

Python packaging and science

If you've ever run `pip install`

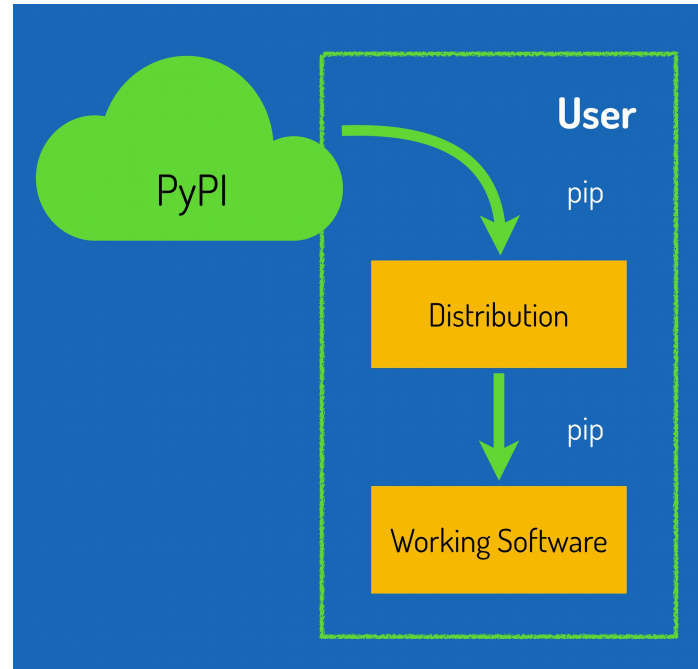
we serve you



(and you are playing on hard mode!)

Video!

pip overview



pip resolver & UX work in 2020

- A stricter, more robust dependency resolver
 - *COULD have been a catastrophe*
- More sustainable maintainership
- Better UX (yes, on the command line)
- Better UX design skills

The most significant changes to the resolver are:

- It will **reduce inconsistency**: it will *no longer install a combination of packages that is mutually inconsistent*. In older versions of pip, it is possible for pip to install a package which does not satisfy the declared requirements of another installed package. For example, in pip 20.0, `pip install "six<1.12" "virtualenv==20.0.2"` does the wrong thing, “successfully” installing `six==1.11`, even though `virtualenv==20.0.2` requires `six>=1.12.0,<2` ([defined here](#)). The new resolver, instead, outright rejects installing anything if it gets that input.
- It will be **stricter** - if you ask pip to install two packages with incompatible requirements, it will refuse (rather than installing a broken combination, like it did in previous versions).

So, if you have been using workarounds to force pip to deal with incompatible or inconsistent requirements combinations, now's a good time to fix the underlying problem in the packages, because pip will be stricter from here on out.

resolver example!

```
pip install "hacking>=1.1.0, <1.2.0"  
"pycodestyle>=2.0.0"
```

Legacy resolver misbehaves; new resolver behaves properly

resolver example!

```
$ pip install --use-deprecated=legacy-resolver 'pyrax==1.9.8'
```

- Errors out with a wall of text.

```
$ pip install 'pyrax==1.9.8'
```

- Errors out with a **more informative** wall of text!

```
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
INFO: pip is looking at multiple versions of oslo-config to determine which version is compatible with other requirements. This could take a while.
Collecting oslo.config!>=3.18.0,<=3.14.0
  Downloading oslo.config-8.3.3-py3-none-any.whl (126 kB)
    |████████████████████| 126 kB 8.5 MB/s
Requirement already satisfied: debtcollector>=1.2.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from oslo.utils>=2.0.0->python-novaclient==2.27.0->pyrax==1.9.8) (2.2.0)
Requirement already satisfied: oslo.i18n>=1.5.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (5.0.1)
Requirement already satisfied: requests<3,>=2.2.1 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (2.25.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
```




```
0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
INFO: This is taking longer than usual. You might need to provide the dependency resolver with stricter constraints to reduce run
time. If you want to abort this run, you can press Ctrl + C to do so. To improve how pip performs, tell us what happened here: ht
tps://pip.pypa.io/surveys/backtracking
  Downloading oslo.config-6.12.0-py2.py3-none-any.whl (131 kB)
    |████████████████████████████████████████| 131 kB 6.9 MB/s
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: debtcollector>=1.2.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from oslo.utils>=2.0.0->python-novaclient==2.27.0->pyrax==1.9.8) (2.2.0)
Requirement already satisfied: requests<3,>=2.2.1 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (2.25.0)
Requirement already satisfied: oslo.i18n>=1.5.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (5.0.1)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
Requirement already satisfied: Babel>=1.3 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from python-novaclient==2.27.0->pyrax==1.9.8) (2.9.0)
Requirement already satisfied: six<2,>=1.9.0 in ./virtualenvs/czi-test-1/lib/python3.7/site-packages (from pyrax==1.9.8) (1.15.0)
```



Helpful output

INFO: pip is looking at multiple versions of REDACTEDPACKAGE to determine which version is compatible with other requirements. This could take a while.

