0:00 Before we move on from our weather app

0:02 let's look at one other important concept around PyPi, packaging, pip

0:08 and this whole external package management for Python.

0:12 Remember, we could see what was installed by saying pip 3 list

0:16 and over here we have various things like stripe

0:19 1.28.0 for credit card processing.

0:23 And we have pigments for web apps

0:26 now what do you do when you have

0:29 more than one application you want to run on your machine.

0:32 And they both use stripe but the stripe API had

0:36 a breaking change from 1.26 to 1.28

0:40 and one of the apps is written to use the 2.6 API

0:44 and one of your apps is using the 2.8 API

0:46 what do you do, do you just reinstall and uninstall

0:49 and push the versions backwards and forward

0:52 like how do you deal with that version difference, right

0:54 you can't run both apps on your machine at the same time

0:57 without continuously reconfiguring your machine

1:01 also, if I was going to give my app to somebody deploy to production

1:06 give it to a user maybe runs on their machine

1:09 how would I know what I need?

1:11 In this list, what is in this list is required to run the weather app?

1:15 Well, you and I know because we just wrote it

1:17 we know that requests is needed and up at the top bs4 is needed.

1:22 But, there is nothing here that makes it clear what is needed

1:27 now there is ways to do that and with packaging

1:29 and packaging up your apps in certain ways

1:32 but the environment itself does not help us here

1:34 so we have these two problems

1:36 different apps may use different versions of the same package

1:39 and it's very hard to tell what's needed to run a particular app.

1:42 Another problem is if I want to install something here

1:45 I may need to run this as admin, also not the best.

1:49 So there is a way to solve this problem

1:52 and there is an external package that will solve it for all versions of Python

1:55 there is a built in version in Python 3

1:59 that only works in Python 3 called Venv

2:00 and I'll go and show you the way that works

2:04 uniformly across all the different versions.

2:06 So there is something called virtual env that is built to address this problem,

2:10 let's see how we can use virtual env to create a special dedicated environment

2:14 completely isolated from the general machine

2:16 just for this weather app and its packages.

2:19 I made a folder called Python environments,

2:21 let's go in there and see right now it's empty,

2:24 so I am going to use virtual env to actually create environment,

2:27 so one question you might want to ask is which virtual env do you have.

2:31 We have the one from Python 3 so you want to make sure you run that one,

2:35 in this case we need to type virtual env as a program,

2:37 if you want to run the other one

2:40 you might have to run it as a module and through Python directly

2:42 so it's something like Python 3-env space virtualenv.

2:49 Like so, but we don't have to do this,

2:51 we can just type virtualenv, right,

2:55 so what we are going to do now is we want to tell virtual env go and create

2:58 a clean empty Python environment, Python 3, for us to work with.

3:02 So we'll say virtualenv and then we'll just give it a folder name

3:05 so let's call this weather. py3

3:08 just to make it more obvious this is for the weather app and it's Python 3.

3:13 You can see that it's copy the Python 3 executable over and it's also setup pip

3:17 setup tools and few other things.

3:20 So now if I look there should be this folder, if I type correctly,

3:22 there will be this folder, so we can go in there and look and there is a bin folder

3:26 so let's go into this bin, but before I go into the bin

3:31 let me ask the question which Python 3 would I get if I run it

3:35 and let's ask pip3 list.

3:38 There is all the stuff that's in the main machine, ok,

3:41 so let's look in we go in the bin

3:43 you'll see that there is an activate

3:46 and what we are going to do is we are going to run this activate

3:48 and we want to run it against the current shells, so we'll say

3:50 . to source activated

3:52 this isn't necessary in Windows, the dot, it doesn't work, I don't believe,

3:55 we'll say this now watch the command prompt change

3:58 now you see it says weather 3.

4:00 And if I go somewhere else, you see now I know that I have

4:03 my Python 3 weather environment as the active Python environment

4:07 so if I ask questions like which Python

4:09 oh now all of the sudden I've got the one running out of there

4:12 regardless of where I am and if I say pip list

4:15 well we just have an empty system.

4:18 Now the version that got installed into our little virtual environment is out of date

4:21 let's just not worry about that for now.

4:24 Ok, so we can actually use this to run our app.

