I Finished the Beginner Tutorial, Now What?

Kelsey Karin Hawley
@kkarinhawley

These slides at https://github.com/khawley/my-talks

About Me

- Software Engineer at Honor (joinhonor.com)
- CS degree from UC Santa Cruz
- Taught at Girls Who Code this summer
- Currently a mentor at Hackbright
- Avid reader and crafter

Yes we're hiring!

This Talk

- 3 Types of Learning Approaches
  - Challenge Based Learning
  - Example Based Learning
  - Project based Learning
- Community
- Summary

Won't Cover

Won't Cover

- Bootcamps, code schools
- College or Online Courses
- More tutorials

Challenge Based Learning

-- 1:00

What Do I Mean?

Using multiple practice based exercises to grow your knowledge in small tangible ways

Appealing to those who:

- Don't want a big project
- Don't have small ideas to work on
- Like Rewards + Trophies
- Want thoroughly covered edge cases and learning tools

-- 1:00

Tools

- 99 Problems
- CodeChef
- CodeCombat
- CodeEval
- CodeFights
- CodeForces
- CodeKata
- CoderByte
- CodeWars
- CodinGame
- DevDraft
- Exercism.io
- HackerRank
- HackerEarth
- LeetCode
- Programmr
- Project Euler
- Rosalind
- Spoj
- Timus
- TopCoder

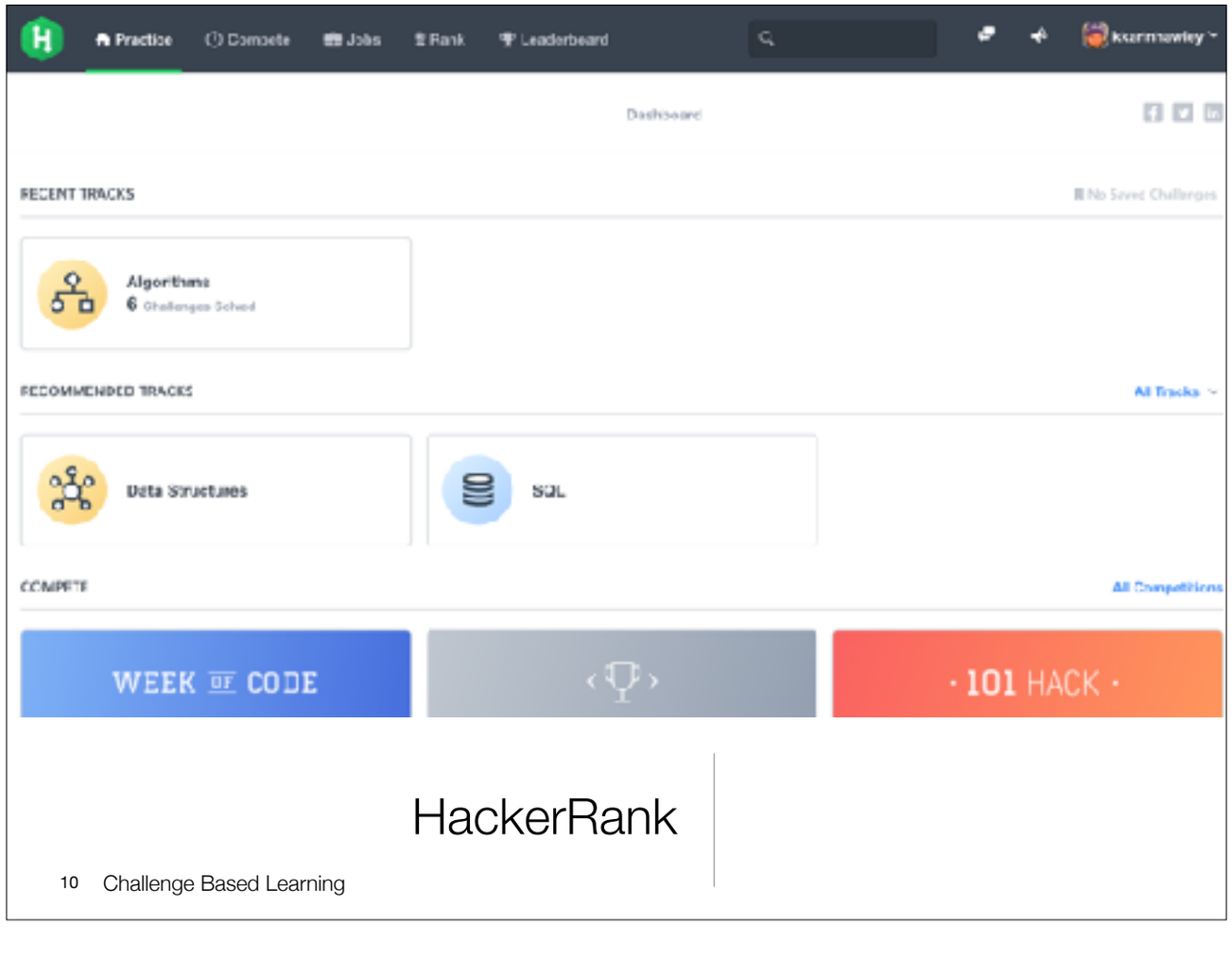
Lots of tools out there - just a quick google search, and definitely not all inclusive set