Downloading REDACTEDPACKAGE-0.3.2.tar.gz (21 kB)

Downloading REDACTEDPACKAGE-0.3.1.tar.gz (20 kB)

Downloading REDACTEDPACKAGE-0.3.0.tar.gz (20 kB)

Downloading REDACTEDPACKAGE-0.2.0.tar.gz (19 kB)

Downloading REDACTEDPACKAGE-0.1.0.tar.gz (18 kB)

INFO: This is taking longer than usual. You might need to provide the dependency resolver with stricter constraints to reduce runtime. If you want to abort this run, you can press Ctrl + C to do so. To improve how pip performs, tell us what happened here: <https://pip.pypa.io/surveys/backtracking>

Helpful output

Collecting git+https://github.com/fiendish/xlrd

Cloning https://github.com/fiendish/xlrd to /private/var/folders/7r/zzgsf_917vn98xy97_274d24z3zxz5/T/pip-req-build-z3za5_mk

Collecting git+https://****@github.com/fiendish/xlrd

Cloning https://****@github.com/fiendish/xlrd to /private/var/folders/7r/zzgsf_917vn98xy97_274d24z3zxz5/T/pip-req-build-6x7m6gbw

ERROR: Cannot install xlrd 1.2.0 (from git+https://****@github.com/fiendish/xlrd) and xlrd 1.2.0 (from git+https://github.com/fiendish/xlrd) because these package versions have conflicting dependencies.

The conflict is caused by:

The user requested xlrd 1.2.0 (from git+https://github.com/fiendish/xlrd)

The user requested xlrd 1.2.0 (from git+https://****@github.com/fiendish/xlrd)

To fix this you could try to:

1. loosen the range of package versions you've specified
2. remove package versions to allow pip attempt to solve the dependency conflict

ERROR: ResolutionImpossible: for help visit https://pip.pypa.io/en/latest/user_guide/#fixing-conflicting-dependencies

This also means that, when you run a `pip install` command, pip only considers the packages you are installing in that command, and **may break already-installed packages**. It will not guarantee that your environment will be consistent all the time. If you `pip install x` and then `pip install y`, it's possible that the version of `y` you get will be different than it would be if you had run `pip install x y` in a single command. We are considering changing this behavior (per [#7744](#)) and would like your thoughts on what pip's behavior should be; please answer [our survey on upgrades that create conflicts](#).

We are also changing our support for [Constraints Files](#), editable installs, and related functionality. We did a fairly comprehensive overhaul and stripped constraints files down to being purely a way to specify global (version) limits for packages, and so some combinations that used to be allowed will now cause errors. Specifically:

- Constraints don't override the existing requirements; they simply constrain what versions are visible as input to the resolver (see [#9020](#))
- Providing an editable requirement (`-e .`) does not cause pip to ignore version specifiers or constraints (see [#8076](#)), and if you have a conflict between a pinned requirement and a local directory then pip will indicate that it cannot find a version satisfying both (see [#8307](#))
- Hash-checking mode requires that all requirements are specified as a `==` match on a version and may not work well in combination with constraints (see [#9020](#) and [#8792](#))
- If necessary to satisfy constraints, pip will happily reinstall packages, upgrading or downgrading, without needing any additional command-line options (see [#8115](#) and [Options that control the installation process](#))
- Unnamed requirements are not allowed as constraints (see [#6628](#) and [#8210](#))
- Links are not allowed as constraints (see [#8253](#))
- Constraints cannot have extras (see [#6628](#))

Per our [Python 2 Support](#) policy, pip 20.3 users who are using Python 2 will use the legacy resolver by default. Python 2 users should upgrade to Python 3 as soon as possible, since in pip 21.0 in January 2021, pip will drop support for Python 2 altogether.

