do have to mention that, there is this concept of model view controller all over the computing world. I do feel that here Django is using the terms differently than, than the other, other frameworks for example, in my, in a general view what is called a view in Django should really be the controller. Then it is just makes more sense to have a services layer as something related to the controller, closely related to the controller.

FROM THE FLOOR: Thank you very much.

FROM THE FLOOR: Thank you for your talk, do you manage apps to and one to ones?

HANNA KOLLO: Repeat it?

FROM THE FLOOR: Do you use the same repository for all the apps?

HANNA KOLLO: The same for all the apps yes.

FROM THE FLOOR: Like Russell was saying, nothing to enforce that you don't reference, yes okay.

HANNA KOLLO: Only common sense (LAUGHTER).

FROM THE FLOOR: Thank you.

NEW SPEAKER: Let's thank Hanna.

HANNA KOLLO: Thank you very much (APPLAUSE).

VINCE: Thank you very much.

DANIELE PROCIDA: Tim who gave me the lead, wave your hand so I can see where you are sitting.

VINCE: Okay so our next speaker is Abdulrahman, he works for the (INAUDIBLE) organiser of the Kuwait ...

ABDULRAHMAN ALOTAIBI: Thank you. (APPLAUSE).

Morning everyone. My name is Abdulrahman, if that is difficult to pronounce, call me Abdul. So I know it is very difficult.

At the beginning I would like to thank everyone for helping me and supporting me, and for ... GDP Kuwait.

I will talk about me, then we will take it from there.

So back in 2008 I left home to study and pursue my bachelor degree, I left Kuwait to go to the United States and I wanted to start afresh, like I wanted to have a fresh start. So, I dumped the operating system that I was using and I adopted Linux, the reason I was surrounded by people who liked Linux, they saw (INAUDIBLE) believe it or not.

Back in 2009 I was struggling with C plus plus and introductory programming language, so one of my friends told me to go and read dive into Python.net. The book it was great. I did.

Then I joined the TWUUG, ... it is ... users group, then I joined 757 LABS which is a space in north Virginia.

In 2011 I saw talking about Django, I was debating myself, ... and Django, that talk made it for me, the snakes and rubies, it is on you tube, look it up.

Then I went to PYCON it was amazing, the community was supporting, everybody was smiling, take a photo with Guido, I did but I lost it.

So in the same year I graduated with ... degree, I did computer engineering and electrical engineering and, yes, you can trust me! After that, I joined Kuwait institute for scientific research, I was lucky to be one of the guys hired back then.

For anyone who doesn't know where Kuwait is, it is here. So, we are between Saudi Arabia, Iraq and the gulf separates us from Iran. We are right here right now [UK].

So this is where Kuwait.

Okay. Kuwait is very small country, so we are about 4 million total in population, so if you find a programmer walking down the street that is like one out of 3 million that is a lot in percentage wise.

I also joined Google develop as a group, as a member. In the situation in KISR was different, we depended on open-source, we didn't have any open-source software but using the word open-source was forbidden back then. We always see reports like that. So reports underscore virgin underscore final, final, final, final, final, final even though that the word processor had an issue tracker or a document tracker nobody used it.

All the software was written in a framework that is starts with dot. All the application were written for the desk top application, that is 2013 so I wanted to change that.

So what I want, what I did is I wrote a law to change that, me and my colleague. We wrote the constitution. We said, okay, for KISR developers you need to follow these rules, you need to have a way to produce your work either by using dock ... or using a virtual machine, you need to use some sort of virtual control, we don't care which software you use, but use one. You need to use a DB it is very difficult to process CSV along the way if you have production software, we have production software that relies on text files.

You need documentation software, also you need logging and issue tracking and communication protocol.

So, writing a law is very difficult and enforcing the law is even harder. So, we stripped down that law to become three courses. We break it down to phase one which is developer tools you need to adopt, this is basically opinionated this is what we knew then and this is what we have enforced. We said, either use virtual box or virtual box and vagrant to manage your VM's, you need to know how to use CLI grab and how to navigate, you need to know how to use package managers and so on and so forth.

The second phase we took Daniele's don't be afraid to commit and we put it there. Basically, how to use ...

For the last phase, basically we tried to apply all the knowledge and throughout the course, to build up a toy projects.

Then a year and a half later, KISR announced they are going to adopt open-source. I have a video here but it is in Arabic, I play it anyway. (LAUGHTER).

That is the, we made the news by the way.

Okay.

So, right now we have three Django projects running in production, there is a Django training course happening right now in KISSER.

A journey of a thousand miles begins with a single step.

These are the links and if you want the presentation, it is admin at Q 80.github.io (APPLAUSE).

VINCE: Any questions?

FROM THE FLOOR: What was the biggest challenges in having Django in a place where in a place where people are used to....

Or things.

ABDULRAHMAN ALOTAIBI: The biggest challenge with that was trying to force it on the law. So basically, we went to the constitution changed the policy and tell everyone to use it. That doesn't work, so, so, we tried to do the up down, methodology and then we reverse it with the course to go from the bottom up and that is what gave us successful story.

FROM THE FLOOR: And how much of a challenge was training people to the Python environment and getting everything installed and deploying a ... application compared to running a JAVA application, did you find it more challenging for people to get running?

ABDULRAHMAN ALOTAIBI: They used the.net. They don't use java over there.

And the situation back then was, everyone was developing for desk top applications, so there was no web applications.

FROM THE FLOOR: A follow up question, thanks for a great talk. How did you keep your colleagues motivated when kind of enforcing them to change the way, how they have to develop software? I mean I would be, I would feel pretty uncomfortable if I would have to be, if I were to be forced to use some certain application in terms of developing software for the company or the research centre?

ABDULRAHMAN ALOTAIBI: Yes, the important thing is you should keep smiling (LAUGHTER), if someone asks you, if someone asks you about something, don't embarrass him and say, hey I told you five times, 5,000 times, just like keep smiling and tell him like, let's go through the steps, one, two, three and so on and so forth.

Thank you very much and have a great conference.

(APPLAUSE).

DANIELE PROCIDA: Thank you very much I want to introduce one of the things more doing at this event. We know a lot of people in this industry who have suffered from exhaustion or depression or burn out, sometimes because of the work or the nature of the industry, Baptiste was very candid this morning talking about his own experiences. What we're going to be doing here is tomorrow, Wednesday and Thursday we have anonymous free counselling sessions. You can make an appointment. The way you'll make an appointment is by taking a post it note with a time on it for a 25 minute slot from - they'll be put on the boards at the back so you walk past, pick off a post it note with a time on it and that's your appointment and they'll tell you which rooms you have an appointment to be in. This service is being provided to us by Cardiff University’s wellbeing service, very generously they're not charging us anything for it, even though it's a substantial amount of time for them over the course of 3 days.

So, we think it is important to raise these issues and make them public and to make it easier to get started in dealing with the issues that some of us face in a more open way. So, I hope that is going to be useful for some people.

In return for their services, what we've said is that during the sprints we'll put together a group that will collaborate with them with the wellbeing service to see if we the Django community can help develop or perhaps start developing an application in some way that supports their work, so we kicked around a couple of ideas, a personal wellbeing tracker, an appointment system, it could be anything, but they'll be there at the sprints on Thursday to have a chat with a group of people who'd like to give something back to them for what they've provided here.

So, I would like to welcome Vicki and Amy from the Cardiff university wellbeing service just to introduce themselves and tell you a little about what they do. {Applause}.

After this we will stop for a coffee break.

NEW SPEAKER: Can everyone hear me OK? Yes fab brilliant. So my name is Amy, I work within the wellbeing team, we're a counselling health and wellbeing team so to give you an idea of who you'll be working with this week it's myself and Vicki today, but this is our team and these are the faces you might be seeing this week if you choose to take us up on the offer of having the 25 minute appointment so we've got Vicki, student counselling manager, Susan our senior counsellor, we've got myself, I'm a wellbeing practitioner within the service and we'll also have Eleanor and Jo who are both wellbeing practitioners and also counsellors as well.

So, why are we here? It's already been explained a little bit but to give you an idea we really want to wake up your wellbeing. We understand a lot of people within the workplace can be stressed or feel quite low or feel anxious so we want to give you a few ideas of how you can improve that if you want to. So, we're here to give you information on managing stress, to be honest if it's not stress, if it's something else we're used to dealing with a variety of problems so whatever it is that might be causing you a bit of difficulty at the moment we're happy to talk to you about it. We'll hopefully give you some advice and we can point you in the direction of resources as well you might find useful to use. But it's just to give you that one-to-one support and to give you a few starting points to improve your wellbeing.

So when I talk about wellbeing to give you an idea what we mean by that, it's quite a broad term and it can cover a number of different areas so you can have your psychological wellbeing which is how your mind-set is and how you feel emotionally. Physical wellbeing is more about your physical health or whether you are able to engage with physical activities that improve how you feel. We've got spiritual wellbeing, financial wellbeing, environmental wellbeing, so that's one that might affect you in the workplace so if your work environment is quite stressful or if you've got a really messy desk or something like that there might be things that start to affect your wellbeing at work. Social wellbeing so how your social relationships are and occupational wellbeing so that's another one that is quite important within the work environment.

So, why are we talking about wellbeing? Why is it important?

Well, wellbeing is everybody's business, it's not just a place for people we're in a counselling health wellbeing team so we talk about it a lot but it's not just our place to talk about it, it applies to everybody, it's linked to the workplace so you think about productivity and performance did you know more than 40 per cent of employees are neglecting other aspects of their life because of work? Which may increase your vulnerability to mental health problems or just stress. When working long hours more than a quarter of employees feel depressed, a third feel anxious and more than half feel irritable. Then the more hours you spend at work the more hours outside of work you're likely to spend thinking or worrying about it.

These are just a couple of facts to get you thinking but I suppose that's what we're here today to get you thinking about your own wellbeing and if any of this might A apply to you. Are you in the 40 per cent of employees who don't have a good work life balance?

I'll pass over to Vicki now.

NEW SPEAKER: Thank you. Hi. Morning. We've got a different meaning for REST but I was interested to see the word {laughter} on Django. Very different meaning! {Applause} so, about our meaning, I'll go into that a little bit, but I'm going to get you working very briefly on your own don't worry. I would like you if you've got an app on your i-pad or old-fashioned pen and paper to write down 5 things which are really important to you in your life. Go, you've got a minute! You're not going to share it. Don't be shy. You're just going to think about it please. 5 things. Some of you look bemused, she's asking me to do something!

When you've got your 5 things I'd like you to think about you know which one rates the highest and draw yourself a pizza or a pie whichever you prefer and give each 5 thing a slice depending on its importance to you. So, here is one that somebody very stressed did.

So can you all see that from the back? The blue is relationship. The murky yellow is fitness. The grey is friendships. And guess what the orange brown colour is? Work. OK.

If yours looks anything like that, you may wish there was time for other things. We'll come back to that in a minute or 2.

OK Amy said that stress is quite common in people who spend a lot of time at their desks. Or on the bus. I checked my emails this morning I wonder how many of you did that as you got out of bed today?

So, our office is everywhere.

So, for some people, to keep their work at a level they feel it should be, they slip into perfectionism which is a clinical perfectionism, not the one we want which is really helpful, gets us inspired and motivated, but a bit further along the line where it causes us not to have any fun anymore, we just feel pressure.

It may be that we are not assertive, our self-esteem is down because other people seem to be achieving more than us. We might be drinking too much or taking non prescribed drugs.

Just not getting the right balance. Loads of things on that slide.

So, as Amy said we're here to talk about self-care a little bit, not to get drowned in files. It's an old-fashioned picture isn't it it's all on i-pads or - I'm an iPhone addict as you can tell, everything is "i" sorry, OK.

And there is also something about sitting at the desks and not getting enough serotonin or dopamine into our systems because we're not moving so just a walk around the block a 10 minute gap every hour I would suggest to our students that are studying really hard to breathe in some fresh air or run round the library - outside - actually increases your serotonin and makes you feel a bit better for a bit longer than the round lasts. Yes. Of course,, it's good to be silly, it's good to have work, rest and play, so I don't know about the head bandage but you know ... ok.

NEW SPEAKER: Some of these are relating to how you get yourself care so things like getting yourself enough sleep so you're not falling asleep at the desk at work, making sure you are eating regularly and fairly healthily, I know all of us can neglect that at times, and trying to get enough time in for family and friends and your gym or exercise routine or if that's just as Vicki said walking round the block as well.

NEW SPEAKER: And you know how many of us have friends on social media and how many of us meet friends down the pub and a restaurant? And how many of us look like that at the end of a day?

So, caffeine stops us absorbing so much so serotonin and causes jitters so if you are really anxious it may be you need to move your caffeine levels down a little, don't stop completely because you will go the other way and get jitters from withdrawal but reduce your caffeine a bit and it dehydrates as well so think about that.

You were told there are post-it notes so don't forget we are here to get you started.

So how would you like your pie to look in the future. If you'd like to make some changes, if you'd like work to have a balance with rest and play then what would you like your pie to look like? The last person thought this ... I don't know about pot throwing but they were going to do some dance lessons.

And work got squashed a bit but actually more productive because of that because there is more going on. We can think more clearly. Our cognitions work better when we've got the balance right.

NEW SPEAKER: So this literally the last bit so if any of this is sounding familiar at all and you think may be your wellbeing is in need of a bit of a shake up or you literally like some ideas of how to improve wellbeing, it might be your wellbeing is absolutely fine but you want to know a bit more then we are offering those 25 minute sessions so that's Tuesday, Wednesday and Thursday and times will be available for you so feel free to come and chat with us, we'd really like to see you.

NEW SPEAKER: And we are dead excited about apps!

NEW SPEAKER: Yes please. . . {Applause}.

DANIELE PROCIDA: And thank you very much to both of you. Thank you to Russell who suggested this idea to us and thank you to Cardiff University for doing this for us. As I said yesterday, please find and mention Cardiff University to thank them publicly on-line if you can because they have really got behind this conference.

We're going to go for a coffee break now. We're running 10 minutes behind schedule so we'll try and catch up a little bit, so if you can be sitting back here by 5 minutes past that would be great. I'll be putting up the appointment signs sometime later today.

(BREAK) ...

NEW SPEAKER: Hello, welcome back, next speaker is a Django Core Developer, organiser of many events happy to say one of my colleagues in potato and not least a cofounder of Django girls, talking about her favourite subject Django admin, welcome to Ola Sitarska ...

OLA SITARSKA: So hey everyone, I am really excited to stand in front of you, and sere your brains for 40 minutes, to talk about admin. I have a very special bond, this is the feature that convinced me to switch to Python and Django in a day, I was 14 and spent most of my days playing on-line games. ... (INAUDIBLE) try for nine months and then raise your child for the rest of your life. So that was my kids, was very curious and I was crazy, how hard to build your own website and game., the first site I wrote was an admin interface for a game. That is the story of learning PHP and crying when boys hacked my games, a few years later, I knew PHP a bit better, they weren't trying to exploit the website a bit more. This was a lesson learned they didn't want to hurt me f spent my afternoons building interfaces for my games, so moderators and other admin cans man analysis the website.

That was frustrating, I thought I was wasting my time, build a better way to change a diaper or something, then out of nowhere, a friend showed me Django, I had no idea of Python is, at six times, years ago, I was sold.

I haven't written a PHP after that. I am kind of proud. (APPLAUSE).

So today I want to share knowledge and secrets of six years building interfaces in Django, I work at amazing place with talented friends called potato, come from Poland, living in London for the last year, organise Django also. A member of the core team and board member of the software foundation, I also called one of my cats Django, so yes, I am a Dork!

I want to talk about three things, first explore the insights of the admin and discover and how messy and beautiful it is, I will share a story of how we built Django girls, how we went from this to that, so kind of, I will show you how it works and we will be able to make it useful for us and organisers who organise the Django girls events.

(INAUDIBLE) Django admin I encourage you to stay. At the end we will talk about the future of the admin, we will see.

So, let me dive right into the first part, the dark and mysterious forest how Django works, since the very first incarnation of the framework, cut to the chase, Django admin is not one of the most beautiful parts of the code ways, it has been harder to get your head around at first and messier part of the code base, but it is good for backwards compatibility, Django admin is nothing else than some views and URL's and models and it is just the usual Django application, that is what I imagined when I first saw it.

You probably couldn't be more wrong if you think is. I think the current state of affairs looks so much different from what you can imagine so let's take a look and see what is there.

The Django admin contains three main classes each of them at least 500 lines long, these are admin side which is basic configuration admin beside the whole thing., multiple classes and interfaces if you want to have a superadmin tool, you can also have that, I will show you how to do that in a second.

There is also a model, the class the most scariest one, the definition will give you a picture of how big it is. Admin class encapsulates all models and functionality for a given model. That takes care of everything, almost 2,000 lines of code.

Then the last, the change list, the view of the objects in the model and also the change list view. So there is no URL's, no views, no models and it looks nothing like you would expect. So to fully understand how admin works, go through the code and see if we can find something interesting.

So the admin side as we said, encapsulates an instance of the Django admin application, ready to be hooked into your URL. At the end of the definition, this is side equals admin site, default given by Django, which you use, when you include URL's, basically just use this class without even knowing it.

Of course if you want you can also create an additional admin, just by creating another instance of the class, flagging it into your URL's config file. I said there is no URL's, so where does this come from? If you look at the source, you will see the admin site, URL's, this is where it comes from, get my URL if you look deeper, configurations for your admin site. Two parts, the first one being admin site wide urls, pass word change, view., this also mean that is the admin users its own authentication system not the one that is available for everyone in Django, you had the views so, all of this views like log in and log out and index, are metals on the admin site class itself. I highlight the url parts, the wrap functions, checking is checking all the permissions and seeing if you are authenticated to see this parts. And the self-logging of course in or out.

The second part is created dynamically and consists of urls that are connected to each of the models. This is basically links to the list of all of the objects in a model or the list of the edit from or at from and all of the things T code responsible for that looks through all of the research model admin classes and then create a url patch list with url created out of the ... using the API and the model name and then includes the urls for each model.

The next part of the admin site class, the registry, some models are registered with the admin site using the register, so every time we do admin site register in your code you actually use the admin site instance again, without even knowing, so since Django 1.7 you can do it with a decorator, as you see, takes care of registering model that is will be visible on your admin site.

One thing I found out when I looked into the code is that you can actually, don't have the create the book admin and admin class, ... look at the model and it has the default settings, if you want to see, this is probably the fastest way to have your admin enabled for a model.

The most exciting things, options, list display, (INAUDIBLE) just fine to, I thought this is probably not the best practice, I don't know if you want to change one thing, this may be confusing for some of you, I wouldn't advise you to do that.

The last thing here on the adminsite class, change things in your adminsite, there is not many important bits but the con fig lines, define header it is title displayed in the title tag in your html, change override H one tag that is on the admin page or the title at the top of the admin screen if you want to make it more yours. Everyone is like, what is this Django, I want to add it, not the Django thing, you can hide it from the managers or the clients.

If you are ready to have your brains boiled a bit. Let's look at the belly of the beast.