Choosing a Tool

Think about the features that matter to you:

- Language agnostic
- Strong solution testing
- In-browser interpreter
- Badges
- Reviews
- Competitions

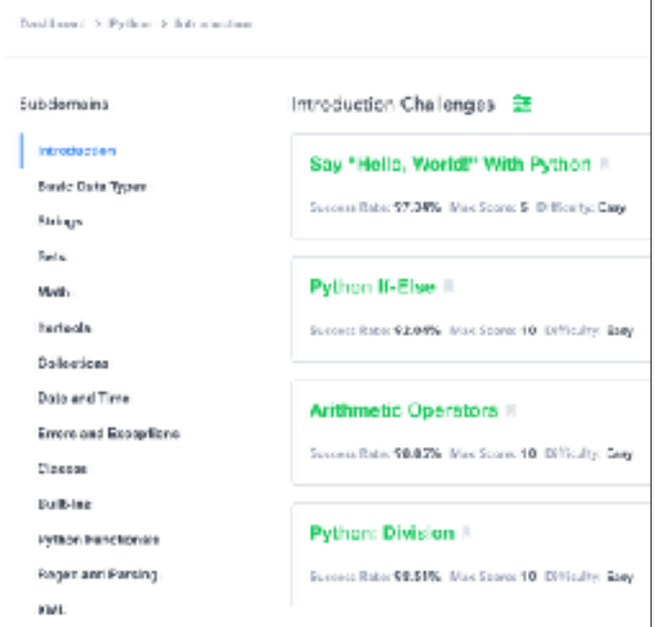
Language Agnostic - Will you be learning other languages down the road?
Solution Testing - really important for knowing when you're wrong or close but missing something
In-browser interpreter - if you travel a lot or don't have a dedicated computer this might be appealing
Badges or rewards if you are goal/award driven
Reviews - Feedback from your peers can help you grow
Competitions - good if you want to compete for prizes or need the motivation competition provides


Practice
Compete
Jobs
Rank
Leaderboard
ksarmarkey
Dashboard
RECENT TRACKS
No Saved Challenges
Algorithms
6 Challenges Solved
RECOMMENDED TRACKS
All Tracks
Data Structures
SQL
COMPETE
All Competitions
WEEK OF CODE
101 HACK
HackerRank

-- 3:15

HackerRank - Pro's

- Good diversity of problems + languages
- Challenges start from easy and go to advanced