4:28 So let's go over to where we wrote our weather app say CD and this,

4:34 if we look in here we can see there is our program, that looks familiar, right,

4:38 this is our little weather app, great,

4:41 but if we try to run it we say Python 3 but we don't need to say Python 3

4:44 now there is only one Python in this virtual environment

4:47 and we give it our program, there is going to be a problem.

4:51 It says, this concept of Requests as a module-

4:54 this doesn't mean anything to me

4:56 there is no module named requests

4:59 because in this environment we have nothing

5:02 we just have this clean environment here

5:04 so let's quickly install what we need, we know that we need

5:06 Requests and we need Beautiful Soup 4.

5:10 We can install them both at the same time like this,

5:14 if we don't forget to put the install command in there.

5:20 Excellent, now if we ask pip list you'll see we have

5:23 Request, Beautiful Soup and nothing else

5:26 I am kind of getting tired of this, so let's just run it

5:28 so we have something nice and clean.

5:31 So now if we list, you can see we just have Beautiful Soup, Request

5:34 and the 3 foundational bits.

5:36 Ok, so again, let's try to run this

5:38 we'll say Python and we'll give it the program

5:41 oh look, it's running, all right, 97201 perfect

5:46 the temperature in Portland, Oregon is 7.4 degrees Celsius

5:49 and it's partly cloudy, how awesome is that,

5:52 so it doesn't matter if somebody else installs

5:55 Requests or Beautiful Soup on this machine

5:57 and they have a different version or they updated

6:00 we are a 100% isolated and just have this working version of Python here

6:05 this clean version of Python 3

6:10 and our own user profile in this folder that we created.

6:14 So the last thing to do is use this, in PyCharm.

6:17 So we are over here in PyCharm

6:20 and if you see when I run it that we are still running out of the main python 3.5

6:24 and when I go to the project interpreter,

6:27 you can see that's one we have selected here

6:30 and we have again all the stuff that we saw in Pip list,

6:32 we can actually go over here and select one

6:35 if I hit show all it lets you have this little management thing

6:37 here is an old one I can clean up, then I can add,

6:40 I can say add local or I could even create

6:43 a virtual environment that a whole process you just saw me do with virtual env

6:47 creating this stuff and all that, just click this button and it will actually do that for you.

6:51 So that's super nice,

6:55 but what I want to do is add the one that we just created.

6:58 So we can just browse over here to our environment,

7:01 and it loads up and we just pick Python,

7:06 perfect so now we have this, we can actually select this

7:09 to be the project interpreter for the current project

7:12 and now look what's in here, you can see it's just that clean environment,

7:15 so let's go over here hit ok, it takes a moment for it to update and index the

7:20 environment, now it's ready to run, now look at this,

7:23 user/screencaster/python\_environments/weather\_py3bin/python,

7:28 so now we are using again form PyCharm

7:31 that isolated clean environment, very nice,

7:33 and let's just check one final time what the weather is in

7:36 let's say 92118- beautiful, the temperature

7:39 in lovely Coronado California is 15.2 degrees Celsius and mostly cloudy,

7:44 hey, where is the sun San Diego?

7:46 Anyway, this is working using our environments,

7:49 using our packages that we have installed form PyPi using pip

7:52 into that environment- lovely.

7:55 So that brings this application to a close

7:57 I hope you've learned a lot and had a lot of fun doing it,

8:00 we are just going to re-iterate one more time,

8:02 remember, you should try to use APIs and verify

8:06 the usage rights when you are doing screen scraping

8:09 but it is a really cool way to get data into your app.

0:00 Before we completely step away from our weather app

0:03 let's talk about the final concept virtual environments

0:06 recall, virtual environments allow you to create isolated

0:10 dedicated environments for a particular application

0:13 in there you have a copy of the Python runtime

0:16 and you have initially, basically no packages

0:20 but you can use pip to install those packages

0:23 and they won't conflict or battle with any of the machine wide ones

0:28 you just have only what you need in this nice clean environment.

0:31 It helps you understand how you deploy to production

0:34 it helps you understand what requirements you want to tell other users to have

0:38 they are going to run your scripts on their machines

0:41 now just creating the virtual environment is not enough

0:44 remember, you have to activate it so you change into the environment folder

0:48 and into bin, and you say .space activate on OS X or Linux

0:52 and on Windows you just say activate

0:55 I believe it might be a batch file.

0:57 But, you call activate and then your command prompt changes

0:59 your path changes so that all the Python tools like Python itself

1:03 pip and so on now run exclusively out of that environment.