Model admin is exactly 1,710 lines long, lives in the options file, I didn't lie to you, when it encapsulates all, now we have got, we are not looking at the admin side as the whole, one model. To change the view, all of the view that comes with that. So, same as before, notice that model admin also have the urls method and everything you can imagine except maybe fielder’s lives on that class, all ... responses, are here in.

However, the class also makes it easy to override things as you will notice later in the talk, we will exercise this.

If I would have to, if I want to explain everything I will probably need to like three hours or so, I don't have that much time. I can give you an overview of my favourite things, you can find, the documentation on the part is very extensive and good if you are curious of the fun things to do, go and look at that.

Here are my favourites. So, there is an actions on top and on the bottom. If you have the long list of objects, if you have a long list and you want to scroll to the top organisers the bottom, default around the top. If you like check, check, check everything, scroll to the top, also turn on the actions on the bottom. If you are like crazy or feeling particularly, like living on the edge today, turn on both of them and I won't judge.

Field sets which are a good send when dealing with huge models, can save a lot of confusion, spend five minutes to structure properly or even hide from some fields so they are not visible for everyone.

Here for example, I grouped into the three field sets just like the one has like a description, so it is more obvious what this function does and the default different options. So it is quite easy to do that and I would encourage you to.

There is also a neat option, list display links that, allows you to specify which fields in the objects table are clickable and they are access to the edit form, sometimes I don't know if you think, first column is confusing and don't want to -- you can also turn on other which is, direct the edit from.

There is also a list of the, list editable, which lists are editable for the change view, this option is little bit dangerous, backtracker issue raised six years ago, maybe they will move it this weekend, the number, is 11313 if you are interested.

The tickets, so suggested that it is currently not recommended to use it in a multi-user environment, if you have more than one person could be editing this at the same time, can easily override and save the old version or even save young values on wrong objects if people order wrongly.

That is dangerous, I encourage only to use that if you are only person using that.

Another one if you need a different default, the other place, you don't have to hack on your ordering in model, you can specify on admin tool which is neat.

There is one neat and another very popular called radiofields, here specify a list of fields, they will show you up as radio buttons, so you don't have a list, but you have radio fields, so if you are sure that this field will only have a few options it is useful because it saves you the one click. If I was doing the screen, I was doing the screen shots, fine, I got that.

The last one, the one I never heard of until I looked into the code, it is called save as., that is save as, I think the correct name should be save as new. That would be less confusing.

When you create a new object, instead of saving the old one, to save as option will turn on a button here on the left, right button, part of the screen, so you can actually if you decide, maybe I will save this as a new object not override the old one, click that, that will create a new object which is useful.

The last one, I want to show you, show full result count. For example, this basically counts the whole table and shows you the number of objects, it can be useful to turn this off, ... (INAUDIBLE) basis for doing the account on the whole thing is quite expensive this is useful to turn off.

So there is a lot more you can do but as I said, I will go deeper to explanation into the options in the second part of the talk, I will not explain everything.

This is the third and last class, change list. As you heard model admin class basically something that also includes the change list view, this method, despite being the biggest model. I don't know, just, didn't fit in on the screen, can't do anything on their own. Also change list, class to help with fielders searching ordering pagination, so you can see the correct things ton list of the, it is powerful and it can do all the stuff with it.

If you want to change the way your objects are displayed on this list, this is probably a place to be however, as you will later see in this talk, most of things you may ever want to need to change are on the model admin class, so if you come to overriding the change list methods you are (INAUDIBLE).

There is a current list of ... admin doesn't rely on many other parts of Django, managed to stay in the same for many years, backwards concat has its own costs, I wouldn't be surprised if you can run the admin from 1.0 to 1.8 today, I might ... one of the biggest reasons is where Django admin structured this way, the fact that by the time Django admin created, the first practices of creating Django applications weren't established by then, we didn't have the ... size and complexity of the admin, it is so dam difficult to refactor it, it is a huge pile of code, I don't think that one person can like refactor it and stay sane or do it like in one Django release or something, it is a huge pile of code and it is difficult to refactor it now.

So, these are the hinges, there are some things, I must ever do when I start a new Django ... this is some secrets how we let 120 organisers from all over the world manage the website. This is a short case studies of changes, implemented to make mine and their life easier.

So, for those of you who missed the memo, the Django girls is a non-profit organisation, officially just started yesterday. So that is amazing! (APPLAUSE).

It is basically in the UK and we started it together with Ola and last year at ... how awesome? It is the field ... show them they can do it too.

So, we started to do this six months ago at the last one in Berlin, 120 organisers from all over the world put together sixty workshops we thought now, 1,245 women at the workshops, been to North America, south, Asia, literally all over Europe which is incredibly amazing, each of the events has their own websites and all of them are managed from a single Django interface, let's look at the website.

Basic content management system we wrote from scratch in Django, on the home page a list of all of the events, click on the event and you go the event page. This customer is able from Django admin, publish in different languages, Hungarian, there is a website from Taipei, without implementing like the whole translation system in all of these things, so each local organiser of the event has access to the admin and can edit only their own page without having access to all of the other stuff, that could be dangerous.

So the first thing we modified to make sure that organisers only have permissions to the things they need. It is like a common privacy thing for implementing is building more advanced, the most important thing was that organisers of event in Cardiff only see an event from Cardiff and nothing else.

So here is how our admin looks like. We actually using we are using the package called Django suit to make it more - looks like more I know nicer, this also has a nice sight on the left so it is easy to juggle the other stuff if you want to quickly so we're using that stuff, that is really important in a sec. We have a Django girls maintenance team now 6 people that basically have the admin access on the web-site so they can create web-site for organisers, manage and help them edit it out too so the 6 people are super admins they have access to everything but all the other people have just access to their own events.

As an example I'm going to use here {inaudible} who are here in the audience and were in Cardiff yesterday so if only want to see events for Cardiff here today not all events happening over the world. It's a short and nice way to do that. What I do is over right and get query set model admin class - I hope the code is visible on the screen. I didn't have time today. Yeah the contrast isn't the best but I didn't have time to change all 120 of my slides! I'll put that on-line after so you can check it out then.

So, there is that we are overriding on our event admin. First of course we're going to get the queries that mill return in this case and we're checking if user is super user so just return whole query set otherwise we filter it out by team so only show events in which the current user is a part of the team.

And that works quite well. So {inaudible} all event.

That was easy enough so now see what happens if I want to modify the object.

So what you realise for example on the list in the form here you can still see all the pages and can over right something and say something that doesn't belong to your event.

So that is quite interest and we didn't want to leave it like that especially if you like we Django girls go to 60 events and this list is quite long, nice problem to have of course but we should solve it. So, quick look into the insides of the admin shows me there is a model a model called get form which does exactly what it does, it gets the form you see displayed on the screen here so added object form.

Of course as admin person I first over write and get the super hold so get the real form and then if our user is an organiser not a super user then we override the queries on a field page in our page event and we limit only event pages when that user, current user, is a part of the organising team.

So, there is that. At the end of course we return the form to. And this is how it looks right now. So it's simple for them they see they have only one choice and don't have to look through all the lists and search and do all the word stuff or anything.

So the next thing we modify when it comes to specific permissions was organisers can edit their page after an event takes place so the first event it was all great and awesome but then they wanted to organise second events in their city and didn't want to bother us and started replacing old web-site with new content and changing the days and we lost track where all events happened and when all this precious information so like how we - we saw our {inaudible} disappear and were sad so we decided to {inaudible} software because hard to core co-ordinate 120 people and decided shouldn't add stuff in so we decide to block it and don't let them so they need to bother us write us to set up new web-site for them. So documentation shows there is read only fields so I thought yes I'm going to use this, this is exactly what it says. So if you put something there like name of the field it won't be editable into admin but it will be visible, exactly what I want except we want to depend on the permission level of the user and our current state which turned out to be trickier and I wanted to magically make everything - without specifying which fields should be read only I don't have to worry about this in future if I move or change something. Want to get it over with.

That was trickier and it took me half an hour to hook my hack into different places of the admin I learned deaf get ready fields is pretty {inaudible} with some other parts which led to some dependencies all the time and finally the brand new meta appeared Django 1.8 came to rescue and this current code looked like. Again overriding that get read only fix method. This now takes an object to - I don't know - I am not sure where - now we check if the user is not a super user so if this is just an organiser not us and then we check if that event is in the past so it's not coming. And if so we turn a set which is just a name of all of the fields available in this model and can see the meta here so it's really simple to get this information with other fields in the model.

And then we return the read only fields if nothing changes and dah-dah! This is how it looks like with all the fields set for read only so no {inaudible} content and after event is only they can't add any more.

That was a nice feature that saved us a lot of time in the past 2 months.

My next favourite thing to do is making my life easier by creating custom fields on the display change list fields. Can be perfect for how things work in a code but not always perfect for displaying in admin for example if I'd like to quickly grab emails to all organisers of the event in Prague but in the database is like many to many field that's connected to the user model and email is there so have to do like 10 clicks to get actually information like who is organising this event and where are the emails.

To solve this I show you couple of examples we've got in Django girls now. First example I just described so on to cookie, who is organising this event and email them with all the clicking like tells them of times. Second copy and paste and put G mail and send them email easy. I did this but creating a method on my event admin called get organisers and then that takes current objects, can see an object there, this is basically always past in all cargos on model admin cast like ultimately by Django then I look through organisers of the team and email list and I join the list to be a come on {inaudible} because that's what Gmail likes and at the end there is a little bit called short description so - I sent it to organiser basically a huddle on the change list table you see it says organisers here and I have a list of organisers and I can copy and paste to Gmail and it works and a nice feature I always start to when I see some information are like too deep into our relations and stuff and I always bring them up.

Another thing good use for is displaying quick information like this event is over or not based on the date. Bad with dates and calendars so I need these kind of things! Again it's a simple method on the admin class it's called is event over in my case, just get object again and say if event is over or not.

Also needs a little thing at the end and I set actually with boolean on it. This displays it as small icons less or no instead of text true or false so this is neat I like that.

And last thing I modify is to make it easier for organisers to up-date - link correctly - like to their names without too many clicks checking their picture is OK so to do this I display the picture itself on the list and this again, nothing more fancy thing, proves examples, it's another method called debt logo display and takes a logo, returns the logo, if there is no logo, it returns no logo. Also has an attribute allow tags at the end so it renders the HTML. And this is how it looks like {laughter} ... yes ... yesterday,, that was only picture I had of people lists. He's gloating a lot so I think he deserves to be there! Move on!

One more easy thing you can modify which may be obvious for some of you but I was personally surprised by numbers of people who didn't realise it exists and it's custom action to model admin class so here I wanted a way to publish a couple of web-sites on our home page at the same time like clicking to each of them editing and saving and I just want to save myself sometime and I was {inaudible} I'm going to show a little piece of code here. So first again you write a method on your model admin class and it's basically I show this show on home page, it takes request and queries it. Of course in this case you can set it more objects not just one object, that's why it takes queries inside object. And then I just up-date the query set to set on home page is true and at the end added a short description attribute this method to kind of display, this is the final what's displayed on the select down list and that's really it.

Last thing I'm going to do is just set - I put this on a list in an actions in the event admin and it works so that was really simple. Like I don't know 2 minute change maybe?

So, as you can see there is a lot of simple small hacks you can do to make your life or life of your clients easier and I encourage you to do your own little experiments but you need to make sure you are not pushing too much. The most important question you have to ask yourself is when to customise admin and when it makes more sense to write something from scratch something custom. There are people who believe you should never customise admin and people who believe you should base your whole web-site on admin and do everything because it's going to be faster. I don't believe in any of those things but into a good balance and common sense and I want to tell you about how the common sense looks like for me. So, like when you do something then you don't want - blame it on me then. . So don't. You shouldn't use Django admin as your end user interface. That is I think should be not allowed and I heard stories where it ended badly then you have to sing like a couple of months or years of {inaudible} to be save from scratch. It allows you to easily look into other ways and modify them. It also good for actual managers of the web-site. Because if you structure admin nicely the way I did here you can make changes without asking for help but it's no good for making interface for end users.

I heard about people who decided to do that and it's not a good idea and didn't end well. You shouldn't customise Django admin when you see the time to modify is longer than writing something from scratch. Admin is nice but is limited and you should be aware of that and kind of look under the hood and see how it works there before you start customising everything but of course there are times when you should do that and whenever you see a small change will influence the way your managers manage a web-site will save you time energy and nerves. This is by far my favourite part and why I do this and managers of your web-site will be incredibly grateful if you make their life and experience better.

Whenever you are afraid you are giving too much power to people it's always worth making sure they can't break a web-site by misreading something. Only make sure they see things they only absolutely need to see, it makes their job easier, they don't have to wonder how it works or figure it out and it's a basic rule of thumb but it's important to mention them, as I said don't blame me blame it on a girl who got {inaudible} yeah I'm going to have to ... yeah ...

So that is it. Inside of admin looks scary only for a moment then it opens you up to all of possibilities. Still a bit limited though and it will be easier to extend admin and many people do so there has been quite a few serious attempts to reveal the whole admin from success and none were successful yet. From the recent one there is a re-write called Django admin too stressful EP Is boot strap - the last few years, can show you how big a task it is to actually do it and the reason why admin hasn't rewritten yet it's not because Django {inaudible} doesn't want to merge {inaudible} great I believe it should stay this way. So Django admin has incredibly back company it's hard to beat that and but {inaudible} tricky {inaudible} and I belief the first time someone said publicly Django admin needs re-write was 2010, sometime ago. It's been 10 years it's classic time less design and we all kind of love it and it doesn't mean you shouldn't if you don't agree with status quo you shouldn't try. Recently someone updated CSS of their admin interface to make it more 2015 and this is how it looks, me use it internally Django project.com for a couple of months there is a pending request for 627 I think that's the number to master of Django and we got {inaudible} from the Djangocon team. We're already using at Django project and love it and what I hear it's going to be published in Django 129 and if you don't want to wait and most important you want to help us test it improve it before it lands you can get {inaudible} Django flat in.

So I believe in the basis of this project was the fact I didn't want to re-write everything and break {inaudible} just like 3 smart and change CSS to make it look modern but {inaudible} the same. There is willingness to do something and if you have some ideas don't be afraid to balance them off Django developers mailing list or come to sprints.

Yeah that was about it. It's just the tip of the iceberg when it comes to the admin topic so if you have any questions I'm going to be here for the whole conference and sprints and that's all I got to say so thanks for bearing with me for being part of my first like very difficult talk at the conference. {Applause}.

NEW SPEAKER: Any questions come to the microphone in the middle.

OLA SITARSKA: ... That's thanks in Welsh, I checked!

NEW SPEAKER: Thank you for a lovely talk and great work you are doing. I was quite excited about Django admin 2 when it first was announced. Do you know what's happening with that or - do you know what the status of Django admin 2 what's happening with that or who is involved or is it dead?

OLA SITARSKA: I don't think there are any concrete plans or anyone is working on writing admin at the moment. I think it's fixing bugs if someone sent me a batch and this is coming in the next few months hopefully but nothing more happening but if you want to try there is a lot of Django coders here to talk about this I guess.

FROM THE FLOOR: Hi Ola Sitarska, thanks for the talk, I have a question about the permissions and about the use case where you said you would like the Django girls organisers to see only their event and not all the others, so by modifying the get queries set method, what happens if I have, what happens if a bad guy or a girl has the url is he or she able to delete those other events which basically you don't see in the interface but they are still there.

OLA SITARSKA: Yes I checked that, I was wondering the same thing, change the ID and go to another object, I tried that and I couldn't get it. I guess it throws upper mission errors, I think Django admin checks the query set and checks if it is there. It takes care of it for you. You don't have to do anything else just that.

FROM THE FLOOR: Just level permissions, out of the box. Thanks.

FROM THE FLOOR: Thank you for the talk, VINCE a question regarding change view, how do you handle big selects or foreign case elects, for like a model which has a few thousand objects?

OLA SITARSKA: I had this problem in other projects there is a few of the apps basically, I think Django admin outer complete. It does it for you, they will give you a search, you can complete, autocomplete the right projects, the Django has ... field, if you put in the name of the field it will not show you the list of the objects but show a little search, so if you click on it, there will be a pop up with the list, with the list of the objects, you can search and then choose the one, input into the ID into the like input text field.

Thank you.

FROM THE FLOOR: Hi Ola, great talk, what is in your opinion the worst part of the admin, what would you like to see improved or removed?

OLA SITARSKA: That is a hard question. Well, I don't know, I really like the idea of the design now, I always struggled with that part doesn't look the best, that was kind of the thing I wanted and I am really happy to see it coming along and I kind of know, jealous I didn't think of it earlier of doing that. That is the one part, hmmm I wish Django admin was more, in the sense you can use the thing already implemented the filters and the huge part of code, superuseful use it in other places, basically in work, I wrote an app, for does similar things, filters and search and pagination, useful to have the stuff somewhere else in other parts of Django, I think that would be very nice to have.

FROM THE FLOOR: Thank you.

NEW SPEAKER: Thank you very much.

(APPLAUSE).

DANIELE PROCIDA: Can I just check that everyone can see the screen okay? No problems at the back there? Is the volume loud enough? Is the sound clear enough?

Yes, do we need it to be a little bit louder? A little bit louder, maybe just a tiny bit louder otherwise I think it is okay.

I hope you are not too warm or cold. Not that we can do anything about it anyway.

VINCE: I will introduce a colleague from the school of maths at Cardiff University. Trying to time what I say so it works in time. So yes, Dafydd is going the be talking about a tool that he is working on that uses Django. (APPLAUSE).

DAFYDD EVANS: Yes.

Okay, I have, Hi everyone, thanks Vince for the intro. I am a Lecturer at Cardiff University at the School of Maths, I am not a developer, I am a bodger with code, this is entitled the Cardiff Maths E-learning Project. This is one person at the moment. I am hoping to recruit some interested parties, it is okay a brief outline.

The motivation for what I am doing here, something about typesetting and then a description of the project itself and maybe some demos along the way.

Okay so writing mathematics requires specific lay outs and fonts, fonts is a huge problem in maths, held us back from splurging on to the web like other disciplines have managed to do.

The way we engage people via the web, lecturer produce PDF documents using LATEX we up load these, the students are adept at managing the things on the phones, it is a dead technology, lacks interactive capabilities can't really engage with the work if you like. In terms of homework submission, many of the other schools on-line submission it is difficult for us, the differentials all the various fonts so the way it is done, submit handwritten solutions and as teachers we mark them with a red pen, very old fashioned. Can LATEX documents used as the basis of some kind of e-learning system. Perhaps I will switch to, I will give you an example of, let me see if I can make it bigger.

Oh right okay. Okay, let me see if I can zoom in. Mathematical, so these are lecture notes I prepared for last year's -- oh, okay. Let's try something else.

Okay, so here are the lectures I prepared for the first years probability, 20 lectures, nice table of contents, automatically generated. This are the problems on the web. Mathematical fonts like these, how do you render these, not using ASCII perhaps UNICODE will save us in the end, it is nice but dead, lacks interactivity. There is a lot of knowledge, there is a lot of material, resources wrapped up in LATEX files, that need to be exploited and put on the web. That is my motivation in this.