Colleagues who are here

Georgia and Bernard:

UX testing/research



Pradyun:

Python package mgmt

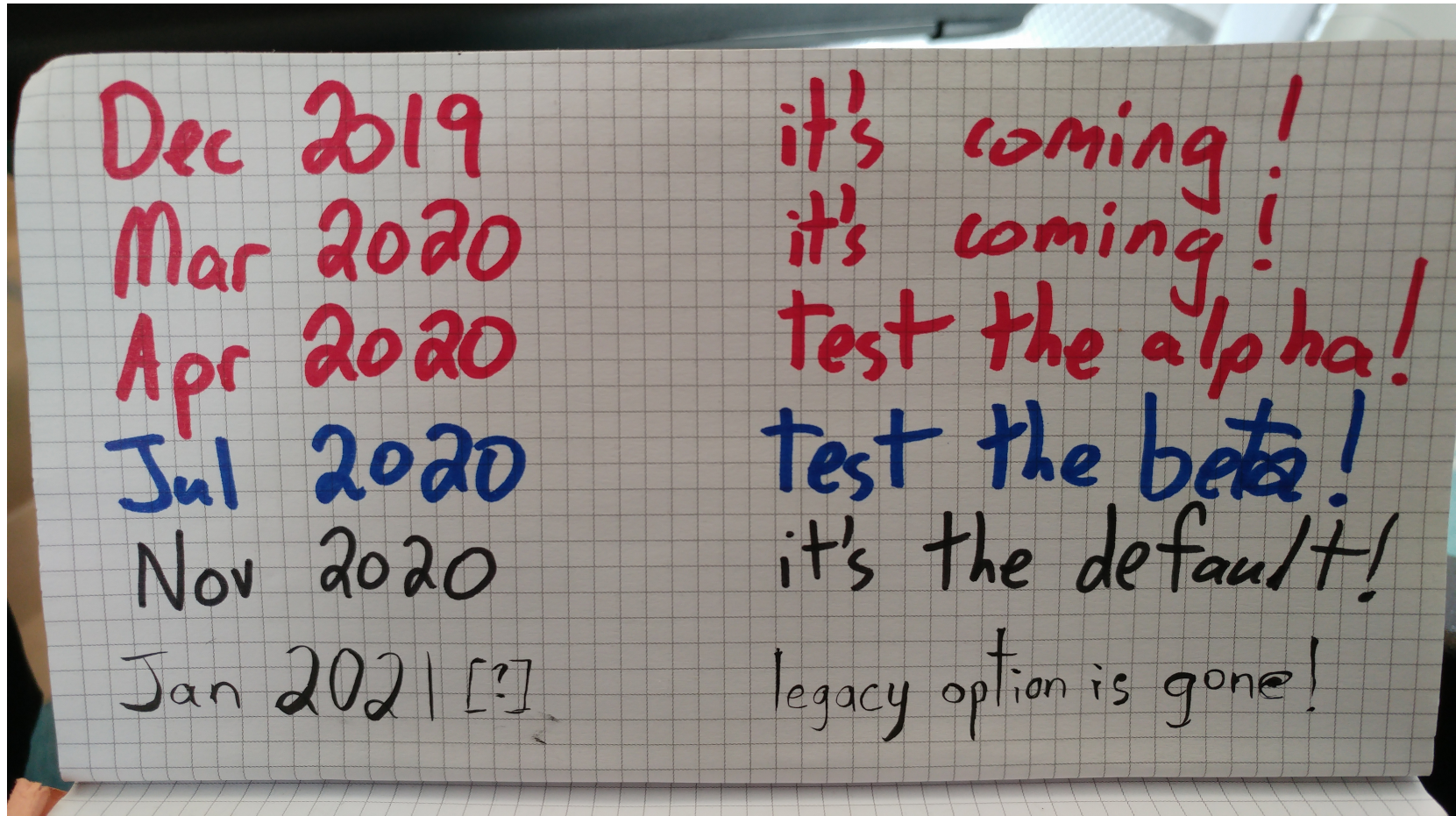


Sumana:

Project mgmt



Rollout schedule



A photograph of a piece of white graph paper with handwritten notes in red and blue ink. The notes are organized into two columns. The left column lists dates from December 2019 to January 2021. The right column contains corresponding status updates and actions for each date.

| | |
|--------------|------------------------|
| Dec 2019 | it's coming! |
| Mar 2020 | it's coming! |
| Apr 2020 | test the alpha! |
| Jul 2020 | test the beta! |
| Nov 2020 | it's the default! |
| Jan 2021 [?] | legacy option is gone! |

Monday, March 23, 2020

New pip resolver to roll out this year

The developers of pip are in the process of developing a new resolver for pip (as we announced on the [PSF blog](#) last year). We aim to roll it out later this year. (*Updated 29 September* to add: [see our migration guide](#).) As part of that work, there will be some major changes to how pip determines what to install, based on package requirements. In this post we share:

What will change

What you can do to help

When this will be happening

To understand what pip does and why we're working on it, please read [our earlier overview post](#).