-- 3:30
Python 2 + 3 compatible

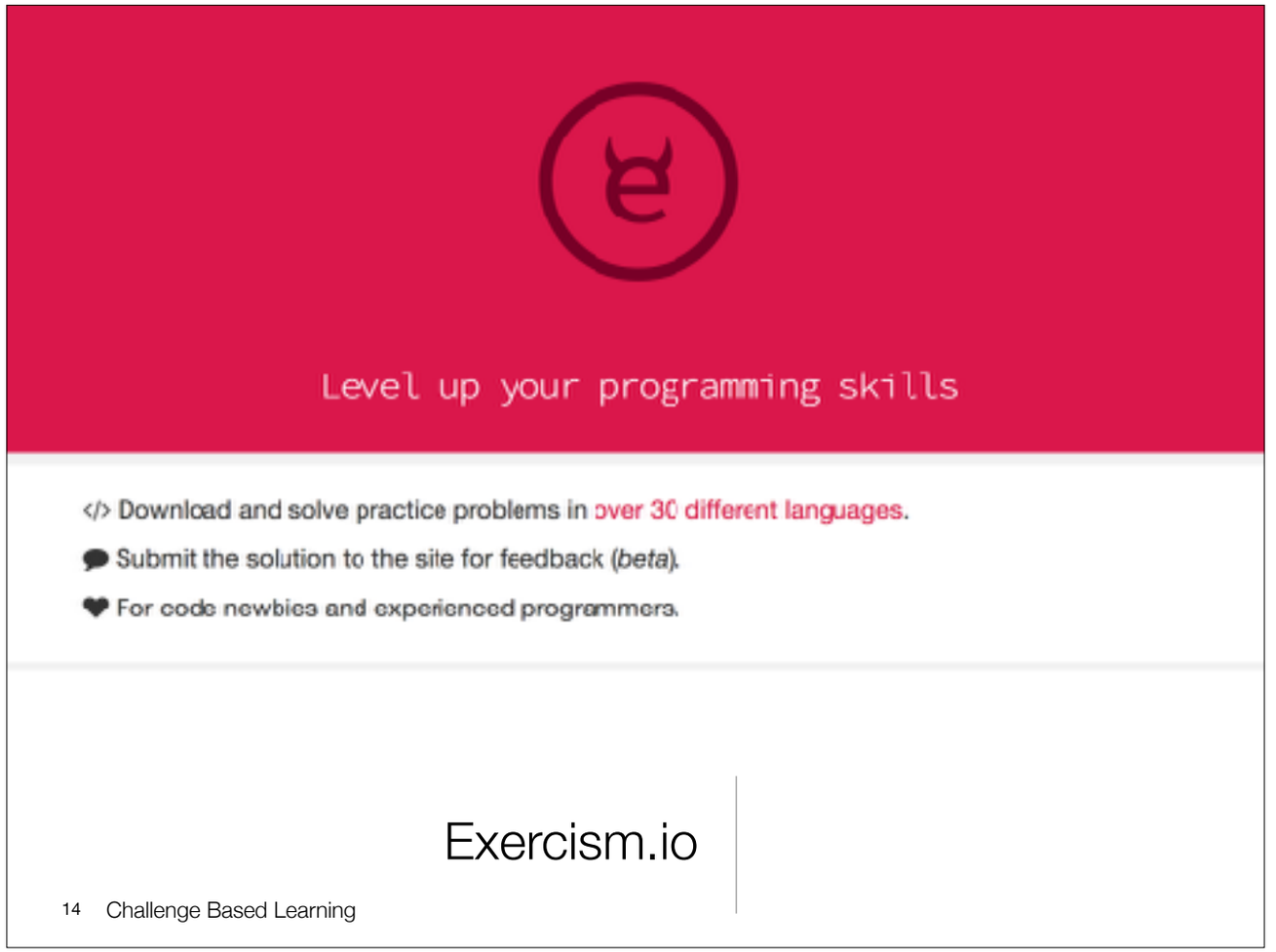
HackerRank - Pro's

- Good in browser editor
- Tests run in real-time
- Error messages

Error messages are shown for each failed test case - could be taking too long, could be missing an edge case, etc