Okay, so okay, a little bit about type setting then, we have got letters symbols, known as glyphs, groups of font type faces. Maths is brought into the modern world by Donald Knuth which he developed part of his tech system that was extended to LATEX in the 80's, now the de facto standard for introducing mathematical books.

, languages, when the author would take the handwritten to the printer, there is the content, various parts would be marked in a red pen, this is to be inset. On the right, familiar to us all, html document, various parts of the content to be marked in various ways on the left, the LATEX syntax, not well behaved.

But where LATEX excels is you will see there around this area here, is that it can typeset equations okay. So what does html have to do, well have to offer in that respect? Until recently, not a lot. But there has been a revolution in typefaces for maths for the web with the system called math jacks, if you can paste LATEX code into the web page it will display it beautifully, we might get a chance to have a demo of that at some point.

Okay, but, math jacks only concerns itself with fonts with displaying integrals and so forth. Doesn't concern itself with structure, because html does it very well, ordered lists, the headings you name it.

The LATEX files I have and many others in the school have, have a lot of structure, really, so really the basic idea, take a LATEX document, piles it so we can replace with structural commands and leave all the kind of rendering issues to math jacks and it appears that we can kind of do that.

So a bit more about LATEX here, we have got structural marker, structures, sections, we can have blocks, theorems. The first line there is for styling the text, so that is for having bold in text. Html has that built in, maths jacks doesn't bother with it, but on the second line, pie, and ... LATEX is rich in cross references and so forth, so you can label various blocks as you see around here.

Okay. Then you can then refer to that label later on in this way here. Something of course that the web is kind of made for right?

Okay, so very ostensible and customisable. This is a nonstandard environment that I have defined myself, to embed homework into the material, that pops up in the PDF, can this be populated on the web and can we build forms that get the students to interact with this on-line?

Okay, so, you might notice that at the top I have written my own class file which marks me out as a LATEX nerd that much is true!

Okay. So I wrote, so it demands a specific set of commands, which is why I wrote the class file. Implemented in Python, processed using regular expresses along with the stacks., ... expect for a context free language, ... the output is comes into a tree structure, like the object model for html.

I call it a materialised path tree, every node will record its path to the root. Okay, this is so we can get cross referencing working when it goes on to, to the browser. Okay, so numbering one of the benefits of LATEX is that it does the automatic numbering for you, if you have got two chapters and you want to change the order, cut and paste. Then run it through LATEX again, all the numbering redone for you, that is done, variables, for every chapter, spin out a new class and it has a number.

Any math node environments, as I said left for MathJax for it to handle. Level commands, book chapter, theorems, these are numbered so we can cross reference and talk about them as we would naturally in mathematics.

Math mode environments then, which we leave alone, important to know which ones we are going to leave alone, anything that is not designed in the list, the pasa will ignore.

A few things on the bottom here, that I have put in myself, to do with homework assignments which is kind of like the purpose of this project. Okay, so okay I built the tree then throw it out in XML, I cut it down slightly, what I wanted you to notice was, you see this mark up here? The parser left it alone for math jacks alone.

So, on to the Cardiff maths e-learning project. This is the, this is how it works as I see it. So the LATEX can produce PDF's, very well. If I run it through the document tree programme, really this programme makes LATEX respectable and well formed.

Okay, but, okay. I can throw it up as XML, style it, but how do you access the various bits and bring it alive? This is where Django came in, I have been meddling with it, probably meddling is not the right word, but experimenting with it for a couple of years in odd moments. I take the tree, I unravel the tree as it reaches the different nodes throws them into the Django database, does that through the models and the main model is the book node model. It is a book type, chapter, theorem. ... May have a title. May have a content node or image, then materialised path, unique ID within the database, parser up load lots of different books, book name or identifier and then spin off down the tree. Labels to handle and these are then converted into html links and so forth. Other classes that we need answers to questions the questions are determined in the LATEX source, the answer is what the user would put in, two foreign keys and a text field.

There is the project link.

So, how can we maintain this tree representation which is xml file gives us or the LATEX has inherently have within it, within the database, I found this wonderful, Django, mptt, ... it has the tree code algorithm, hierarchical data... ... we really want when we retrieve the block from the database, we have to be quick, rendering the fonts takes long anyway, we don't want to be delayed there. The material is, yes, I mean it will be edited perhaps but that is the kind of longer time scale really.

So the Django mptt app some of you may be aware of it, makes it easier to use the trees with the Django models, the model can be a as a sub class of the mptt model. The tree is updated automatically when you insert and move the model around. It is nothing I have really, I haven't really used that part of it yet. But the obvious instance methods you would expect from a tree based class so you get the ancestors, the children, the siblings, get the root.

Customs form fields, at the moment, the direction of the arrows, nothing gets written to the LATEX source at the moment. That, it maybe down the line but you know, some edits within the Django system maybe written back to the text database if you like, at the moment it doesn't do that, but it pulls it all through.

So there are no forms really at the moment that within the system that get to work on the LATEX data, the form is for the user interaction, that could change.

The best of all, is that it provides some really nice template tags, built rendering the trees.

Snippets now, in the models I have got the book node, that covers all node in the LATEX tree if you like. The first line is something you have got to put in according to the documentation, I have put it in to link it up with the other tree nodes. So we have a in type, chapter, number, title, the materialised path as well which is useful for linking between the book nodes.

In the views then, so for example, get a particular chapter into the context, by whichever means necessary, I tend to do it on primary keys, my url's are not beautiful, but something to work on.

Then in the same context, attach the sub tree, so you get the descendants of that node in the tree and it includes the self. If you want to display a chapter then all the information below that chapter, so might be some subsection and figures all are included in the sub tree. When you get to the template, load the tags, then there is recurse tree. What my templates do, switches on node type, have got a node class as well, to avoid a long switch statement. Node class is the levels and so forth.

Okay. So, if for example the node type is a chapter then you just throw out a div that says what it is. It will put a heading there and it will pull the number and the title out of the node and then fantastically then call it built in tag their children and it will call the template recursively on the children, so automatically then will pop out the theorem if it is there, or the next definition, this is where the power of Django comes to help me in this way.

Okay, so I have defined a couple of LATEX environments for interaction inspired or ripped off the exam class by Phil at MIT so two types of answers at the moment, two types of way that the student can interact with the system, so the first is the homework environment. This is for the lecturer then, this is essentially a list of environment, begin and end questions here. The first question, then the lecturer can optionally choose to include a solution, a sample answer of some description, it may be and I would hope to progress this a little further than I have done by today, that you know, you could use kind of regular expression to, some kind of metric between the string, some kind of string similarity metric to see if the student free text, when I say free text. That would be ... a score, that might be a little overambitious. So then there this is this test, the Mctest, nothing to do with every time you go McDonalds, it is a test., here is the first question, which is the odd one out? This is again inspired by the exam LATEX class, so here are the four choices, you can mark up the correct answer there in the LATEX itself. Okay?

That is kind of it really. There is something there on the schemer, so the blue nodes on the right are the one that is come out with the LATEX, that part of the database, materialised of course and then the module then this is really just a course, what we call our courses, so a course could have many books, questions come out in the books, books come out in the home work, student can follow the module and so forth.

Let's go for a demo, let's go live. One minute? Yikes! OK well let me show you this any way. So, this is the tree in the Django admin. What we can see here is that the latex has all been chopped very nicely and you can see some of the text based formatting has been replaced with HTML but big chunks of mathematics has been left alone. Find some so there is some mark up here and so on. So I probably can't give you a demo of the interactivity, but this is me here, so this is the course I taught last year, lecture notes, and OK so if I now click on one of these chapters this is where the recursion really helps OK so this will recurs down all child notes of the tree and the fonts are left to match and it's really rather good. I don't know how MathJax does it but it really does a fantastic job and it's all from the latex file no editing done on the server, you can show high boxes and so forth depending on some switches you could not display the proofs, you could not display the solutions and so forth. Let's see if I can get away with doing one. Here is the multiple choice, maybe I won't go there but because I am logged in it will allow me to submit an answer to here now so I can start the answer and OK this is where the student has to come in and start writing so here is one I prepared earlier on the clip board there it is. Then I can save this and it will pop up a preview in this window here and I can keep editing and so forth and go back. And then when it's done, I can submit, and then it locks it. OK. And the idea is it's to get students to do this before a lecture and another part I'm going to be working over the summer is to take all of the submissions then which you can see now in the admin and actually funnily enough turn them into PDF to take them into the lecture theatre in standard format to do peer assessment and peer marking. I could go on but don't think I should. {Applause}.

VINCE: If anyone has any questions if they could make their way to the microphone. We have time for a few questions so ...

DAFYDD EVANS: I'll put up my final slide but do go on.

NEW SPEAKER: The first thing to say if you are a bodger, I don't know what the rest of us are because that's very impressive. As a Django nought with a maths degree this is absolutely fascinating to me. The question I suppose is and you touched on it at the end are you looking at doing this stuff that goes the other way. At Oxford we didn't get the joys of our lecturers providing us with all of our lecture notes because they're mean like that so I embarked on writing one of my lecture courses by hand in latex and decided it took 7 times longer than doing it on paper did I decided that's because I didn't understand latex and didn't have something that could go the other way so to build my latex book and get something PDF would be amazing.

DAFYDD EVANS: Going the other way. The way I get HTML is throwing out tax based on element way, fairly easy to go back to latex. Unfortunately the academics the older academics are always going to use chalk and talk right, the middle aged academics I mean like myself you know. I am happy using latex and text. We like text right OK so I would like the definitive source to be the latex files and it doesn't really check whether latex well formed. It assumes that the source code has been passed through latex and been found to be OK before turning them into, before chopping it up and putting it -

NEW SPEAKER: Has interest for peer reviewing stuff and you could go on and edit make amendments and submit bug fixes to the lecture notes without down loading entire latex document. That would be awesome.

DAFYDD EVANS: That's the idea.

NEW SPEAKER: Math ML when I was going through when tea years ago that's where the future was going to be. I'm interested it hasn't gone anywhere. Where is it?

DAFYDD EVANS: If you see MC squared written out in math ML you can't read it as math mathematics but if you have experience of latex you can read the maths through the mark - you could put this stuff out as math ML but it wouldn't be human readable that's the problem with that. Good question. As I say a lot of work to do on the Django side, a lot of work to do on Python programmer development side. Tried to load up to production service this morning and everything fell apart. Any help, anyone who interested I need your help. {Applause}.

DANIELE PROCIDA: OK. Before lunch, we have a very special guest of this Djangocon. It's the Cardiff writer and poet Peter Finch who is one of Cardiff's most celebrated poets. He's also a bit of an adventurer who knows how to go on adventures just by walking out of his front door and through the streets of his own town and you may have seen that we have tickets for some city walks in the evening after the conference, Monday, Tuesday and Wednesday. Peter will lead you on a walking tour of Cardiff starting from here. You will discover some things about Cardiff that Cardiff doesn't want you to know and some things it's tried to hide and some things it's tried to invent. So he'll peel back some of the veils that Cardiff has laid on itself.

By the way, if you are going to the clink on Tuesday or Wednesday night, don't go on the tour that night because it will take too long for you to get to the clink so change your tour ticket.

Any way that's enough from me. Please welcome Peter Finch. {Applause}.

PETER FINCH: Of course I might have got this wrong I was under the impression I was coming to a serious academic conference {laughter} and I have been working on some translations from the middle Welsh, I can see some of you on the right hand side there going to sleep already ... these translations have taken me a long time and I thought that none the less I think I should share at least one of them with you. This is a translation of part of the Mabinogi which is the great set of partially mythological tales that come from Wales and this is the blodewydd(?) with section rendered as I have done.

I was boom ... I have a good scree...diminuendo kick ... fut... I am star up and star ... my modernness appalling, nastem, nastem, fob on me ... Stirling, I list em, I list em, why? I was bleened, bleened, doged ... goths, goths, ooh, I am store room ... star light, star light star light, star light, star light, star light, me, God, gaths, stir, of star light, why? I ... splod... oak... broom ... meadowsweet ...

Because those 3 or 4 were what the lady of the flowers was made of.

Any way enough of this. What I'm really here for {applause} - is to tell you just a little bit about Cardiff the place that you're all sitting in.

Cardiff was pretty much nothing when the Romans came here built a fort where the castle currently stands about 2000 years ago and people have said to me when I address groups on the history of Cardiff, well can't you make it a bit quicker please? Can't you speed up the history? And here I am starting 2000 years ago. So I said OK Welsh, Romans, Welsh, Normans, Welsh, Bute, David Cameron and the Welsh again {laughter} {applause}.

We're a population of 2000 people, that was all a thousand years ago when the Normans came here. We did a few things after that. We burnt a martyr, rawlings white in St John's square, we - he had the wrong religion so they set fire to him.

Then suddenly industrial revolution 250 odd years ago arrived and it was that single event that made the city. If it was not for the coal in the hinterland of Cardiff and the iron being manufactured being dug up and subsequently being exported through the port that Cardiff became, we'd still be pretty much a market town or a small fishing village. The population during the last 200 years boomed boomed and boomed again to get us to the size of place that we are now which is about 340,000 people. Not that large E number and I've been asked to come up with some phrase as tourist brochures often do about Cardiff and I've said, well, it's a city that is not big enough to get lost in, but it's about big enough to have an affair and get away with it. {Laughter} and a woman at the back put her hand up and she set: oh no it's not! {Laughter}.

So, I am leading a walk, as you've heard, we're not going to take in on this walk the night life that Cardiff has apparently become famous for, trains full of people dressed in costume arrive here from places as far away from London, Hull and Portsmouth to spend their weekends partying. We've got something like 1300 places where you can buy alcohol which is punching slightly above its weight I would say. So Cardiff is a mixture of that, this having a good time city, and of course it's a sports city. We've got the stadium, we've got the cricket stadium. We've got the Cardiff city stadium. And these things together with the preponderance of galleries and places where you can get cultural entertainment means that the city, the youngest capital of Wales punches well above its weight.

I thought I'd give you a couple of poems to try and put this together in entertaining form. I looked at those Sunday supplements that list things that are in and things at are out. Things that are fashionable this week. Things that are not fashionable this week. And I thought I would compile a list for the city. And here it is. I called it mewn mass, mewn meaning in and mass meaning out. Mass comes from Maes. It means field in Welsh. Originally if you went out of your cottage you were in a field so you were out. We've got rid of the fields, this is what is in, in this city.

Coke, legals, shirt tale boy, bay, whooper, coryum, jap and mall, sleepers, ice ball, 5 a.m. suit and arm band, massage fat arse, plain crap, train, Tom Jones, Ewan, Bala, Cerys, anything dragon, blue birds, flag, café, money, mermaid, reach flying throughout, Hemmingway Heaven, bosons avenue, Atlantic walk, leisure village, glass waving slates, stacked, the Russell Goodway memorial round about, cycle way, speed stripe, pile of bricks, euro time, multi-screen disposable glory, future proof young assembling Assembly, absolutely and then mass ... out what we're not ... any longer ... Victoria park, coal and steel, mild, pale, Harlech, soul fire, shaving, Dunleavy, night philanderer, Jim Callaghan corporation, west of England, west light, vic, ship matie, 10.30 AM Spanish club, afternoon, valley day, Portland stone, metal street, black mortar, Cath vale, Tom Jones, Shirl, walking, crock, up town, Caribbean, red light bike, over time, Christian marching, salvaged beam engines, pink Floyd, flood trolley, Estonian Polski coffee bar, Italiano, fog, welcome to Cardiff general, that's the pearl building, do not walk in the gutter... sorry...

Now the astute of you will notice Tom Jones managed to make it in {applause} ... now since we are in conference, in a sense, how many people here have sat on a Committee? Can we have some hands up if you have sat on a committee, give me an idea that's about half of you just under half of you. Committees have a particular thing in Wales. Anyone here who comes from Wales who knows anything about this county will know that absolutely everything here is decided by a Committee by a gathering together of people who will gather and form a Sub-Committee who will then form advisory Committees to arrive at a solution perhaps maybe and this is how it goes.

It is morning. At the meeting I can't seem to make my mouth work. No one else is affected. They all talk in fervent streams. The meeting is so engaging that by item 3 I'm asleep with my eyes open but no one can tell. Years ago an arts management brainstorm boredom drove me to assembling plastic kits on the table edge, plastic glue stuck the agenda to my sleeve and being the arts no one mentioned it. Creativity, but this is real business, different, I guess.

And item 55, someone proposes we should agree and everyone's hand go up for something to do I put mine up too. I have run out of paper clips to unwind. The man next to me clacks his briefcase catch, says, the train, and goes! I could have done that! Now can't! By item 20 my agenda margin is a Maori tattoo and I have my pen standing like a tree on my note pad's edge in the dark far corner the accountant is in Maya meditation or dead, there are ravens calling in the Chairman's voice and sea sponges drifting in my head.

At item 30 I am stretched out to avoid back contact with my chair. There are 506 grain marks on the table. They pass me a sheet which asks my expenses. Asks my expenses? It has all led to this. You can smell the sweat as me stand hauling suit jackets over crushed shirts, the sky, the dark sky will gleam in our head lamps as we chase it home. We have made great progress. My mouth says this. How the hell does it know? Oh yes {laughter}.

{Applause}

I will finish we will get us to lunch about the minutes, about two minutes early, about a poem about the time I moved into the private sector and I had to attend my first business conference which was quite a shock to the system. Went a bit like this, this is building business.

At the moral build conference for middle managers, at a hotel so big the atrium has clouds. We sit at the back in our suits with our badges saying: My name is Peter. Our bottles of Malvern Water....

The overhead on the platform says "love the customer" we are looking but everyone here is staff. The Chairman has an associate taking the name of the people at the back. The overhead reads you only succeed by being in front. The Chairman is saying change is a rocket, innovation is the way and win or die this man has a way with words.

The guy next to me, my rival at copiers has a list on his lap headed "things I want to shove up the Chairman’s arse!" it goes, umbrella handle tooth paste holder, two wars of soap, frozen pigs tail, cattle horn, flashlight, snuff box, wire spring, salami, mortal, peas, vegetarian burger, Ken wood kitchen mate, 400 an two stones toner bottle -- I ask him why the toner? (LAUGHTER) he make that he says!

Signing my name at the bottom of the list and handing it to the collector of topics for that afternoon's open session.

The Chairman is leaning forward uncharacteristically ranting "if we don't hit targets soon all our arses are going the be on the line" he says, sly out to reception and ask if they have any spare toner?

(LAUGHTER).

Best to be prepared!

You can buy these books outside Diolch yn Fawr.

(APPLAUSE).

DANIELE PROCIDA: Thank you very much for that Peter! [laughing] that was wonderful, do take a chance to go on the walks, pick up a ticket they are free on the tickets on the tickets page, can only take about 30 people at a time, which is why he is on three times.

There is a book stall, they have books, the publishers are giving us the Djangocon organisers 35% of their taking, so you will be taking a lovely souvenir or a gift to take home and contributing something towards us. So it is lunchtime now and we will back here to start again with the next session at 2:15 thank you to all our speakers this morning, see you after lunch.