What will change

The most significant changes to the resolver will be:

Since we did know ahead of time that it is hard to get users' attention about infrastructure that they depend on, and we recognized that many people assume that they do not need to keep apprised of changes coming in pip, we did publicity and outreach (some of which is catalogued in [#8511](#) or [this wiki page](#)), such as:

- A message in pip output saying:

ERROR: After October 2020 you may experience errors when installing or updating packages. This is because pip will change the way that it resolves dependency conflicts.

We recommend you use `--use-feature=2020-resolver` to test your packages with the new resolver before it becomes the default.

- Emails to PyCon, PSF, and PyPI sponsors from their PSF sponsorship contact, asking them to tell their engineering departments about coming changes
- [Email to pypi-announce](#), [python-announce](#), and similar lists
- Podcasts: [Podcast.__init__](#), [Test & Code](#), [FLOSS Weekly](#), [Software Developers Journey](#), [Real Python](#)
- A bunch of [tweets](#), many of which were retweeted by the @`ThePSF (Python Software Foundation) account
- New GitHub issues and comments on existing issues for some projects that had already linked to [#988](#) when mentioning their problems, plus a few new issues for projects that I chose by looking at some lists of popular/widely used packages
- [A short YouTube video about the changes](#)
- Blog posts [like this one](#), on [the Python blog](#) and on [the PSF blog](#)
- Emails, Slack and Discord posts to NumFOCUS's mailing list, Python user groups, civic tech and similar groups

(The announcements of future changes also attracted comment on Hacker News, on the r/python and related subreddits, and elsewhere on news websites in Japanese, Chinese, Russian, and other languages that I can only read via translation services.)

But of course there is room to improve, which is why I ask: what information sources do you pay attention to about software

User research (examples)

Getting support with pip

Most participants said they needed to get support with pip at some point. The mechanisms used were all very similar - there was a lot of overlap.

By popularity support mechanisms were:

1. Search for it, and land on a Stack Overflow article
2. Search for it and land on a blogpost, article written by someone who had the issue previously
3. Go to pip documentation
4. Ask their colleagues, friends, in person or via online communication platforms

A very small number of participants said they never had issues with pip, and so wouldn't know where to look.

- Providing the operators in the documentation was helpful
 - some felt they were "in the way" of the solution
- Experienced Python users saw recommendation for forking package (solution 2) seen as:
 - unnecessary by experienced Python users
 - unhelpful and possibly dangerous for unexperienced users
- The error causes were verbose but helpful
- Experienced participants thought link for Dependency Hell was useful (and funny)
- Force install information seen as confusing
 - Why can't I do it?
 - This might be something I wanted to try
 - If I can't do it, why tell me about it?

Findings

Error message

- high-level findings
- detailed findings

Documentation

- high-level findings
- detailed findings

Error message

High-level findings

1. Users did not understand the meaning of `~='`
2. The possible solution documentation link was lost in the error message
3. Most users had no understanding of `Error: no distribution found` message
4. Format of constraints was difficult to read
5. Some experienced users asked for concrete actions to be displayed in the error message
6. Use of red text made error message reduced readability

Detailed findings

1. Users did not understand the meaning of `~='`
- 15 of 17 did not understand this
 - This clouds understanding of the issue meaning they could not identify the right troubleshooting path
 - This was across all-levels of Python experience and profession

What's next?

- When do we rip out legacy resolver?
- UX training for packaging tools maintainers
- We COULD work on cool features
 - `pip upgrade --all`
 - Better support for reproducible environments
 - but we are out of money
 - <https://github.com/psf/fundable-packaging-improvements/>

Thanks

Thanks, CZI & Mozilla

Contact us:

<https://discuss.python.org/c/packaging>

<https://github.com/pypa/pip/>