HackerRank - Con's

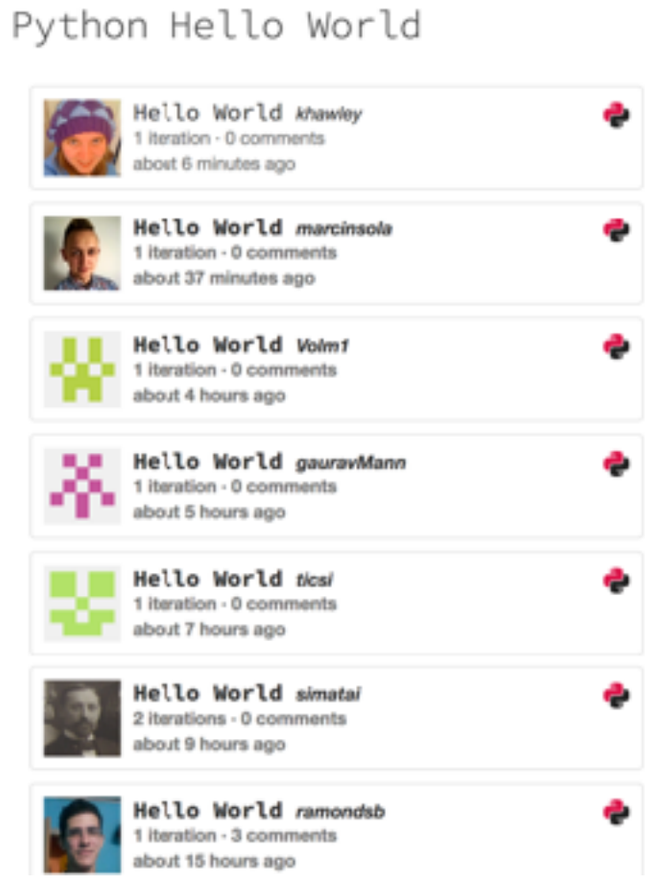
- No community feedback nor interaction
- No way to work and test offline


Level up your programming skills
</> Download and solve practice problems in over 30 different languages.
Submit the solution to the site for feedback (beta).
For code newbies and experienced programmers.
Exercism.io

---5:30

Exercism.io - Pro's

- Able to review other users' code
- Able to leave and receive comments on code
- Can join a team
- Community driven open source exercises



--- 5:30
Supports python 2 + 3

Exercism.io - Con's

```
» exercism submit hello_world.py
Your python solution for hello-world has been submitted.

Programmers generally spend far more time reading code than writing it.
To benefit the most from this exercise, find 3 or more submissions that you can
learn something from, have questions about, or have suggestions for.
Post your thoughts and questions in the comments, and start a discussion.
Consider revising your solution to incorporate what you learn.

Yours and others' solutions to this problem:
http://exercism.io/tracks/python/exercises/hello-world
```

- Requires some knowledge of the command line
- Relies on editor + terminal on your machine
- Must complete an exercise to receive the next

More Tools

- 99 Problems
- CodeChef
- CodeCombat
- CodeEval
- CodeFights
- CodeForces
- CodeKata
- CoderByte
- CodeWars
- CodinGame
- DevDraft
- HackerEarth
- LeetCode
- Programmr
- Project Euler
- Rosalind
- Spoj
- Timus
- TopCoder

--6:45
Lots more out there to explore
Find one that works for you
Look for one that is active and updated

Summary

- Great way to build incrementally on a new language
- Built-in tests tell you when you're missing something
- Built-in community can cheer you on and offer guidance
- May include a job board or competitions for prizes

Just be sure to pick one that appeals to you

Example Based Learning

-- 7:00

What Do I Mean?

Reading through someone else's code to gain knowledge or using someone else's code as a starting point

Appealing to those who:

- Need a starting point
- Want to see how others write code
- Want to see abstract concepts in use
- Don't want to write code from scratch
- Like reading and reviewing other's work to grow skills

-- 7:00

Finding a project

Need some comfort with GitHub and Git

What to look for:

- Readme/setup instructions
- Under active development
- Contributor is responsive
- The code easy to read and follow along with
- Able to google for foreign terms being used

First you'll need a basic understanding of using git and github, as this will involve finding projects through github and likely cloning them. If you don't feel comfortable navigating and cloning projects yet, I'll have a link to a few tutorials at the end

Finding a project

What to avoid:

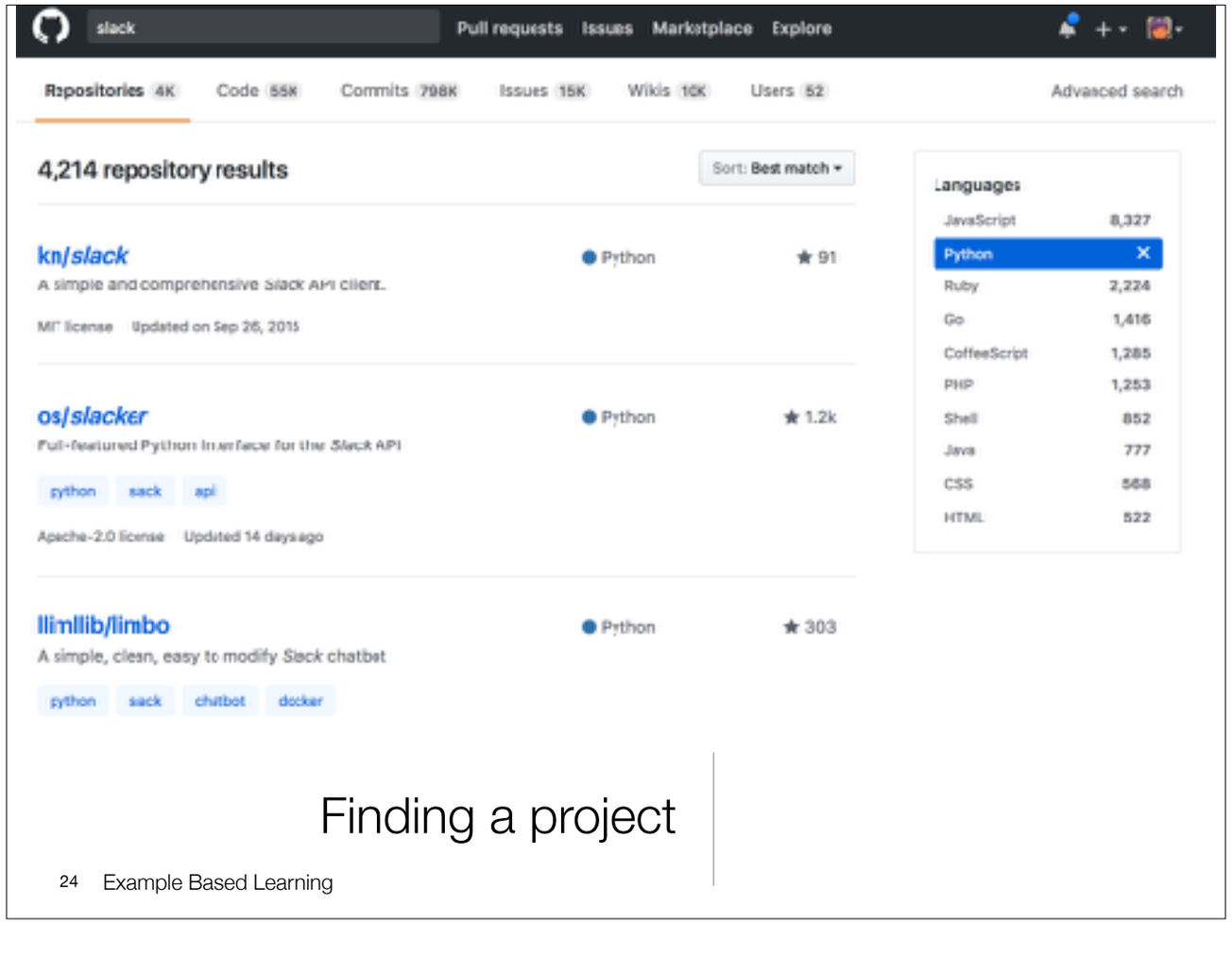
- It uses a lot of outside packages that you don't understand
- Its overly complex
- You're having trouble building or running based on the Readme instructions
- Its not clear where you would even start reading the code

- It uses a lot of outside packages that you don't understand
- Overly complex - opening 5+ files to follow the logic

Finding a project

Tips for searching:

- Lots of open repositories to explore on GitHub
- What tools do you use? Do they have anything on GitHub that you could explore?
- Friends or engineers you admire on GitHub? Explore their repos
- Find one that is *interesting to you*, as you'll be spending lots of time reading and working with it



Finding a project

Oh look a slack bot

Reading Code

Reading through someone's code will take some time

1. What their code is trying to do
2. What does it require to get there
3. What are the lines of code actually doing

Macro - Understand what their code is trying to do:

Is it a bot? Is it a command line tool? Is it a webapp?