(APPLAUSE).

(LUNCH)

DANIELE PROCIDA: I have one small apology for lunch, tomorrow we will make sure that the food is better labelled for vegetarians and people with other dietary requirements, the other thing I didn't realise that we did have meals prepared for people who for example were vegan, you needed to say that you were, in order to have it brought from the kitchen, I didn't realise until most of the way through lunch. I hope you got lunch if that affected you., we will have that sorted out for tomorrow.

Next up we have got something special, so I wonder if anybody knows what, what that is? [on screen], no it is not a hash tag. And it is not a badly unformatted comment.

It actually one of the Django communities most valuable resources and it is the one that as a new and still as a Django programmer, I turn to time and time again for help and assistance and to be got out of hole that is I have dug myself into. It is our official IRC channel. It is staffed night and day around the clock, every single day of the year, by an army of volunteers all around the world covering all the time zones. We are going to recreate it or re-enact it, we are going the enact it live on stage here, with three of its most dedicated members, doismellburning, apollo13 and MarkusH. You can't imagine my surprise when I learned they had different names. Florion who is a member of the core team, Christian and ... also a member of the core team, they are going the be in the IRC channel and you are also going the be in the IRC channel, just like in an IRC channel you can speak and ask your question, contradict the advice, ignore the advice (LAUGHTER). Ask the same question in a different way. Come out with your outlandish opinions and as longer as you have got a microphone you are welcome to put something to them and you will see how it works when you are on-line because, after this, it will be really nice if some people who previously had not experienced or ventured into our IRC channel went to have a look perhaps to help out, perhaps to get some help and contributed to the liveliness and helpful part of the community. Thank you to all of them.

(APPLAUSE).

And. Yes? Okay. Are you on this one? Okay. So I will get this opened up. It is a Mac, you can manage that, Geraint grab the two microphones. If anybody is ready to ask our friends here a question, put your hand up and Geraint is going to pass you a microphone. Let me get this mirroring. Oh. How are we? So. Okay. So that is, let's close some things, new window, (LAUGHTER).

I will need the lap top after, we will leave you in their hands off you go.

FROM THE FLOOR: Hello? Yes. So why doesn't my cycle tag work, work in my jinja template.

NEW SPEAKER: That is how you say it.

(APPLAUSE).

You know, you have to write in the way we understand you, for this to work. In an Australian accent.

Who uses jinja, the thing it doesn't work.

RUSSELL KEITH-MAGEE: But I need the Django one.

NEW SPEAKER: Why?

RUSSELL KEITH-MAGEE: There is this page I have got, trying to use.

NEW SPEAKER: So, let's go back, what is your actual problem?

RUSSELL KEITH-MAGEE: Well I have got, this is being credited as me by the way!

I have got a project where I was told that Django's template language is too slow, I need to know my templates to be fast, I can't use the cycle tag that is documented here.

NEW SPEAKER: The problem is not probably the template, probably somewhere else in the code.

NEW SPEAKER: People turn up, they have got an idea, they have got performance problems, working it out is hard, the best thing is work out where the problems are, you might have heard that jinja2 is faster, I don't know, but the first thing to do, find out where your problem, if that is your problem, I will amazed impressed. But the Django templating language is good and I am surprised if that is where your problem is.

DANIELE PROCIDA: Actually I heard that the Django template language, the template engine sucks is that true? (LAUGHTER).

NEW SPEAKER: That is not why Django sucks talk here? Right?

DANIELE PROCIDA: Why do we need another template language?

NEW SPEAKER: One way it was a way to clean up the API for the Django template, now we have the support from the languages, even so if we don't use it, we still have nice clean API which someone else might be able to use.

So for instance, if the Django template language is too slow for you, whatever.

NEW SPEAKER: Furthermore the project that already used jinja and haven't used Django, they can move to Django,.

NEW SPEAKER: The jinja support is brilliant, if you have got them to use them, if you are starting something from scratch, use the Django. ... Disagree with me, tell me I am wrong, I want to learn here.

NEW SPEAKER: Hello.

FROM THE FLOOR: Hello, I have been used (INAUDIBLE) recently and I spent all morning trying to figure out what is going on with the permissions, I found a, basically about five years ago and it is people talking still talking about backwards compatibilities, so why is that not been solved already?

NEW SPEAKER: So I didn't get the first thing, what have you been using?

NEW SPEAKER: The models,.

FROM THE FLOOR: It doesn't create the permissions, generate the permissions of the ...

NEW SPEAKER: Does anyone know about proxy models?

NEW SPEAKER: Well so we know whom to blame right?

NEW SPEAKER: Can you help us here, you know about proxy models.

RUSSELL KEITH-MAGEE: I know that Malcolm knew them a lot more than I did, that was his grand idea. I don't know why they are not working so.

But...

NEW SPEAKER: Can I ask a question?

NEW SPEAKER: Of course, what is your problem? (APPLAUSE).

FROM THE FLOOR: If there is an issue with.

NEW SPEAKER: Permissions of the proxy model when you create the proxy model, it doesn't generate the permissions and Django admin is confused because it doesn't create the permission for it. So there is a comment on a guest, on ... that solves that. But I really don't know why it is not fixed. I just wondered why for five years?

NEW SPEAKER: Well it is, they are not many people using proxy models I guess, who is using proxy models in here, quite a few, you never ran into this issue?

Well, if there is management command already, the next thing to write up a patch, submit some tests and a request because I have never heard of this issue, I personally don't use proxy models and I probably be, don't need permissions for them. As long as nobody picks the code up it will rot us forever, poke us with a stick in the IRC channels.

You could also go hash tag ... to get advice if you want to fix the bug yourself.

NEW SPEAKER: I was thinking about that, I think they are very useful anyway.

FROM THE FLOOR: I have a question about the admin.

NEW SPEAKER: Oh, Ola?

FROM THE FLOOR: My project works in development but my server says 400 bad requests.

NEW SPEAKER: You are allowed hosts so, release notes for Django.

NEW SPEAKER: 1411?

NEW SPEAKER: I think so.

FROM THE FLOOR: What is the ...

NEW SPEAKER: Standard Django set up, you have got some sort of proxy web, Apache Amazon load blaster, that is what you have got? So, that will be the host names are used for a variety of things, the host that has come through, so to construct like 4 urls sent out for password reset e-mails and if you miss configure your fronting proxy you can allow anybody to set the host header to whatever they like, send out password reset details, go into their domain. Django added in -- yes? 144, the documentation explains it better than I tried to garble. Allowed hosts, new setting, keep you safe, safe by default. You have to set that in production with the host name of your site so that you don't get hit by that problem.

FROM THE FLOOR: Okay, I will try it.

NEW SPEAKER: Release notes you read them when you upgraded right?

NEW SPEAKER: Seriously, the release notes are incredibly well put together, hands off to the team who put them together, they have got everything you need for the update.

FROM THE FLOOR: I am trying to use the admin to ... I can do the edits if I want to expose the admin because I want people to be able to filter and everything but it doesn't work?

NEW SPEAKER: The admin is not the user interface please.

FROM THE FLOOR: Why?

NEW SPEAKER: Ola covered this in her talk.

NEW SPEAKER: It is possible that the user gets access to some of the data you don't want them to have.

FROM THE FLOOR: Why is the admin there if it is not secure?

NEW SPEAKER: Also give your users to dead it the user information, they can check, I am superuser and save?

FROM THE FLOOR: Ah, that doesn't sound good.

NEW SPEAKER: The admin is intended for those administrators, those managers of certain components and they can use it for certain things and it is pretty great which for what it provides, so if you have some website that is way more user facing you probably want to implement all those user facing designs, all the user facing forms yourself.

FROM THE FLOOR: Thank you.

FROM THE FLOOR: Quoting live from hash Django, is there a way to get the list of classes ... where the (INAUDIBLE) at the moment.

NEW SPEAKER: What?

FROM THE FLOOR: Is there a way for me to get a list of all the classes in my app slash models.py, where ...

NEW SPEAKER: You are basically talking about registry work and get models from app? Or get models?

FROM THE FLOOR: I just want the models?

NEW SPEAKER: Why?

FROM THE FLOOR: He hasn't told me why.

NEW SPEAKER: Ask him why, that is one of the things, sure you can use the iterate over the items and check if the current module is the -- why do you want that? You have the API for that.

NEW SPEAKER: It is the Django.apps and from there you can get all the models that are available in your project.

FROM THE FLOOR: Okay cool, I will try that out.

NEW SPEAKER: It sounds like Florion is sounding flippant when he asks why, that is the best advice, ask yourself why you are doing something, take a step back, are you trying to find a solution to the problem you have developed while trying to solve your actual problem?

FROM THE FLOOR: Hello, I want to build a Multilanguage website with Django, how can I start multilingual designs and models?

NEW SPEAKER: I am lucky to not have that problem! (LAUGHTER).

NEW SPEAKER: Um, Marcus?

NEW SPEAKER: Django models translation exists, I don't use them or know anything about them. I will not use them, why should I not use them?

MARKUS-H: You want to make your website international or the content of your models of the database?

FROM THE FLOOR: Basically I have some projects in the database and I want to have each project in each international website but I want to be able to switch between French and English for instance.

MARKUS-H: On the interface level, do you want to translate?

FROM THE FLOOR: Also in the database, I want to be able to add products and translate the description in different languages.

MARKUS-H: So they are, Florion mentioned, the Django model translation does some not so nice stuff regarding the database level because, if you activate a new language, it is modifies the database, new column and depending on the content of the site it may take, you don't want that, at that time it is probably ... looking action, the site is down at that time. You want to have something more advanced like, the old thing is called Django ... I have no idea. A couple of years ago, there is also this thing from (NAME - INAUDIBLE) something.

NEW SPEAKER: This is what I like about helping in hash tag Django, I learn things every day, I help people get into Django and get them started but still even up here I am learning things now which I really enjoy.

NEW SPEAKER: The easiest thing to move to Germany, they usually don't translate anything, you should be right.

NEW SPEAKER: German site is translated I disagree there.

FROM THE FLOOR: There is a project called Django parolour which does that.

I had a small question, my house is on fire and I think since my server started burning my website doesn't respond, is there a Django app that can have Django fireproof or something.

NEW SPEAKER: No just call the emergency services!

FROM THE FLOOR: I have, I am over here on your right, I have a whole load of Django 1.4 apps that I want to upgrade to 1.8.

NEW SPEAKER: Do the steps that, the release notes,.

FROM THE FLOOR: Is that going from 1.4 to 1.8, do I need to read all the notes in between?

NEW SPEAKER: Do them step by step, unless you know what is happening in all the releases, the general recommendation I get for that update one by one, step by step, safer.

FROM THE FLOOR: Is there major release like summary release notes?

NEW SPEAKER: What would it give you, from 1.4 to 1.8?

FROM THE FLOOR: Like the 1.5 release notes is there a summary of all of the things.

NEW SPEAKER: Important things and -- there is no summary from 1.4 to 1.8 for the simple reason it would be the compilation of the existing notes, we don't change that fast, something you have to remove in 1.6 ...

NEW SPEAKER: (INAUDIBLE).

MARKUS-H: There shouldn't be any changes in the ... so 5.1 and two that make you change your API, 1.4 allowed hosting, it was a security release, but if you look through the major release notes, 1.5 and six, you are probably safe for the first bit. If there is something you have to fix in your apps, but generally, see 1.6, 1.5 release notes.

NEW SPEAKER: (INAUDIBLE).

FROM THE FLOOR: Why was 1.4 called LTS, if we can switch from one to the other?

MARKUS-H: Not about switching directly from one, doing lines of code and then being on 1.8, but it is the possibility of giving companies in general the opportunity to stick with the version for three years, where they can run this, the project on this version without having to fear that they don't receive any security updates because if the projects grow, the upgrading tends to get more expensive and depending on the company financials it might be a burden, quite a high burden so they wouldn't choose Django if it is not, if they are no releases.

FROM THE FLOOR: Still have three years I want to, after three years I want to upgrade, I have do to do all the upgrades.

NEW SPEAKER: Have you tried upgrading (NAME - INAUDIBLE) versions.

FROM THE FLOOR: Yes and I switched to Mac so (LAUGHTER).

FROM THE FLOOR: I tried the apps thing you get, I get this weird error ... apps are not ready yet.

NEW SPEAKER: Django.set up. A new entry point for 1.7, 1.8. 1.7. Perfect.

NEW SPEAKER: I found a really interesting security vulnerability in -

NEW SPEAKER: Stop - stop -

NEW SPEAKER: And I publish on my web blog so you can read about it.

NEW SPEAKER: That's probably the moment we already kicked you out the Django you not post the link.

NEW SPEAKER: Why is that?

NEW SPEAKER: We have a security process and we don't want to leak any information about security issues to the general public to give us time to fix them. This obviously changes if it's already public but, well depending on the site, we probably still try to get it on for another 4 or 5 hours to actually repair fix but in general there is Django repair.com slash security I hope and if not someone create it quick and you can see a process there for reporting security issues and that's the only way and don't blog about it, don't tell someone that you've found something aside from us -

NEW SPEAKER: For security Django project.com if you found some security problem or issue.

NEW SPEAKER: So I am maintaining third party library and supporting 2 versions of Django at once and we're shipping migrations and now Django is starting to ship migrations and now I wonder what I should do because I am really confused if I merge migration S with user 1.7 and 1.8 and I am missing migrations if I do it the other way round I am relying on things that are not there yet how do I fix this.

NEW SPEAKER: You are using ...

NEW SPEAKER: No I am talking about supporting Django 1.7 and 1.8.

DOISMELLBURNING: In usual you should use the lowest supported Django creations {inaudible} then use upgrade and at this point we generate migrations with oldest supported Django version so migrations are basically for 1.7, if you create them in 1.7 they should work aside from backs yes but 1.7 migrations have been experimental feature in that state that we can ship stable migrations in 1.8 so in 1.7 you enhance what's currently I agree.

NEW SPEAKER: But when I ship my - I create migrations with 1.7 then you change the user model I think, user email field for instance or the email field max length changed if I in my app have an email field and somebody is using my app and tries migrations it changes - {inaudible}.

NEW SPEAKER: You should set the length to whatever it was before so this will practically override Django's now default for email field and your app won't have any migrations in that regards.

NEW SPEAKER: OK.

DOISMELLBURNING: It's a little bit annoying but the thing with migrations is you have to be really, really careful so I wouldn't for instance use any default methods or be really careful because you can't delete them anymore because the migrations file will still reference them so at some point if you decide you don't need this any more you still have to move the meta into the migrations files to keep the old migrations working so in that sense use all the tricks you can to generate your migrations and make sure that they work even if you change your model files or if we change the model files.

NEW SPEAKER: I've got a bug I've got a bug. I am building a neven(?) app and when I use the create form I get that time is not nigh - what does that mean?

DOISMELLBURNING: What do you get?

NEW SPEAKER: I am filling up a form and there is that time filled in it and when I enter the value it says that that is not nigh(?) I don't know what it means.

NEW SPEAKER: Time zones.

NEW SPEAKER: Oh the time is not nigh ...

DOISMELLBURNING: Where is he hiding?

MARKUSH: Over there behind the letter.

DOISMELLBURNING: Do you get an actual trace back?

NEW SPEAKER: Can you place the trace back please? {Laughter}.

MARKUSH: {Inaudible} we don't have advertisements.

NEW SPEAKER: I will check the documentation thanks.

NEW SPEAKER: Hi, I am here - right so let's assume I have a web-site which will be published only in one language and that's the default one, let's say Polish. So I have to enable the internalisation to have the admin in a Polish language. And I know that I won't be translating my web-site to any other language. Should I keep wrapping all my strings with get text function? Does it have any like - does it add any layer that could slow down my web-site even though I'm not using translation because the strings are actually written in Polish?

MARKUSH: If you know the web-site will only ever be used in Polish I think it's pretty perfectly fine to just use Polish things and not wrap them in you get text. I personally tend to always strap everything in you get text and if I ever going to have to up-date it, it will save me a lot of time.

NEW SPEAKER: All right but -

DOISMELLBURNING: It will slow down yes but the question you have to profile how much it slows down for you really because for a few strings between the latency of the HP connection and the {inaudible} you have a 4,000 meta {inaudible} more who cares aside may be Google.

NEW SPEAKER: Thank you.

NEW SPEAKER: I've got a web-site that's trying to use 2 apps and there is a name conflict. They're in 1.7 and I know there is something in 1.8 to help with that. Is there a way without {inaudible} both {inaudible} fixing it getting them both to 1.8 to get over the name conflict issue?

DOISMELLBURNING: So if I understood you directly they have the same Python in both package name?

NEW SPEAKER: Yeah.

DOISMELLBURNING: OK this kind of works in 1.8 - yes 1.7 too. The thing still is depending on how other application reference the models in those apps even if you re-write application name of one of those applications you have to adjust insert library accessing disk so in general I would suggest to avoid it. If somehow possible use name spacing in the sense you put project name underscore something like that. So I tried it once and gave up. If you have better luck with changing it, tell me.

MARKUSH: Also running problems with migrations if they depend on that app because they're out coded with the name at the time the migrations are made.

NEW SPEAKER: What I actually do I went through and renamed every reference to one of the apps which took a while but ended up fixing everything.

APOLLO13: Yes I'm not sure anyone is coming up with any solutions. Good luck. Thanks.

NEW SPEAKER: I've got any reason that you should use get text or end get text which is if you are writing strings in a language which has plurals different to English or more than one plural forms then it's really handy to be able to use end get text and trust your my creation files to give the right plural form rather than having a lad of python that puts strings together. Some languages zero is a different plural form to 1 or more than 1 and that can be really nasty.

APOLLO13: That makes a lot of sense.

NEW SPEAKER: We're using Django 1.7 and over the time we built up a lot of migrations and I wonder if that could cause our tests running slowly because most time it's running much slower than it used to be.

MARKUSH: Yes because it was the first orientation and wanted to get it out to be useable by the general public. They are not great in 1.7. They do their job. 1.8 solved a couple of problems now. We have couple of things we want fixed as well. Bunch of improvements and personally I found my {inaudible} to do this. Yeah, it's not solved but it's intentionally not solved.

DOISMELLBURNING: And there is some discussion going around for making migrations faster in the sense you may be even at some point be able to skip migrations for tests completely but we still need to hash out the details so help and implementation is very welcome in that regard. Yes, we've got someone to fix that for us - you're staying here for the sprints right, Andrew?

NEW SPEAKER: Long flight, right? ...

DANIELE PROCIDA: OK, thank you. {Applause}.

This goes on 24 hours a day, it goes on day and night, sometimes these gentlemen sleep and other people come and take their place on the 24 hour Django help desk but basically whatever your question if you can formulate it comprehensively and express it politely, somebody is bound to at least have a go at trying to help you on the Django IRC channel. So if you have not ventured in there, there are people as helpful and generous with their time and effort as Florien, Christian and Marcus who will help you so thank you very much on behalf of the entire Django community for your work on the IRC channel. Really appreciated. {Applause}.