Understand what they require to get there:

What are they importing? What did they install from 3rd party packages?

Micro - Understand what that code is actually doing:

What is this function specifically doing? What functions is it using from somewhere else?

Dig In

- Why did they write it this way? Could it be written it differently?
- What happens if you change a variable value?
- What happens if you change part of a function?

Once you've got a basic understanding and have it running on your machine, try messing with it.

```
» python
Python 2.7.9 (default, Sep 12 2017, 10:44:58)
[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.42)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> my_set = set([1,4,5,6,9])
>>> my_set[0]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'set' object does not support indexing
>>>
```

Dig In

Use error messages to your advantage

You might find the developer left custom error messages. Or you might get lengthy python stack traces to play with.

Remember: You aren't breaking their code by experimenting. You are safe to frollick and cavort creating chaos and errors. Use this as an opportunity to learn!

Dig In - Next Steps

- Are there open bugs that you could fix?
- Can you add a new feature?

Take-aways

- New 3rd party packages?
- Exposure to a builtin you hadn't used before?
- New concepts you hadn't seen?

Python package - requests
Builtins - like sets or filtering lists, d
Concepts - decorators, generators

Summary

- Find a repository that you can follow along with
- Ask questions of the code
- Don't be afraid to tinker
- Use what you learned

Project Based Learning

-- 14:30

What Do I Mean?

Building your own project from scratch

Appealing to those who:

- Already have a project in mind
- Can't find a good example project on GitHub
- Are self-motivated and goal oriented
- Creating a portfolio

-- 14:30
Some people learn best by diving into their own projects
Like creating goals, expectations, or doing project planning

Project Management

- Beware of Scope Creep

My biggest piece of advice

Project Management

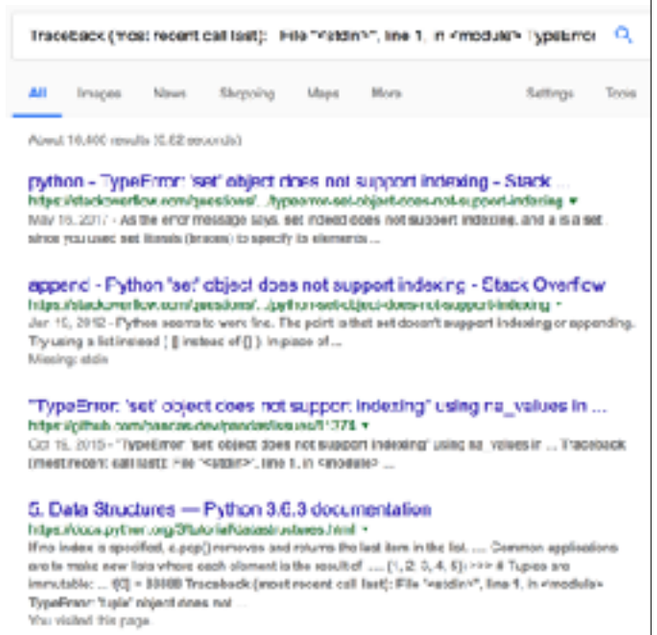
- Map out all the project to-do's and break them down
- Focus on small, manageable pieces
- Break things out further if you're getting discouraged
- Celebrate each success

- To-do's can include anything: research, experimentation, coding, documentation, setup, tests

Transition: When you find yourself getting stuck...

Google is your friend

- Utilize StackOverflow
- Copy & Paste error messages right into search



StackOverflow for questions, for errors, for "how do I ___ ?"

My Passion Project

-- 17:30

Castle Dice

- Castle Dice has *a lot* of components
- Each component had multiple interactions with its surrounding components
- There were pieces to track pieces to track pieces...
- Complexity was high

-- 17:30

Castle Dice

Breaking it down:

- 5 types of dice
  - Each die had unique sides. How could I store & access them?
  - How could I 'roll' a die to get a random result?
  - Sometimes dice are added together. How could I keep that information?
- 2 unique decks
- ...

I broke down each component into even smaller pieces
First I built a die roller. How would that even work?
Then I determined how I could store the 'sides' and change my die roller to roll these unique sides.
Etc

Castle Dice

So, is it done?