We'll set up now for Lucie. Very pleased to welcome Lucie Daeye, she is a social scientist who picked up Django just last year and now she is already using Django in her doctoral studies. Lucie. {Applause}.

LUCIE DAEYE: So hello everyone. My name is Lucie or you can find me on twitter at patjuek, it is a word in Korean, it is one of my favourite dish, that's why.

First I want to thank Djangocon Europe for letting me do my first talk ever at a conference so right now I am shaking at hell and trying not to run away.

I want also to thank Django FR and my centre for {inaudible} for paying most of my travel expense this week, if they'd not done that I couldn't be here doing this talk.

I organised Django Girls in Paris last April and I also try to launch py ladies in Paris but now it's a work in progress: I am PhD student in social science but I'm ore geographer and doing research in Korean studies currently I don't have any degree in computer sciences and I started to use Django and python one year ago. So why am I here? I'm here to talk about Django and social sciences.

For my research I have to create, manage and keep quantitative and qualitative data on a 100 restaurant it can be addresses for numbers, name, GPS co-ordinates but also transcripts and text from interviews I'm doing with restaurant owner. I am third year PhD so I am baby researcher and I reply on other students and advisor for guidance on how to do science or basic stuff like what software to use.

So, when I started my research I got a bit overwhelmed by my data. I didn't know how to keep it organised on my computer. First I blame myself. I'm so disorganized in real life so it a must be why my files are everywhere on my computers and in so many folders then I went to my advisor and friend to seek help. Social scientists have been doing science for so many years I supposed they would know what software I would need. What kind of software was I looking for? Software that could transform and transform info collected quickly on field into usable data and centralise all kind of data and access it easily. Write and cut transcripts put labels on some parts to be easy to find them after. And you can see that my demands were not - were only simple. And I had also another list in mind what I called a nice surprise. And also the surprise I was hoping to find in the software found that maybe it was more adventure.

So with my programming knowledge I already knew some boring text could be automated. So I was secretly hoping that my dream software would use Google or open street map to find and import addresses GPS and extra I was hoping to be able to write comments on my data, stuff like where I found my document, or my general filling after an interview.

I was also hoping to do simple maths.

So, with the both lists in minds I went to my advisor and I ask her the name of the beautiful software I must use to make my life easier. Her answer was use spread sheets. That was not the answer I was expecting. So, I was already doing that and it was a mess.

So, if I'm here telling you the story that was me at the time {applause}!

So, this made me wonder why she was managing her data like that. I thought that maybe she had only transposed the old way to do research, something like putting all kind of pieces of paper in multiple red folder and put all this in an organised shelf. So, I wondered and I check with my other PhD students my colleague. They knew around computers like me so I suppose they were more aware what kind of software was available at the time. Sadly they were acting the same way as me doing what I called the new old way. It's the same but individual one.

Something was different from my advisor. My friends they were complaining they were losing data all the time because they were as me not organised at all.

So, I was back to square one. My data was still an ugly mess and there was no mechanism to save me. I went back to my field research but something convinced me to do something. So I was accepted on a research group from the university to do a big street survey, ask people if they have 5 minutes and you would give them 30 minutes. And we use the paper survey with both open and closed question. We were 10 people working in different district and in different language. In the end we had around 800 survey to digitise, the plan of course to put all data in spread sheets.

But that's not the shocking part. The shocking part was they used 4,000 euro of budget on 30,000 euro budget on one person manually input every piece of data on the survey. Most of it, it took this person one month to do it why?

Because this person had to find manually the address of every restaurant or grocery store mentioned in every survey. Can you imagine what it look like to do it manually? Let me explain it for you. On every place you have to do request on search engine then you check if address is correct, then you copy paste it in the goods sell of the spread sheet and you do that again and again and again. So I think it has been exhausting for this person and most of it error prone.

All of his was complete nonsense to me. I was shocked that something so basic could cost that amount of money. As you may know we {inaudible} money in our field so it's a complete waste to spend it like that.

I went back home and did the same things I do every time something like that happened. I complain do friend who is also a developer. I told him that it was not acceptable and that I had to find another way to do it. He told me about Django.

So, at the beginning I was not supposed to learn how to code. I am a PhD student and I'm supposed to have no free time. More over learning something that everyone in my life will see as not {inaudible} to obtain PhD in social science will be seen as a waste of time or a way to procrastinate from your real objective so with this in mind I chose only to use Django admin. For me it was like a virtual directory. I had a list of restaurants so it's here. So I have this list and I could click in one of any of this and have this, it is a form with all the info I collected on the restaurant. So for me it was easy to use and maintain it was not a big table with everything put on it sadly, it only worked for a time because I wanted more features. It's just the admin. But what I was missing most was the maps.

There are already nice softwares to do maps but I didn't want to use {inaudible} or software dedicated to do maps because I only wanted to do what we call locate a map. This means you have a map with different points and every point is the name of a different restaurant. So, it's really easy to do and I didn't want to use big software. But I had to face the truth. I must learn basic programming to do that.

The moment I realised that something wonderful happened I heard that a group wanted to teach women basic programming so applied in July and was able to attend Django Girls in Berlin.

During that day I used my first line of Python and I learned that I could do it. It gave me confidence to develop it myself. So I was sitting in the airport waiting for a flight. I added some more features and right now working on adding a new part to deal with transcripts from interview so this is - so the name is quite ugly, it is KRL restaurant localisation. Here is the home page. You have a small map with just locater map but interactive with all the restaurants then you have also a big table with the most important info that I have to keep in mind. And the second screen shot is what you have when you click on the name of a restaurant on the map or table, you have all the info collected but also a nice thing it's a familiar extension, it's interactive and I keep all my photos of restaurants like that.

So, building my web app was easy and didn't require advanced knowledge. I just needed advice on how to start a one day workshop to learn basic Python in Django and confidence to work in 10 hour. My small app was up and running by using a framework in a library that were already existing. I didn't invent something extraordinary. I just invented something to solve the problem by compending stuff already written.

It may be seen as something common for developers but it's not for someone engaged in social sciences. I had to justify to my peers why I choose to do something that seems so distant from my field. My peers advisor understood why I did this. I have been {inaudible} students using Linux for so many years, when I told her I built my own software she was not even surprised.

So, why does it matter you may ask yourself why I am here talking to you about that. It is because I think that social scientists really benefits from learning basic programming knowledge but it is not that easy for us. I need to say that, we don't want to be left behind but more importantly, you can help us. So, what can you do? First, you can help Django girls by volunteer or convincing your company to be a sponsor. When we read all the implication we received for Paris event a lot of the women were studying social sciences humanities or language at a university. In France, women are more than the 70% of the ... you are indirectly helping student from social sciences and humanities.

But the most important part for me, is that this even does not target women already engaged it is open to every woman. You can also be a part of initiatives and create them at your university or community centre.

You have knowledge and you want to share it, most of the time there are empty room at university maybe sometime at community centre that is all you need. We have a friend, planning to do a small workshop for PhD student in social sciences in our university. We share the same ambition, we want to give the tool to people who have been denied the right to learn it. Nobody will care about teaching you basic coding or the basic tool like the tools you need for your work. It started to see people with such high qualification being deliberately left behind and not be able to develop the software they definitely need.

You can also do teaching material that I would assume that your reader come from them.

I have a huge request to people writing teaching books or tutorial on programming language, stop assuming that we all come from stem. It is silly, but really annoying. When I try to learn all my myself, all the teaching book okay, you learn basic stuff, have fun by doing a sequence, so what the hell is a (NAME - INAUDIBLE) sequence, I am here how to code, I know I will have to improve my mathematical knowledge at some point but not following basic stuff like loop. There is a book now, LO ..., she did a workshop yesterday, it is a really good example of how to do a teaching book aiming at total beginner who doesn't have the stem degree.

The step for me, to teaching material to social scientist, this would be useful to them, for example, crawler or scripting can help you do all the boring work like we saw in the beginning at the about the expenses street survey.

So my conclusion, I want to conclude on why it is important that you help us bring programming in social sciences and humanity. We, I mean by that PhD students and researcher in those fields wants to get in programming. But sadly, we are currently fighting convincing our institution that they need to change the point of view on programming. So we can spend the same amount of time writing tutorials or teaching other enthusiasts, that is why I am looking for helping hand. I am asking you to be welcoming to beginning regardless of the background, help us in bringing programming in all the field, by bringing tutorial ... doesn't assume that your resources come from stem.

Share to people who you thought were not interested in that kind of stuff. (INAUDIBLE) thank you.

(APPLAUSE).

DANIELE PROCIDA: Thank you very much Lucie, we are just going to wait a moment while we get the microphones back up to the front.

While we are here, perhaps I can start by asking you talked about some quantities at the beginning of your talk. Overall, over the whole course of your research how much time or money do you think this is going to save your having spent some time learning something new while you were doing it?

LUCIE DAEYE: I can't say for the money because for the first year I was unpaid PhD student that is one of the other problems from social sciences so, I can't really spend money I don't have in a way. But I think the most important part is not how much time, time is important but how much money we saw that maybe it is more for laboratory or research centre. For me at my scale, it is more gaining useful hours and most of all not be preoccupied by the question because it will be easier, to put it with my computer and I think it is a very important knowledge because from some software, GI software, you can build extense in (INAUDIBLE) it is more for what I could do next, that is more than for the right now I think.

DANIELE PROCIDA: I think my question may come into that, one of the reasons.

FROM THE FLOOR: One of the reasons Python is big in the ..., there are numb py, astropy, libraries for specific disciplines, do they already exist, is there scope for developing a similar set of tools for social sciences beyond just basic statistical stuff that probably the ...

LUCIE DAEYE: Right now there is nothing.

RUSSELL KEITH-MAGEE: Do you believe there is scope for something to be developed.

LUCIE DAEYE: Maybe with the statistics, I had an example that, it was in the eighties, I know a guy that developed his own statistical tool for interviews and in the social science context. They were no software that at that time. So choose to develop his own software. Right now I think, we can use different tools but we need guidance, the right now and more people to be involved to, to really understand what we need. I think we can benefit from also there is, I think, a great team working on the linguistic and, I can't remember how to say in English. But I think right now it is quite difficult to see what we need, but we need all we can find because we have to convince that, that is a field we need to be in and, I am sorry.

FROM THE FLOOR: Thank you for your talk., one of the things that struck me, coming from a stem background, I never considered the sequence that is never appropriate for a tutorial. Do you have examples of nonmathematical sort of starter projects for people who are learning Python and Django to help them?

LUCIE DAEYE: Yes like I said the work, by Tracey Oz bourn is like, she say that building an app in Django, it is only combining a collection of stuff. She, she is giving some kind of a Meta tutorial on tutorials. A tutorial you can choose to do whatever you want, she said, the problem from the documentation for example, is that the tutorial is only to build pool, it is only for that.

For with this books are aiming to do a Meta tutorial, that helps you do a collection of objects, so could be a store or a Blog or. But she doesn't aim at something in particular. So she gives you the tools and she say that you have to do this, that and etc. but, you can build whatever you want, it is not a strict and precise tutorial. So I think for the moment it will be very benefit to have that. Like now I am training, trying to find teaching materials, so maybe I have to think about that but in general, I think I have some idea but, I want to find other people to, and brainstorm about that.

FROM THE FLOOR: Thanks.

DANIELE PROCIDA: Thank you very much. (APPLAUSE).

DANIELE PROCIDA: Okay. Yann Malet is here to talk about salt for Django developers, so, are you ready to go? You are not ready to go?

YANN MALET: Not yet.

DANIELE PROCIDA: Try this one.

Okay we will try that again, this is Yann Malet, talking about Salt for language developers.

YANN MALET: When I did started to present or prepare this talk I looked for a way to put configuration management in a way that was approachable for people who don't have this abstract knowledge on how to deploy on plenty of servers. I found like the other things was to get on the vocabulary and know what we are talking about. So, this talk is going to focus on Salt in particular and on all the terminology and vocabulary used in Salt. I hope with this, it will help you to get started with this project. For me as a non-English speaker, it was the hardest part of the project and enjoying working with it.

So before diving right in this, I say a quick word about myself. I have been working with Lincoln loop since 2008. ... We have a paper version of this book., PDF don't seem like real thing, so it is real paper. It is real install.

You can find this presentation on git app.

As I told you, this is going to focus on the terminology and we are going the see bits and pieces of what is required to build a Django application but this is not a ... this is not a crash course at the end you will not get out of the right and write and deploy on all your servers, it is a long journey, but hopefully this will get you started and trigger this like the, the interest in diving into this domain. Which is really interesting.

The first thing is when you start deploying on multiple servers, master is fairly common names that wasn't didn't bother me, that was straightforward. I will jump to the other one.

At the other end of the chain you get a bunch of Minion's, English might find a play on words, but Minion is like a slave. When you get a slave you can do things like, yes it is a part of live demo going out which kind of scare me, I am terrible at typing.

So the first thing is, you can do things like asking,.

FROM THE FLOOR: We can't see it, it is too low.

YANN MALET: Okay, ctrl L ... ooo ... so this tell us like all the minion connected to our master, this is impressive list is one, the interesting thing is you can start doing things like asking, I want to get all the minion's that for some reason you have multiple distro I want to get all the minions are running Ubuntu and you get the result, but you can do things, I guess you cannot read anymore, you can do things like I want to target all the minion, so servers that are of a particular roles. So let's say the roles web, so web server so if in your environment you get a bunch of web server and you need to do something on them, install the new version of sage, because issues that need to be rolled out, this is how you might target this particular thing so you can do cmd.run and then ... there is a command you want to run. This part in Salt is ... you can have a command on your choice, targeted on the list of servers can be on, like you can target the minions that have more than four gigs of RAM or a particular distro or a combination of everything.

This is the first thing about the minion, the next thing, configuration management, the end goal to have a non-configuration on all the slaves and all the server which at the end is like make sure you can deploy on your form your software. This is done in Salt by having a place where declare the minions should be in at the end. The way this is done is using, the term of this is state, state are located in, in file I will show an example. So, the extension of state is sls, this one is our user state, this is a yml file. The simplest way is the one by default, yml, similar to the one you get in ... but you can have directly Python to build a dictionary which is less common but it is possible and json and anything else, but the default one is yml, so in that case we are making sure we have a dev's group which is, which is present on the installation and every time the state is run, we will make sure we have the dev's group, it is a way of making sure that we are looking for a part and replacing it another way of doing it, ... as file we will see like using, earlier the discussion about jinja using a jinja as a way, instead of coming from database, the model in case, they will come from a store of information which is called pillar and grains, I will come to them in a minute.

The part which is starting here. It is interesting we are in the pillar, we are defining a list of users that need to be installed. These users have also the key, so what we are doing there is we are iterating the release, making sure all the users are created so it might be all team members to say you have 20 team members who need to access your file you will find them in pub key and push and the new team member will be added.

So, here it is ... let's try to run this thing ...

So Salt is also self-documented so the documentation is using documentation string inside the packages and you can query this using SYS doc and let's say we want to know about state doc SLS ... which is starting here - so it's testing a given state. So my goal here is going to be execute the state users.

So what this is going to do it's going to deploy on my file in that case only one minion but you can imagine there are plenty of others, it's going to deploy all my user put the pub key in the right case so they can get access.

Interesting because you can also say I for some reason a user got deleted by someone or you need to check something, you can also just target one Salt tidy, Salt tidy is a unique name in all the states. So in that case I'm going to run exactly the same thing just for the way users.

So, we have things like states, how you want your minion to be, but this would be nothing if you could not have viables injected inside this state and the name for some reason it's Pillar. So native English speakers might think totally logical, for me I was just not getting it. Yeah OK pillar. So pillar is a dictionary where you are viable target minions, it's easy again a YAML file and you can do things like pillar dot item and maybe I want to know my wsgi config and the wsgi config is on this port but maybe I want to know something about - thank you ... about the users so it's a dictionary that will get injected into your states. What is important is pillar is a dictionary that can apply to one or multiple minion.

At the other end you can have also viables that belong to only one minion like the RAM. How much RAM do I have in a particular minion? Because may be some server configuration is a given percentage of the RAM. May be I want to give 80 per cent of the RAM for memcache or maybe I read post cress when it's not your job and I saw all this percentage for its viable and I want to templatize and create default configuration which is safer.

So, grain is exactly like - it's similar than pillar but it's for the minion, it's one-to-one with minion where pillar can be applied to multiple minion. So here it is on a graph which might help.

And the next steps is actually how you are going to say which state is going to be applied to which minion and this is done in the top file which is again a YAML file, which is again a YAML file where you specify your target, in that case I'm saying all the minion I should apply my users, I will have all my users on all the minion I will install on a bunch of further thing but I have also - pillar and states that target particular roles so I wouldn't solve for example varnish on all the roles that are varnish. Memcache one the ones that are cache and so on and so forth and the organisation you will find some people who will find things that prefer to have names based on us name rather than roll, this a bit like you choose your own way but in that example we are doing it based on roles.

So then we have this magical thing that's going to happen when I am targeting a configuration on a minion. We're going to grab information coming from the pillar plus the top file which will give us the name of the state data that will be applied to the minion and viables coming from the grains, we're going to smash all this together and push the states to the minion and at the end the minion will be configured as described so it is fairly nice and you can push this on all your cluster.

So this will give you the like the one command or one click deploy that people are looking for.

So, now let's dive into an example of states because I mention them but I haven't actually shown you what this will look like.

Again it is a YAML file. It is a dictionary which is pronounced as YAML. Let's go SLS - I will jump the first few line. Interesting here but what you're doing here you're actually declaring what the state of the minion will be at the end so line 7 here, I am saying I want - this is a package name I want this package to be installed, then line 10, I am saying in this block I'm saying I want to add a new REPO with digital key so I can check security of the package I'm installing I want file to be installed and managed to by salt and where it's interesting we are specifying a template. So the state is a YAML file where you can insert viable but it can also let you build configuration on the fly based on the template so let's look at what - at this default - varnish default - so this is like a very simple template where we are having - we have only 2 viables the port and the memory that will get sub seeded at RAM time and it's interesting to see that actually the memory in that case calculated based on the amount of RAM available. So, this is an example and you can do something similar with your database and you might also want to have like the override if you want to specify for this particular computer, I want to do something different.

And the last bit like package install is again fairly straight forward. It's interesting to see this bit here because this is probably like the other thing in configuration management is to get the ordering right. Like ordering is really tricky possible when you are installing you need to make sure that everything run in the proper order so you need to be explicit about what should run after what. And this button here line 28 is actually helping you to do this because what this is doing is telling you that at the end I want to have this varnish service whatever it is running, but I also want to reload it if I do a change in the configuration file. So if you push a configuration file of varnish it will do a varnish service reload.

So, until now, we have learned that we can do remote execution that we can install package, control the state of the minion, make sure service is running that we can generate configuration file based on a template which is really similar to a Django template. It's jinja one.

But you will tell me that the building in salt don't have everything you want to use or may be your favourite wsgi server is not installed.

It's interesting to know you can also drop to shelving out or reduce to a script that is one script or one liner, it will be several line, where you can pass on environment viable or again use template. This is a very powerful button and something like as simple as installing a wsgi project you will do a check out then you will do a command that will have a install {inaudible} and requirement file, you might want to add a bit of will file to make it faster but this is really the ID, it's that simple and once it's in place the deployment is really one command like you don't need - anyone can deploy no one will fear about it and it will self-repair if someone mucks up the server which is an interesting point.

And this is one of my last useful slides about the debugging and one of the best sudo in salt is the last one, you can put a break point anywhere in salt and get your codes top there and get PDP and get inspector code viable and know what's going on if what you're getting is not what you expect which for me has been a huge change compared to what I was experiencing with before.

This is just {inaudible} salt and management and much more. You will have salt cloud you can provision on AWS or digital ocean or vagrant - and mines that will get again another data store. It's a huge system, growing really fast, extremely powerful tool.

We are trying to package all this as an add on the high performance Django book because some people told us they would like to apply what is recommended in the book and to be able to deploy on their cluster so we are packaging this and it can be used either as a blueprint or as a learning material.

Thank you very much for your attention. May be have time for questions? {Applause}.

NEW SPEAKER: Thank you very much. Anyone want to come to the front for a couple of questions?

NEW SPEAKER: Hello. From all the Salt talks I heard, it's always about deploying something on several servers at the same time. From your talk I have the feeling that it can also be used to deploy different things on one single server, so if I have to deploy Django apps figuratively different apps but I still have to create a database, create a user for them wsgi configuration for them can I do this with salt?

YANN MALET: Yes the salt state I was referring to the one we're going to release is one that will complete in this time which will go from the wsgi server. We are using DCMS project to be the thing you installed and it will also install engine X, it will install varnish in front, set up the database, take care of the configuration and we're going to provide like as a starter kit we're going to provide 2 providers, one you will have all in one background images that you set up and you can just deploy everything to this box which is a vagrant box and it will be all in one and the other alternative is to build a cluster on digital ocean where you can specify I want 5 app servers and we will do the provisioning the sitting of the web servers and you will have like your cluster, like big newspaper that can {inaudible} page you.

NEW SPEAKER: OK but I have one server.

YANN MALET: Yes this is the case like background images is only one box.

NEW SPEAKER: And I want to deploy 5 apps.

YANN MALET: Yes. It's just -

NEW SPEAKER: OK thank you.

NEW SPEAKER: Thanks for the talk. Have you thought about anything about configuration and management? So all of your talk was about OK I have some state it should be in. Is there anything in Salt how you can say that you can use to say OK I did something in the past and I don't want it to be there anymore? Is there something about that in Salt integrated?

YANN MALET: Yes - it's actually - there is plenty of way. Like I give you the example, you are adding users but you have also people leaving the companies and you might want to make sure their user isn't any more on none of your servers and you remove them publicly from the place and there are buttons in salt built into remove users to remove packages to remove everything in Salt so it's just a matter of just like doing it.

NEW SPEAKER: Have you evaluated similar tools and can you say a feature or thing that makes Salt cooler than Ansible or public(?) because everything you say looks the same in Ansible or public tool ...

YANN MALET: {Inaudible} not having the jinja syntax and for me the selling point we used other tool before and for some customers they are using some of the tool you mention. The selling point for me when we are coming with Salt for Django developers is it's really match well HTML model of Django where model is not database it's pillar and grains. You get templates, you get a view which is kind of your state. It's really the same paradigm and you can use the same tools. It's like the mental shift is smaller than with the tool I have seen.

NEW SPEAKER: OK thank you.

DANIELE PROCIDA: Thank you very much. {Applause}.

YANN MALET: I have a few books if there are people in interested to look at it and there is coupon in your bag where you have 10 per cent off e-book. Thank you for your attention.

NEW SPEAKER: I am very pleased to welcome Mathieu, who is a Djangocon organiser himself, he was part of the team that organised the team in the south of France last year and he is talking about switching to py.test, in a large project from another framework, I don't know which one. We will find out.

(APPLAUSE).

MATHIEU AGOPIAN: I have been working for Mozilla for over a year and a half, recently a full time employee, a year ago I was a freelance, I was saying okay, I have been using py.test on other projects and it works well, I like it much, much more than unit tests and if I want to switch this, so I am working on the add on's website you can see if you are using fire fox which you are obviously doing. Yes, so, I thought if I want to move this whole project to py.test, I would rather do it now because I can't be fired being a freelancer. So, I am going, I would like my main subject here is how to switch but, I would like to give first, quick overview of what is py.test and why you should be switching.

If you are using py.test can you put your hands up?

Excellent.

For those of you who are already using py.test the first part it is going the be very familiar I think.

So, just a small piece on vocabulary, a test is just a small piece of code that you want to write to make sure that your production code is working correctly. Test runner, I think one of the most known ones nowadays is nose. Something that will gather a list of all the tests that you have written and that will allow you the run them easily with various parameters or in the various order. I am going in this talk mix and match about unit test of nose because they are working together, nose is the way you run them. It is a bit of a simplification, I think that will do.

Oh yes, just one point the last thing here, when I am talking about fixture, I will try the be specific if it is a data fixture, the ones you maybe used to in Django, so either Json files or sequence files, it is usually data you want to test. In py.test you have the injection system which is also called fishers I will be talking about that later.

What is py.test, it is a project that is actually older than nose from what I know, written by Holger a clever man, started a few of the well-known projects, heavily maintained by a team of really clever people nowadays.

It's features, you can write what we call no boilerplate test, meaning you can write the first line here and have a meaningful output if it fails if you replace one plus one, equal two, to something more meaningful to a function to make sure that the output is what you are waiting for. If it fails it doesn't give you much information if you do that with py.test, it will tell you what you receive and what you are waiting for, I will show some examples of that.

In unit tests you write in the second line which is a bit more of a boost.

There are a few things here that are also available using nose. One of the most likely feature or one of the features I like the most is the very last one which is the output capture, meaning that when you are running a test, very often you have out putt that is polluting your tests, report. So you will have some debug printing or some log outputs and sometimes it is a pain to see exactly what you want to know from that. So the output capture will capture all the output., you will not see the output in your test, only if it is failing., it will show you only the output that is relevant to the failure, so this is really useful.

So you can mark test, this is kind of putting a tag or a name or a label on a test or a series of tests and it is very useful if you want to, to run them separately that is the last line on the slide.

So it is also about failure reports, there is a link on there. I will post the link to the slides. If you go on this page you will have a lot of examples of how py.test is giving you much better reporting on your errors and failures in tests. For example here, you are trained to compare two dictionaries, it will tell you which key has a different value from one or the other dictionary, it will tell you more keys in the left or right dictionary, it is easier to know that for example, if you have very big dictionaries it will give you the information that you need instead of just spitting out there is an error because they are not the same.

Dependency injection, this is code fixtures I would like to name them. They are used to replace what you think know, set up modules set up package, set up class, all those replaced by the dependency injection mechanism which is actually just putting the parameter names in the test view, R F, which is a fixture by py.test jungle.

You could do that in the way that is documented on jungle at the moment which is usually just unit tests like using self-.overwrite and then you can overwright your settings. Here you don't have to import anything or reference anything. Py.test when collecting all your tests and when running them will just put in there what you need to run the test.

So that is really handy, you can of course write your own fixtures and there is quite a few that are provided in the core of py.test or in ...

I can't pronounce this name. Parametrization, is the way to have the same test with different inputs and different expected outputs instead of having a loop, a four loop or whatever in your test, and if one of the iteration is failing you will then have to investigate and say which one failed and why.

You could just write it this way and py.test will actually run this test more each input and tell you if it failed for which input it failed and what output it expected.

So you can configure py.test quite a big deal. This one is the global configuration, it will allow you to put to specify the markers you are using, this is for documentation purpose ifs you want to know all the markers you are using, you can know that on the common line for example, if you define them in, it is optional.

I am not going the go into too much details there. The context of py there is local configuration so you can be configuring your test differently for a package for example. I have just put here a little trick or hook we are using. We want to run the test with the test file, we don't want to invoke the py.test with the Django ... model with the parameter we will do it like that, also do a lot of things in the py file, so importing it here will automatically fix everything.

Plug in's, if you are using Django and I think you are, you should be using py test Django, which is awesome, well documented and maintained. I really love the third one here, the py test cache you have a thousand tests, and 50 of those that are failing for some reason, you don't want to rerun the 3,000 tests and to get to the first one. So just rerun py.test with the dash, dash lf for last failed. It will only run the last one that failed. If you keep doing that, it will only run the one that is are still failing, so if you fix one of those you will only have 59 on the next run and 58 and so on. You can use that one with X, which is failed fast, then run the first that failed and then the next one.

Pytest-sugar, you have a beautiful output like that.

Pytest-django, there is a lot of things to say, this is maybe the most useful feature from py test, between two runs instead of using the database each time, it could be a few seconds depending on the side of the database, each time if you run one test, if it starts rebuilding the whole database it can be unnerving. Reuse the db in the pytest I showed you earlier, it will use the same db, if you are doing migrations or changing the base and you need a new database, just run pytest with db and it will override this setting.

Also pytest gives you a feature that I like, if you want to write unit tests, ... this will help you to okay now my test is not optimised because I am doing access to the database, maybe I don't need the access to the database. If you want the access to the database one test or one module or a class, you can do that. There is also a trick to automatically provide it so apply this mark to the whole suit, so you have to specifically or explicitly provide the, run the access through the database.

So why should you change?

First of all I think it is more pythonic, I am being a bit cheeky here, I really like the simplicity of just using it the way it should be. Using asserts the way it should be. You can do that in non-pytest world, it doesn't give you much information but with py.test which is the failure and the assert, you will have detailed output, so you can just use asserts and be done with it and not learn all those asserts abstracts in your factory being ...

So it is as I said actively maintained. I put a few links here to the repository. At the time of the writing of this, the last time I modified this slide there was, there are were seven open py test issues and 313 on those.

So why change you might ask? Well why not? I think py.test has a lot to give, a lot and a lot. There is not much that is holding you back from using it. Py.test is compatible with your current code base, it can run a unit test and nose tests without any issue and if you start using py.test all the future tests that you will write for py.test will benefit from those features.

Also this is my favourite argument. Name is hard right, nose, what does it have to do with tests? I don't know, just joking here.

Why not change? This is a question I asked myself a lot because we had a lot of tests and unit tests using like the proper way to write unit tests with test cases that are inheriting from Django.test.test case, those cannot use the fixtures like the code fixtures from py.test, so dependency injection which is sad because it is really nice, but you can still use fixture ifs you had like, fixture that is are just running on all your tests. There is also no data fixture bundling so Django nose has a very nice feature that will just or is it? Okay.

Okay. So I will go very fast. You can go okay, so how do you change?

I made a few slides to show how many changes we needed to do of course there is too many. But, as I said, py.test is compatible with unit test test so you don't need to change much. Out of the changes only a few related to py.test although the rest were test clean ups and code clean ups I wanted to do at the same time. So if you start using py.test instead of this common line, it should be working in most cases.

I won't talk about that sadly. Anyway, if you have any questions, I will be happy to answer them. Maybe we don't have time for questions?

Yes okay. (APPLAUSE).

NEW SPEAKER: Yes, if there are a couple of questions, same deal as before?

Or not?

RUSSELL KEITH-MAGEE: Question by way of small plug for a related project. One of the tool like, py.test sells itself as one of the advantages better than a run output and being able to see the output conditions better. Slow down okay? Can't hear okay.

One of the advantages that the py.test people say that py.test has is, better output when the tests are running and you raise that one.

Are there, have you played with tools like cricket which you can use to look at the test while it is running rather than just improving a text output, console output. Sort of comparing it, is it big an advantage having a belter text output, should you be looking at other ways to run?

MATHIEU AGOPIAN: Yes, very good question, yes, I am not sure about cricket, I never used it. What I know is having straight away the maximum of information about why the test failed is very useful and, sometimes it can be sometimes you need to write code to know why it failed because you had like compare reason between two body contents of two pages you know that you are rendering and you don't know what changed in this huge output and py.test will tell you okay, at this point, this string is, this sub string is different than this one that you were expecting.

RUSSELL KEITH-MAGEE: ...

MATHIEU AGOPIAN: Unit test wills tell you, it is different, paste all the output that you are expecting.

RUSSELL KEITH-MAGEE: Plus, minuses ...

Okay.

DANIELE PROCIDA: I don't think you mentioned adopt py.test month?

Okay.

I think that is really good sign of a very healthy and welcoming community, so one of my projects was selected for that, I had couple of volunteers helping improve testing in my project.

MATHIEU AGOPIAN: I done a py.test, some help and mentoring and yes to be with them, if something went wrong.

DANIELE PROCIDA: Thank you very much.

(APPLAUSE).

We are going the go for a half hour break now. Or well, let's make it a 25 minute break, so back here for half past four. The lightening talks sign up board is at the back of the hall over there.

DANIELE PROCIDA: Before we start, Simon from Serren books the publisher will be packing up very shortly. If you want to get some books you can go now before he goes but there will still be some books tomorrow and Wednesday. The lightning talks board is at the far end of the marble hall. We think we have enough room at the museum tonight at the conference dinner for everybody whose ticket says that they want to come to that, so when you got your conference ticket you tick the box that says I want to come to the conference dinner on Monday night at the museum, I think you're OK.

And, Tim, I've got a cable for you so Tim come and find me.

NEW SPEAKER: Now it's my pleasure to introduce you to the next speaker, Erik Romijn; he is a Dutch Django developer extraordinaire and the developer of {inaudible}.

ERIK ROMIJN: I want to tell a tale of Django developers and security a few months ago I was asked to do a security talk at Djangocon. Last time it was a few years ago. I was thinking about what to talk about and I came to think about what it really is that I do here, what is that? Why am I here? What could I adhere? What is it really about for me when I think about security when a work on security? And the thing I realised is that I used to always think that it was about knowing a lot of things like knowing how the most obscure exploits, how the most crypto graphics docs work. However that led me to the point where I would basically never feel like I ever knew enough about any of these things. There are still many things I don't know and don't understand. However, what I came to realise is that a lot is also not so much about knowledge but about the way you think about things.

So, the first thing I want to talk about is the many things that I don't know. Because outside of all the things you could know about security and how see a sequence attacks were comes down to the idea that computers work very differently to we do and we know this because it's our daily job to translate what people tell us and what people think into explaining that to a computer which doesn't like us have the idea of I know you said this but you probably mean this. They don't have as much of an idea of creativity of what people meant versus what they said which is something that comes very natural to us.

So, when I look at something and how it's been built and whether or not someone may have done that right or not, a question I ask myself a lot is how does it actually know that? Because computers make their decisions mostly deterministically on a set of things they know. You enter your pass words and then it logs you in or not, it says your pass word is correct or not, therefore there must be some process by which it's determined whether or not these pass words are correct and it's a deterministic process and it's going to do the same thing every time in every situation.

Another way I look at it is would someone else if I have the same problem how might someone else implement it in software and would they think of all the little pitfalls in there? How would they explain it to their 3 year old? Because the same way if you explain something to a computer and it's very easy to make assumptions about how this would work if you explain to another person and what they might assume and that they know what you meant but when you explain do computer what to do it doesn't know what you meant, it doesn't know what the idea was behind this, it doesn't know anything except that which you told it to do with the information that you gave it.

Let me show you a little example of oege; it's not a Welsh food it's a Dutch name of a server; this is in 2006 when people still did PHP. This is when Django was just released. And oege at the university where I studied it was shared literature where students could play with and everyone could request an account and you could run a few speech scripts so it was a Linux place for students in the days when you couldn't rent a server for a dollar a month.

And there was a sign in process. It didn't look this fancy because it didn't exist back then but it was the same thing. You enter university, user name and pass word and if it was correct it would create a local account on the server for you or change your pass word if you had one and email you the new details.

Is this typically a thing you would find that a PHP script on a web server is able to modify user accounts and reset pass words of other users is this typical thing you find? Some people in the first row have seen this before. Most PHP scripts you run would not do this sort of thing on purpose because it's risky.

So, seeing this I figured that maybe I wasn't sure how this was actually built, but given that somehow this script is apparently able to reset other peoples pass words on the local machine so not in some web application but in the local Linux machine reset their pass words and create accounts that seemed word while to look into so I had a shell there I tried to read the script and unfortunately can't read it. That's a shame because if I don't know how it works it's probably going to be a little tricky to figure out how this may or may not work and how it may or may not be properly built.

So, I went a little deeper because evidently the web server that is running the script is able to read it so I'm not but I'm seeing the result in my browser so the web server must be able to read this, there is no other explanation, it must be able to read the script, what if I could trick the web server into reading the script for me? This might not work anymore nowadays because people separate these accounts but this was in 2006 and nobody would separate this so if you couldn't read someone else's scripts yourself you could write your own scripts where the web server would try to read the other scripts so I open the page and how comes the PHP script. I wrote this in hand I don't know if it's valid PHP but it's probably close {laughter}. I hadn't done any for I think 6 years and it's a pretty straight forward script. It takes HPI, it reads queries, based on used name and pass name I gave it, if that's invalid it guesses my email address then calls do sudo and a little bash script with user name I entered and my email address. There are issues with this but that's not the issue I want to get to unfortunately we don't have enough time to explain all vulnerabilities present in the system {laughter}.

Apparently the user that runs the web server has the ability through sudo to execute the script with arguments every time because it's different for different users and if I - it would not be unthinkable that someone who wrote this probably thought they were the only one running sudo script as a web server user so I tried what if I make my own script use someone else's user name and my email address not my official email address a random email address random user on the system and it mails me a new pass word for them.

This also worked for the root user. {Laughter}. Also there were issues in the script itself so you could actually without changing pass words I later discovered obtain full root access. I emailed them about this and they were upset someone had find their way through all their careful scripts and they employed a little hack that made this just slightly too hard to figure out at least too much work. : Now I don't think you'll find exact vulnerability very often but you don't even need to know PHP to find these things out, it's just looking at clearly this PHP script has the ability to run things as root or change things to peoples user account so it's easy to forget for someone to check at the every appropriate step whether or not you're still setting the right user account whether you're still authorised to do that.

So I think it is to relentlessly ask questions how this thing works, if I was to build this what would be the simplest way to do it? I know it's wrong but what would be the simplest way and may be someone else made those same choices. Taking things apart to see how they work, this is easier in some cases than others, and to think a little creatively so outside of what - not from the perspective but what the person who built it who was trying to do minimal work in-between all their other work to make a simple user script but to think beyond that to think which things they might be forgetting. When I look at the page now in 2015 I did see this so I think they've made some dramatic improvements.

I should add one little note: don't ever do this kind of things on other people's systems without their permission. I sort of had permission. I would not do it again with the sort of permission I had back then. They call it computer fraud, nowadays they know how to prosecute this and it doesn't end well.

So, Django. Being interested in security I like Django a lot because it's reassuringly secure that's what the new web-site says, taking care of SQL injection cross site scripting cross site request for {inaudible} and there is very good authentication system and when I see it when I use Django I reduce the number of ways I can screw things up. I can build my own authentification system but if you read the documents of Django authentification you would not realise the details in there such as if somebody tries to log in with invalid user name so user that doesn't exist then we still hash their pass word which has no function, usually we do it to compare to real pass word but we still have their pass word otherwise there are tiny differences from which people with relatively few requests figure out which users requests and which don't and that we consider a user vulnerability so all these details not just in authentication that's just the part I know best that are easy to overlook and things we've learned through many years of having Django and having security issues and security hardening as we like to call it. So don't build your own things without looking at the code of Django at least.

That doesn't necessarily mean there are no ways to screw up with Django. It's reasonably easy to introduce code into your web-site so anyone can introduce any code on your server. Need to do 3 particular things. Set your session engine to the signed cookies backend you set your serialiser to the pickle serialiser then you have a secret key which is not actually secret. Do you think these are in defaults? I see someone nodding yes even. I'm a little disappointed at that. This one used to be default up to 2 weeks ago. Session engine definitely not. Secret key still a bit of an area where I like us to do better but it's very tricky.

But, so, it is possible to do things which are very bad but they're not a default and each of these settings comes with massive warnings about how if you screw them up combined with other settings you will be introducing very bad vulnerabilities. So, even though things are possible, Django makes it very hard for you, much harder for you to do this and especially harder for you to do this by accident.

There is a nice security in Django page in documentation, I'm not going to read through it because I think it would be a poor way for me to spend your time. It details - sorry if you want to know how things like SQL injection works and there are a few educators where SQL may still be an issue or if you want to see how CRF protection works there is other resource. There are 2 scoops of Django which has just had its 1.8 final release so you can order the print buy the e-book. This has a massive chapter on security. It's getting so large it was split up. Which has been refused by many people as well over the various resets this book has had. Yeah I would absolutely even without the security chapter buy this book. Don't think about it just buy it. If you cannot afford it email me and maybe there is something to be worked out other than that there is no reason not to have this book.

What like about 2 scoops of Django is it goes into security features that you wouldn't really think are security features like Django forms are incredibly important security feature.

Here is an example from the Apple developer portal and there is a select option here which allows you to select these seemingly useless IDs. This is something you use when you register a new I phone app so if you are IS developer like me if you want to publish new I phone app you go do this part and select something here. However if you manage to select particular values here, you can read other people's data. So, manipulating this value means - which I found a way to do - meant that I could publish an app on the app store and Apple would say it's all fine and go through their week long week view and everything would be fine and it would function but as soon as you started it would steal all your drop box credentials and there was nothing drop box could do and there was nothing that was wrong they used document features. We don't have time to go through the whole vulnerability because it's apple's certificates and other crap but there moose little select option here and it would list the number of values valid for your account but if you would just edit one of these and submit the whole thing, then it would just accept it, so there would be a number of valid options and it would produce those in the select when it's rendered it, but if you would just submit any other value it just says fine, I've created your account with the other ID that belongs to drop box or Google or Facebook and now can read your data.

If they will have used Django forms I think this is all some horrible form of Java. They have some custom think that I think it's even more horrible, they have their own custom weird framework thing. If they used Django and they would have used a normal form where you could select multiple options then those same options would be used to render the field and those same options would be used to validate in coming data. In Django if you mess with the values in select it will not succeed unless you use something other than standard Django forms.

So it makes it very easy not to introduce vulnerabilities. It seems rather innocent but sometimes these things can escalate into very bad vulnerabilities.

One last resource I run a tool called Erik's pony check-up which checks 12 things in your Django web-site deployment. It is all pretty basic but there have been Djangocon size for example where you have D book multiple enabled, something people often forget which is painful. If you run this test and you manage to score 100 per cent then you become eligible for the Erik's pony check approved sticker. I only have 4 with me but in my hotel another 50 so if you have 100 per cent score come find me. I'll need to see the result to make sure of course you are not cheating and then you will get one of these self-designed stickers, I made them all myself.

So yes it's a basic tool. It's a good one. Everyone should be able to pass this.

It was raised in the earlier session already, if you are in doubt whether you may or may not have found sue security vulnerability in Django then you email security at djangoproject.com so if you think it might be a vulnerability but not sure always better to email first then maybe we can conclude it's not a vulnerability that it should just be publicly tract in track git hub but the opposite is much worse. If you think it's OK and it's not a vulnerability and you report it publicly that means we have a publicly disclosed security vulnerability in Django which means we have to do all sorts of painful processes at much greater speed than we usually want to do, ignore pre announcement times, it gets very messy and unpleasant. So if you are not sure, think you might have found something in Django, first email we can always tell you it's fine and can be retract publicly.

So I want to run through a current issues I see in Django. Not all of them anything to do with Django. They relate to Django.

So his is the web-site of the conference. Or is it? This is actually the web-site of Djangocon Europe. Who thinks it is? Everyone seems very confident. How are you so sure it is not. It's different it was the web-site. 2 months ago it looked like this. Or did it? Because when you look at this web-site, when I was looking at this web-site that was not an encrypters connection. It wasn't using HTTPS. Now HTTPS is essential in the case where you do anything where you transmit any kind of secret data back and forth so if anyone looks in on your site that means you should be using HTTPS. It's so question to intercept log ins that happens over HTTP. It is so simple - it's legal in the Netherlands to do this, not in this country but in the Netherlands it's fine to listen into open Wi-Fi networks and the unencrypted traffic that goes along it. But the Django website is different because there is no log ins on the web-site, there is links to teeto(?) but - {inaudible} you could argue why do you want to use HTTPS here and the reason is there are 2 parts typically to HTTPS. One is assuring confidentiality so if I listen in on your traffic I don't know what you are sending back and forth. But the other part is integrity. Because there is display in HTTP there is no guarantee the site I am seeing that any of you are seeing that is actually the site you are trying to visit there is no guarantee, I am seeing the site that any of you are seeing, there is no guarantee of integrity, it is fairly simple, illegal but simple any of you can learn cycle tag 30 minutes to intercept that and change the schedule. Not particularly useful for a Djangocon maybe, not so much that the it is secret but it is still important that it is correct. And it was the way it was written, so the number on issue I see in any website, Django or any other, absent or misconfigured https.

How do you do this properly, first of all enable https for the entire site, there are ways to do this for parts of your site but they are very error prone and there is not a point to it.

Once you have done that, that is what they say when they enable https, it is important but any other step next to this, is you redirect all users to https, if you offer them as a choice, most of them will use the https site. In addition to that, once your https is stable, you can have strict transport community, the power of redirecting people, so it is, it makes your browser remember that it is supposed to visit the site and only do it through HSTS only if nothing dodgy is going on., it reduces opportunities for people to tamper with this. Once you switch it on it has a lifetime and it is difficult to disable during the lifetime.

Also set the secure flag on all of the cookies, which means they will only be transmitted when the connection is secure and all these things you don't have to work them out on your own, since Django 1.8. We used to do this, this is part of Django, and these are just a bunch of settings.

Then the last thing is to make sure you have to write ciphers, versions, there are a choice of generators that will help you because https is flexible protocol and there are many, many cipher configurations are they are very weak, something from something I could break or just becomes simpler to break or only in certain cases, no need to use the weak cipher. The hsts has a test, updated with all the latest vulnerabilities and you should just be able to score an a there, my bank currently scores an for, so it is easy to go for a higher mark, if you want to know how it all works, I can recommend this book, excellent book, typically use for https and it explains many concepts of cryptography as well.

It goes into all the vulnerabilities we have had in SSL and how all of them worked and the type of things we do now, how it affects other systems.

This is my favourite technology book.

Number two, happy path only testing. So, let's say we make a view that based on the class based generic view allows us to see the details of the customer, define the customer model and define the template. Pretty basic. Being good Django developers we write a test, so a little test, I am assuming there is an obvious created, so we can retrieve the detailed page for kusmur one and make sure we have got a valid response. It is a bit of a naive test but it needs to fit on the slide. However it may want be smart if everyone on the internet can view the details of our customers, so let's put it behind a log in page, this may look daunting, wrap it in log in required, only see it once if you are locked in.

We will need to add a log in, in our test as well, log in's are required, a little further on we find we have multiple views that require log ins and so you extract it into a mix in, and you say log in required, mix in. This all works, let's see what happens if we would introduce errors, let's say I were to remove this line, would our test now fail?

Is our tests would fail, that is good. We broke something, this would break something and our test fails. If we break things our test should fail. If the template doesn't exist, what happens?

The template doesn't exist. Yes.

The test would fail, this is good. If we would accidentally forgot this view requires log in's, what would happen to the test?

The test would still pass.

This isn't actually a detailed view, I accidentally turned it into an update view, I lost my train of thought half way, this allows updating, what does our test do with this change?

It still passes.

So, basically this view which is supposed to allow viewing details only to people who are authorised to do so and it is appropriate test if we would accidentally write in the view, allows anyone on the internet to update the database, the test would pass.

The test would say 100% and wow that is true, you have ... you have tested all the lines, you have tested you could view the details of that page, it is more important even if you test the other cases, if you are not locked in and trying to do this, what if you are trying to publish a post, maybe if you are locked in and you are not the right type of user?

This may seem like something you can just work out and something that seems obvious and you won't forget when the project is the side of the single slide. I run across projects that had serious ... published their entire confidential data out on to the internet. They didn't have any tests., I haven't found a good way to measure yet whether I am doing it efficiently, I have a lot of tests, test all the failure conditions but they don't show up in code coverage, that makes them a little tricky.

Third one is media up loads. Not often discussed this n this context, the Christmas speech of the Dutch Queen was leaked the day before...

So when they say leaked and hacked what they mean is someone figured out the url to video was this and the url of the video to last year was identical, except it says 11 and the date of Christmas doesn't typically change, someone figured out, change the year. Sequential number here, based on other videos within a few minutes you could find a video or any other vid you they would publish as long as you knew the intending publishing date. Christmas but many other cases where you could do this in advance. This also can affect Django websites I see this a lot. People up load things and they assume that the url of the image will remain a secret, only viewed by certain people, but the names are incredibly predictable or easy to guess the other files and just as easily accessible.

Something else to note from media, if you do, redirect as I said in a previous part that does not affect your media and study files if you have deployed it. Because that in case your web server, your apache serves it directly, which means Django will not be able to do this part.

Doing media up loads properly, ideally place them on a different storage and different domain, the reason is that up loads from your users are unreliable, they could contain a lot of java script, so you want to keep them secret. Some of this more in the Django documentation.

Depending on there are some conditions for this as well, the file type, the, serve them with the correct type. The browser will not interpret the image as a bunch of java scripts. If you need access control like the queen, two things you can do. The very lazy option, to serve the media files through Django, simple and lazy and slow. It means your Django thread is busy. I do this in a few places where it is a very low traffic site and I can get away with this.

The other option is use, accents file, supported by all major web servers where you can tell the server to serve a certain file and not do it yourself, because the request is through your Django app you can do all the access control issue, so you get the benefit of the access control and the logging and whatever you want without the performance hit. Slightly more to configure but just as safe.

Number four, out-dated software, three versions, 1.4. There is 1.7 and 1.8. Any one run six in production? You are only missing a single security patch. Anyone 1.5 in production? Django 1.3ah there is still people, 20 patches missing.

These are sort of rough numbers so I haven't checked each of them whether or not they apply to all versions, release two weeks ago, only apply to 1.8. Anyone on one point zero., I wouldn't raise my hand either.

So, it isn't affects Django, ... use so many dependencies in my project, what I use nowadays is a, requires an io which tracks all, looks through all the requirement files and dependencies in there and figures what release you currently use and what the latest release is and whether that is a little out-dated or whether there is a security vulnerability. This costs a tiny amount for private projects but for public it is free.

I don't want to do this anymore.

So, we have been talking a lot about how to do some Django and the programming side of things. But there is also the side of the humans who try to use our software. This is the on-line banking software, the one that is (INAUDIBLE) it is called ... which translates to my ING a pretty standard naming format, uninspiring but it works.

But when people visit this you would expect them to go to [name] or go to the main website of the bank and then click on internet banks. This seems obvious to me. At some point I was with a friend, she would Google mijn, which is a generic word like my car.

Enter a pass word. Never checked what she was doing, or looking at. I have seen this in a more extreme form, where this Blog, I have never heard of it otherwise, wrote about Facebook log ins, this became the article, Google placed it at the number one hit if you Googled Facebook log in. Tens of thousands of comments, restore the ... writing the user name and the pass word in the log in form, but it is funny, these are also the people that use our products. These are the people that, that log in to our systems and just say you know, they are just idiots right, they deserve to have their money and Facebook accounts stolen but, because the reality is these people are not aware of this, they don't realise these things, I don't have a solution for this problem but the similar thing is pass words. This is a cloud of common pass words and not very surprising. If you allow people to choose their pass words most of them will choose horrible pass words and people in this room who should know better.

I use pass word strength checking and most of them suck. I absolutely hate them but this is the only one I found that I like. Made by drop box, also hit a lot of pass word checkers and it is extensive in that for example, fairly short pass words but they have to be complicated. Then dictionary words but you need a lot of them. Allows everybody to use a strong pass word without using dogmatic rules.

Google authenticator, you see the tokens on the key ring, I use SMS a lot. There is many sides to this. We can't cover all here, if someone phoned your users, I am from IT, can I have your pass word, some of your user wills do this it is a training component, it is a big issue. If you have all your media that is fine, if people phone the users and ask for the pass word that is a serious problem.

Now, when I tell people nowadays about you know, you should really be doing https on pretty much everything, someone is going to say, the NSA can read anything, and there is no point to https anymore, what about heart bleed there was vulnerability, they will complain how they made it absolute and they don't follow cryptography and there is no point anyways.

There is some truth to this, but, this is only a valid argument if the only thing you would be worrying about in the world is the NSA. There is absolutely no chance that with my million Euro million budget I stand a chance between government agencies with nearly infinite budgets and immunity ..., this is a satellite launch -- so people with this motto and the budgets there is no way I can make a difference there, but that is not what I am trying to do.

What I am more worried about is these people with ski masks and lap tops turning up anywhere. I can explain why the NSA was able to compromise the system., I can't explain how the person at Djangocon was able to access my system. I can't explain it to myself, I can't explain it to my complaints or the people whose data I am responsible.

Backup and recovery, of course we all make backups and documented recovery plans which we have tested recently. This is the first thing about risks is it okay if your web line is off line for a minutes, a day, how about two days, how about 12 hours? You have to make choices when you make the choices then you can plan how you build for a system that is able to meet what you need.

So, if you can't be off line for five minute this is not a problem, for the things I manage it doesn't matter, fine if they are off line for half a day, we try not to. We have designed the backups and the recovery around that thing.

Looking at what risks we have, what things are you trying to defend yourselves thing from, it is important and difficult.

So, I have been talking about all these things, why should be looking at risks https, why Django is so great.

So, which leaves me to talk about why this actually matters so why would you bother, surely no one will try to compromise anything you own? I literally had this response, it has been fine for the last year, so we are not going the fix anything at all. It was badly broken.

But the way I feel is that, as the developers of the systems even though our ability to make decisions over them can be sometimes limited that doesn't mean that our responsibility is any less, if we build system that is we know to be flawed, if we refuse to take our, then we refuse to take responsibility for that which we have chosen to create. Sometimes I get people on the phone and they say hello, I lost my password, can you change it for me? Then I say no I can't because I don't know who you are you could be anyone telephoning me. Surely you wouldn't want anyone else to be able to phone in and change your pass word because you would lose your account and some of them get annoyed but some say that's very good I didn't realise that would be a problem. There are so many things that don't occur to people. I was surprised by explaining to them, no that's a very bad idea and it will all go horrible, how many of these people suddenly came to realisation and hopefully complain about these things to other people as well if they see these procedures not in order.

That doesn't always work though. Some people - there are a number of people who don't really like me anymore because when they come up with their plans I say, well, you know, that's going to be a problem here, that would open the door to massive fraud. So, when you try to improve security on your own systems, on other people's systems, on systems you built for other people, not everyone is going to like you, not everyone is going to appreciate this because many people find security really annoying until something goes wrong. It is a very - it's not very visible as long as nothing goes wrong and which makes some people think it's obsolete.

However, when I find that sometimes getting to me a little like nobody actually cares and none of my clients a little overstatement - but sometimes it feels like none of my clients actually care about any of this and I'm just doing this for myself. When I find it getting to me a little I always remember this quote which I think applies to much of our work, also {inaudible} security by Isaac Asinov, there is no way I can single handedly save the world or perhaps make a perceptible difference but how ashamed I would be to let a day pass without making one more effort.

Thank you very much.

{Applause}.

DANIELE PROCIDA: We've got time for a couple of quick questions so thank you Erik.

NEW SPEAKER: So, when talking about https, what is your stand on public keeping(?) because you talked about strict transport security but not about keeping -

ERIK ROMIJN: I haven't looked at key pin(?) much, yes it is a thing, I'm not sure how widely supported many of my users are on 7 so it's not a big thing for them yes but I think keeping is a great development that yeah I should have said more. Next version of this slide back I think it will be in there.

DANIELE PROCIDA: OK Erik I'm sure you'll have many people coming up to you for further private questions about their vulnerabilities they may have so thank you very much for that talk. {Applause}.

For our last scheduled talk of the day before the lightning talks arrive in 20 or so minutes time, Maik Hoepfel on privilege and moral duty. {Applause}.

MAIK HOEPFEL: Hello. So, I am Maik. I am a core contributor to Django Oscar and I currently held my meric(?) sell car parts and {inaudible} and recently I have both been thinking but also renting a lot about us web developers and I'm cutting my talk short so you have extra time to yell at me at the end of the talk. Here goes.

We live in a world where the most comprehensive collection of human knowledge or dating sites that create millions of relationships each year can be run by only a handful of developers. Our ideas of open source have changed software are taking a hold in hardware in our education and as commons are even impacting economies. Other developers are stopping deaths in traffic and are providing infrastructure to battle diseases.

We can build tools to find earthquake survivors in an afternoon and we can power and visualise immense amounts of data armed with nothing more than a library or 2.

Meanwhile, what professions can we compare ourselves to? How many lives can they impact? Are we carpenters, mechanics, even architects have less potential to have such a positive impact on so many people’s lives.

That same power that can change and save lives can also make a lot of money, but does mean our skills are in high demand, which results in a range of perks for us.

Like seriously who else gets to take a week out of their work life each year merely to assemble Welsh cakes, French wines or Polish beers? Most of us don't have obey dress codes and have flexible work hours. How many of your non tech friends get free lunches or showers at work or have the ridiculous job security that we enjoy?

Not to speak of the money earned. If Germany is anything to go by, all of you on your country's top 10 per cent earner's.

I'm not saying we're the top dog. Other professions get comparable privileges but if that privilege wasn't inherited they usually had to work long and hard for it. Think lawyers, doctors, elite business programmes.

Many of my peers didn't even finish university. So make no mistake. You belong firmly into the 1 per cent.

This brings me to my first point. Companies are out bidding each other to get us to work for them. They pay hefty sums to recruit us to get more talent on board. I am left with the impression that many of us take no advantage of this. Too many of my friends are afraid to ask for a 4 day work week or a wage that they think is fair. I see Mums and Dads who love no one more than their kids but still submit to unnecessary 60 hour grinds.

I know asking for something out of the ordinary is scary. You might not think that your skills are that valuable but if you are here I assure you they are. Please realise that you have excellent standing in negotiations about the amounts you work and the compensation you get. Come join the light side and negotiate more time for you. Get a Home Office day. Free sometime to exercise and eat healthy. Be there for family and kids. Have time for your hobbies.

And a decent company would rather keep you even if you work a little less than see you leave and talking about the value you bring to the table and the compensation you can expect might feel strange to you, I understand that, but it is every day business for your superiors. No one will be offended by reasonable request.

What's the worst that can happen? There is plenty more companies looking to work with you.

For the last 4 years, I have only worked 2 days a week for pay cheque. During that time, I lived in Australia to work in a motorcycle restoration shop. I had a go at starting my own business not far from here in lovely Yorkshire and I travelled all across Europe either hitch high hiking or on my motorcycle. I taught others to maintain bicycles became a maintainer for open source project and was able to be there for my family when most needed. Yes, I don't have other mouths to feed but one can't call my lifestyle modest and my savings are in good health. It's likely not the life you want but it serves my point: there is a whole lot possible if you ask for it.

As a note, I am not advocating you shouldn't work 40, 60, 80, 100 hours if that's really what you want if that's where your heart is. May be you have significant shares in a start-up. May be you are the start up. I just hope you are aware of the law of diminishing returns. I hope you're aware of what was said earlier by the wellbeing service and please don't just do it for pay cheque. Don't let a ticketing web-site be your life’s legacy.

At the end of the day, most web-sites that we built aren't terribly important. Most frequent of Django sites are built to leave comments on articles or share photos. My friends and me sell car parts, heating elements, tickets or cigars, we process bets and payment or build technology to watch cat videos on all our devices. Yes, there is worse things to do with one's life and I'm not questioning that it's enjoyable. My point is most of us hardly change the world with our day job.

But our world still needs plenty of changing. We live in a time where sexism is still rampant in our industry. We're still at a point where it's not common sense but worthy of celebration that same sex couples get equal rights. Corporations and governments are leading the biggest war on privacy yet and that was led on our home turf. Who is going to fight in it if not us?

Things look even bleaker when we try to look at the big picture. People risk their lives every day to flee oppressive regimes and we let them die at Europe's borders. The systems we have created are fuelled by war and injustice. And the current level of climate change threatens both nature and humans.

So, there are plenty of things wrong with this world. We yield the tools to fix them. And we can make time for it. I say we have the moral duty to make this a better place.

Thank you.

{Applause}.

DANIELE PROCIDA: Thank you very much Maik. We'll have some questions. Try and keep the questions short so Maik has sometime to answer.

NEW SPEAKER: Thanks Maik. It's a bit of a downer to talk like that and I hope that doesn't affect your future in any way but I'm absolutely behind you and I agree and when I look at a room of really bright people like this it amazes me how little we actually invest in changing some of these things and I think that's something really that I miss and I would welcome any movement by people in this room now or in the future to get together and actually identify things that we need to fix the priorities and actually start doing something ourselves and actually setting up an initiative. What are your thoughts on that?

MAIK HOEPFEL: Well, we should get started somehow. I do think there is really good initiatives existing already. I think people network quite well already be it parties, NGOs, be it the great work Django Girls is doing. Personally I'm more a fan of joining existing proven initiatives whatever they may be if they're political or whatever their goals are, but I'm certainly happy to discuss or get together and see what we can do and what existing options are.

NEW SPEAKER: I think it's a good response. I think it's important to compare organisations like Greenpeace who seem to have spent a lot of time marking time and not actually making any progress and the person Paul Watson who left Greenpeace and who has sunk 7 pilot whaling vessels almost single handedly and just brought to justice half a dozen ships that were illegally fishing in the southern ocean the southern tooth fish that's severely endangered and he's making a difference and he's getting off his seat and he's not waiting for big organisations to do it, he's doing it so if people round here were interested in getting things done; I would certainly be prepared to stand up and I think you would too.

MAIK HOEPFEL: I agree but I think ultimately we - I don't want to tell people what to do. The idea is that people have an issue they care about and they get involved there. I think that gives the most motivation.

NEW SPEAKER: Hi, really quick, thought it was a good presentation, a lot of good stuff in there. One thing, you said we are the 1 per cent. We are most definitely without a doubt absolutely not. There is no one in this room that makes 10 million dollars a year. So I admit we're definitely privileged, we've got a lot of stuff going for us, I practically fell into this gig and it's awesome but we're not the 1 per cent and when we say things like this group is the 1 per cent because they're really fortunate we're really missing the whole concept of there is almost everybody and there is a very rich group of people who are profiting off everyone else's misery so awesome but I really want to make that part clear. {Applause}.

MAIK HOEPFEL: (APPLAUSE) ... I know that is a provocative statement. You are white male first world country, university degree, earning well, I think it depends on what you compare it to. I do agree that also ... meant something.

NEW SPEAKER: I think if you said compared to the whole world maybe we are the 1% but our individual countries or the western world, then we are definitely not.

FROM THE FLOOR: What if you start a business and a project that you realise your business and project will destroy someone's else's life, integral. A few guys took it out codec, took down directly 10,000s of jobs, how do you deal with that? Like you took bread from their hands, from the hands of thousands of people.

MAIK HOEPFEL: I think that is a moral question you have to find your own answer to, I think it depends on whether you believe in the current economic system or not if you think that is a justifiable action to take or not. I think most people will think that is a good thing in the system of capitalism.

But that is really your personal values I am afraid.

FROM THE FLOOR: Okay but with the capitalism and, can we really change that? We cannot definitely, I mean we cannot change it with the photo app but we can change the lives of 10,000 people, there is the problem. That is more of a part for the first part of your talk tomorrow. Yes.

DANIELE PROCIDA: Thank you Maik that was very thoughtful and intelligent. (APPLAUSE).

And also very courageous to come and say some of the things you did, thank you very much for that.

I just want to say that actually in the attendees here today, we are, we have several people from 1% club in Amsterdam. They have built a Django platform for social projects a crowd funding system for project that is have social value. We were going the mention this tomorrow or Wednesday but during the sprints, we hope the get together some people who would like to work on maybe setting up an enduring Django for social good or Django for social value project that is part of the Django community. I am not sure what form it will take or how it will work, but we hope people will start talking to each other about it. So thank you very much. Once again Maik.

(APPLAUSE).

It is time for lightening talks we will get through as many as we can. We will get through as many as we can now. Peter what time do you want to take people out on their walk? Given that they have got be back at the museum by on the steps at say five to? Five to eight.

NEW SPEAKER: Half past eight. We won't take that long, maybe maximum 45 minutes on this. If you are not on the walk with Peter, please be at the museum at 7:45 we will be at the museum at 7:45. All right, so everyone knows that, I think there is room for you all.

All right if you don't get called up today, we will have another go tomorrow. Russell, Andrew, Marcus, can you line yourselves up there?

In case you don't know the rules of lightening talk club, the first rule is that you have a maximum time of five minutes. At the end of five minutes you get cut off. In the last few seconds, I will raise a hand and you can start clapping very quietly like that. Just the two fingers then when I raise my hands right up that is five minutes gone, burst into applause and the speaker will stop speaking.

You all ready to go?

Russell and your talk is...

RUSSELL KEITH-MAGEE: I am a Dr .

Yes, I am Russell, you may have seen me at the mike asking questions when I am not asking questions, aim a veteran of the core team and the member of the advisory board and the president of the software foundation, according to the organisers I have a much more important title. I was issued the very important ticket of dork1.

Just so you know!

I am founder of the beeware projects, a big part is Togo, a cross platform allows you the build natively in Python. They are work in progress, if you are interested, come and talk to me through the sprints.

I am the cofounder of a start-up company with the stresses that go along with that. Also a doctor. My doctorate is in computer science, "models of learning and development ... (INAUDIBLE)" the reason I raised this disclaimer is, this is not medical advice. What I am about to say is not medical advice, it is personal anecdote. I hope by sharing I encourage others to do something I avoided doing for a very long time.

You may have noticed I have my fingers in a lot of pies. I have a stressful job, I volunteer with Django, got side projects, have a son with learning difficulties, I have a daughter, wife, over the last few years the combination of the stresses has taken its toll. Over the last number of years, temper is short, found myself being frustrated both with myself and with others. I found myself doubting my own ability sometimes not just imposter syndrome stuff, but a genuine belief I was incompetent sometimes. My enthusiasm to go out and socialise with friends, to organise to go out to have dinner with a friend, disappeared to almost nothing.

I also had intermittent health problems with shoulder and back muscles. Wasn't a continuous decline, good days bad days, this pattern has been going on from somewhere between 18 months and two years. A few years ago, I hit bottom. By about nine in the morning I had no energy to continue. I didn't have energy to give a dam about anything.

I went back to bed, I buried my head in my pillow and I cried.

That was the final straw for my wife, I have a very loving wife who is my absolute strength in everything I do. She pushed me out of the door to see a medical doctor. I was diagnosed as being in the middle of a major depressive episode. I am okay now. I am being treated. At the start of the treatment, way too early to declare victory, but in this short time, the release of knowing what I was feeling was not right, can be treated. I will take any improvement I can get.

The reason I am going public about this, in retrospect I put off getting help for far too long, I didn't want to admit failure, I didn't want to admit I was mentally ill.

Most studies agree that depression affects one in 10 people in the population, our industry statistical prone to the tune that discussions about burn out are almost a constant undercurrent in day to day conversations, I want everyone who is here today, and anyone who watch it is video, to know you are not alone, I am by almost every measure a very successful member of the community and I have depression.

My plea is that each and even of you, don't ignore the signs and don't ignore the signs in your friends, don't put off getting help as we heard this morning, Daniel arranged to have people from the wellbeing service here present for the next of the few days, make the most. I used a similar service about a year ago and it was remarkable the effect that that had although it didn't get me to do the one thing I should have done which was seek medical advice.

If you have a vague feeling that you are suffering from burn out or depressed, seek help you don't have to feel like this. Denying you have a problem won't make anything better thank you very much.

(APPLAUSE).

DANIELE PROCIDA: Thanks Russell. Thank you Russell for that, for saying that.

Just to remind you, on that back partition there are the post it notes you can pull off to have your appointment with one of the counsellors that are here this morning.

Okay, I challenge you to keep up with this!

Thank you.

NEW SPEAKER: I am here to give one of my favourite stories that is perhaps not well known enough. The story of what the person who is arguably the first software engineer, this is Margaret Hamilton, pictured here with the entire source code of one of the more major programmes she wrote in her career as software engineer, that programme code is the code to Apollo landing modules, now, just take for a second and look. First of all, quite how much paper that is, how it was input but secondly you know, Margaret she was in charge of this project, she was the lead developer and coined the term software engineering after what happened with the Apollo 11 mission. So it is not well known what happened on that mission, of course everyone knows that Neil Armstrong and buzz were in the module, think went down to land and almost ran out of fuel.

One of the things that isn't mentioned outside of software programming circles what happened with the software? Why in fact this mission was almost complete failure apart from Margaret's excellent programming?

Now one of the thing that is on the lunar modules a landing radar, it is difficult ... the landing radar guides the module on to the service, tells you how high you are and it is important for landing. On board there is a small computer that takes the signals, gives the astronauts of how high all sorts.

There is a second radar, this is entirely useless in landing, but trying to dock with the command module. Spinning around the moon who is "never remember my name" but by themselves the computer can handle individually, but can't handle both of them at once. During the landing of Apollo 11, both were turned on, the checklist erroneously had turned on the rendezvous radar, then it showed the craft moving like this and all this caused and overload in the computer core, received about 15% more instructions than it could handle about to shut down and entirely stop giving information, manually fly the module back to orbit.

But thanks to some very sensible coding Margaret Hamilton and her team put priority based scheduling, anticipated the failure, thought about this and the computer rebooted, and ignored everything apart from the high priority landing signals, dropped back to a mode where it ran properly. This is because they knew that obviously when you are writing code for a thing that lands on the moon, you don't get to sit there, there is a bug there and -- it runs once. So what I want to encourage you the do, much like we saw earlier in the security talk is not only think about, but expect failure, your failure, your code will fail. Don't say, proactively go there, find failure and test things, much like Margaret and her team you will write the solution to the problem before it happens thank you. (APPLAUSE).

DANIELE PROCIDA: We have Arthur Johnston and Owen Campbell please to line up?

After that ...

NEW SPEAKER: Okay, this is about a reusable Django app I wrote a couple of weeks ago because I had something to do, I needed to do something along writing a master’s thesis because when you are writing a thesis you get it, pretty much get trained, so you have to have something else to do. I came up with a thing which I noticed on some other sites that when you log into the page you get a notification like a security canary, okay somebody locked in as you and this website, I thought this can't be that hard and famous last words isn't really that hard at all. So, this is just a general admin with what you mentioned before. Works with pretty much with all the lock in things that sense the appropriate signal from the Django provides.

When I type in correct user name but wrong password, well, the log in fails luckily.

But what you can see is, there should be.

NEW SPEAKER: You missed an R.

NEW SPEAKER: Oh yes.

Right. Yes, you can see it now provides some mess am, or write an e-mail to the user who has unsuccessful log in. The user information there, that is kind of not really nice but it works and the other thing is, what is actually helpful is when you type in the correct user name and pass word you get log in information somebody he logged in from this IP address and all of this stuff. A few lines of code. It is I call the app Django log in canary, the magic happens in here, where I connect to this single Django sends and figure out the user and just send a mail.

Yes. You can install the app with Django log in canary, might be helpful for some of you users.

(APPLAUSE).

DANIELE PROCIDA: Daniel, can you come and line up please? Someone ... Owen ready to go? Go on. You need a lap top? You go and I will finish logging in on the lap top.

Owen Campbell you have a minutes.

NEW SPEAKER: Owen. Good afternoon ladies and gentlemen, my name is Owen, what I want to do is to invite you to join our sprint on Thursday and Friday for the Axel Rod. Actually what I want to do to persuade you to come and join us, to encourage you the do so. In order to do that I probably need to explain a bit about what on earth it is about. Very simply, the project is an attempt to create a Django based web app on top of the Axel Rod library, which begs the question, what on earth is the Axel Rod library, why should anyone care enough to want to spend their time and effort helping us to improve it.

To answer that question, I need to wind the clock back a bit. To 1859, the origin of the species, theory of evolution, one of the significant scientific publications ever made, despite its brilliance, he knew at the time that it couldn't answer everything that he could observe in the natural world around him.

For example, it struggled to deal with the fact that around us we see examples of co-operative behaviour. We see individuals within the same species co-operating with one another when survival of the fittest might suggest they should be competing and the winner takes all.

Now that conundrum existed for a long time and still exists for today, let's wind the clock forward a bit further, 1981 published a paper to shed a light on that for the first time. They created a computer simulation of a population and they allowed the members of that population to engage with one another in a very simplified game that was run many, many, many times in what they call the tournament. The members within that tournament could execute all different sorts of strategies and the tournament would show who had succeeded, whose population was growing. They attempted to make a simple simulation of a real world population. But the significant discovery and the papers results were significant because what it showed for the first time was evidence that co-operative behaviour can evolve.

Now, the problem here is that axel rod didn't publish his code. We can't recreate his experiment. So we wind the clock further forward a little. 2014 where Dr Vincent night of the Cardiff school of maths and one of the organisers of this week in fact decided to do something about that. He initiated a project and an open-source project to try to recreate axel rod's ... he did it, so anybody can get involved and see what has been done, can contribute to it and can help take this work a little further. That library is up and running and published on pypy, that is a massive step forward in making this sort of work open to anyone but of course in order to use it you've got to be able to write code, you've got to do a bit of work in Python and that's great for everyone in this room but hardly opens it up to the world so that brings me to the Jackson rod project what we really want to do is put a web application on top of that which will allow anybody to create an Axel Rod tournament to populate it with whatever strategy they can come up with and to see what happens and see the results. We want to allow anybody to interrogate the results of previous tournaments that have happened and analyse them in whatever way they see fit and to do that we need your help. We have an empty Django project sitting there. It's had absolutely no work done on it whatsoever. We will be starting it on Thursday morning. Come along. Say hello. Have a look. Write a few lines of code. You could be contributing to the theory of evolution itself. Thank you very much. {Applause}.

DANIELE PROCIDA: OK Arthur Johnston on scraping with selenium.

NEW SPEAKER: My name is Arthur Johnston. I haven't been doing Python long. I am a PHP developer. Only thing I've done of interest in my life I created the first fully automated PPI system in Britain so if you are still getting phone calls for PPI, that is me. {Laughter}.

I just thought I'd show something that I showed - one of the reasons I love Python is I'm building a new site called welcome dog which lists all locations round the UK you can take your dog, hotels, cottages, and a lot of that can be easily done using scrapey which is a nice bit of kit. Only problem with it is it downloads a dong then scrapes through it so any funky angular stuff or things that react now it doesn't deal well with but something I came across which was interesting the fact you can see there is a site here called pet pyjamas and that's a scrapey script for it which is really short and concise and all you have to do is look up CSS slides and it's easy feed it URLs tell it where the stuff is and it will get it and build you a Json object that sort of thing and I used that in a lot of sites. Then I came across something called trip advisor which is wonderfully designed site 480 K so go and have a butchers but one of the things they still do is this now get of code which scrambles stuff and do it for everything address URL and in this case a phone number which I came across and it was like no I can't do anything with that but when I was doing testing with selenium I suddenly realised it interacts with the browser, it boots up a browser then interacts with it so it's from the user's perspective so it will see all this stuff and can help you in this kind of situation so instead of using selenium to do tests what I do I booted selected to do the CSS elements then instead of going testing making sure there are a couple of countries and so forth instead of finding where those elements are it distracts them so waited until the page loaded put Java script bits and pieces together then we can extract that. I won't read all of it because it will take fore ever and clever stuff as well like web-site button thing if there is a button that says go to web-site. It will boot up but it will boot up to a trip advisor that then says I am redirecting which is a real pain. How you going to get that web-site because it's do steps removed from where you are? And the great thing with selenium you can wait for it to boot to see if that strip of light page, if it is, wait for the next one, grab that one, close it down then carry on with your tests. And then at the end of that you can boot out kick out everything that you've got, your address, your phone number, the web-site and so forth and you can see longitude and latitude as well and boot that out as Json and that by passes - may be a bit criminal I don't know but by passes all of that kind of stuff and yeah you can get to data that is put in obscure ways or if someone is using Java script front ends then you can use selenium to get it to I think it's cool that's why I like Python thanks very much. {applause}.

DANIELE PROCIDA: So, just to let you know, our speech-to-text transcriptors absolutely must leave at 6 o'clock because they've been doing this all day long and they will be completely fried as I think they said if they continue past that. So we will continue with the lightning talks to give everybody a chance. I just apologise to the people who were relying on the speech-to-text transcription. I should have planned a bit better for an extra person so it's entirely my fault. But to Hilary and Sheryll, thank you so much for the huge work you've done today. {Applause}.