Castle Dice

So, is it done?

No. But I tinker on it every few months and make small improvements as time allows.

Castle Dice

My takeaways:

- This was a really big project.
- I should have chosen something smaller
- There were a lot of unknowns I didn't consider.
- There are a lot of knowns I still don't know how to approach
- I learned a lot
- I had fun
- It doesn't matter whether I 'completed' it

My lessons -

Summary

- Try to choose a small, manageable project
- Break it into even smaller tasks
- Ask for help when you need it
- StackOverflow and Google are your friends
- Everything is "work-in-progress"
- Have fun

I still consider Castle Dice a work-in-progress, not an abandoned project

Community

-- 20:30

Why Community Matters

- Community groups are actively developing in their language and tools.
- Members can offer advice or resources

The previous styles won't teach you to code, only give you ways to practice.

Members can help you with your projects, connect you with other beginners, provide you with even more resources.

Mentors

- Both help and encourage learners.
- Have tips for overcoming tricky concepts.
- Teach you how to think about a problem, without just giving you the answer.
- Care about your growth

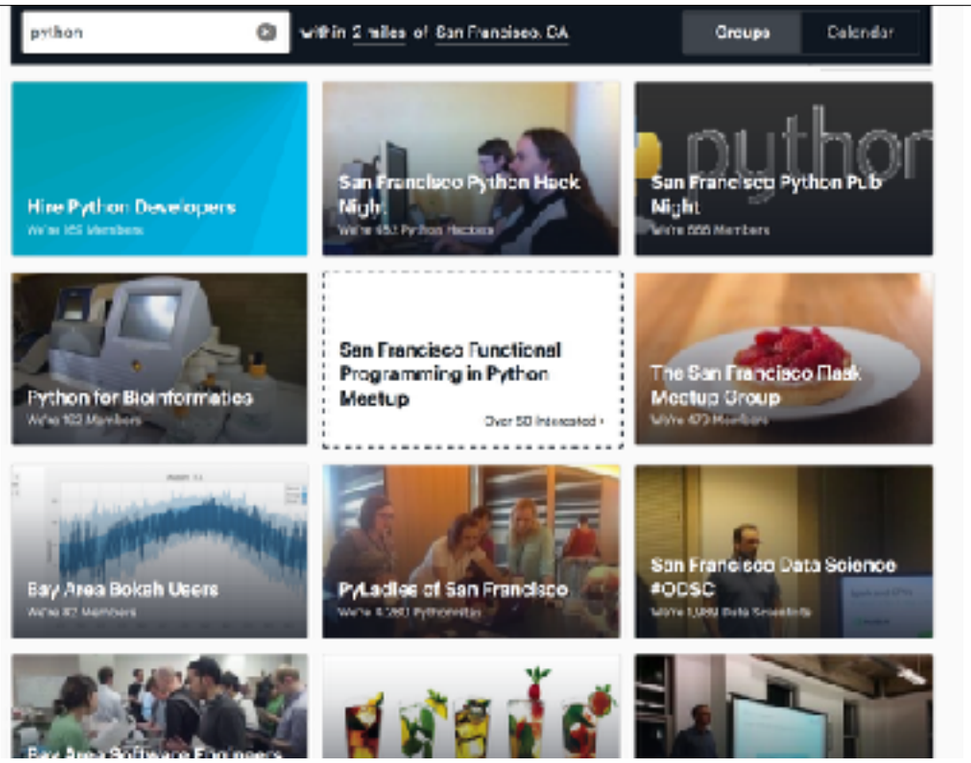
One of the most valuable things you can get from interacting with coding communities are mentors.

Mentors - super valuable
- help + encourage
- Were beginners once + can sympathize with pains/pitfalls of a language. They can also share tips/tricks they've learned
- Good mentor will coach you to the solution, not just give you the answer
- Will want to share in your successes

Python specific

- PyLadies
- SF Python Meetup
- SF Django Meetup
- PyCon
- NorthBayPython
- PyBay

Mix of conferences & groups that focus on Python.



Meetup.com

Even more by searching your city + Python online

Other languages

- Waffle.js
- RailsBridge
- HackerSpace

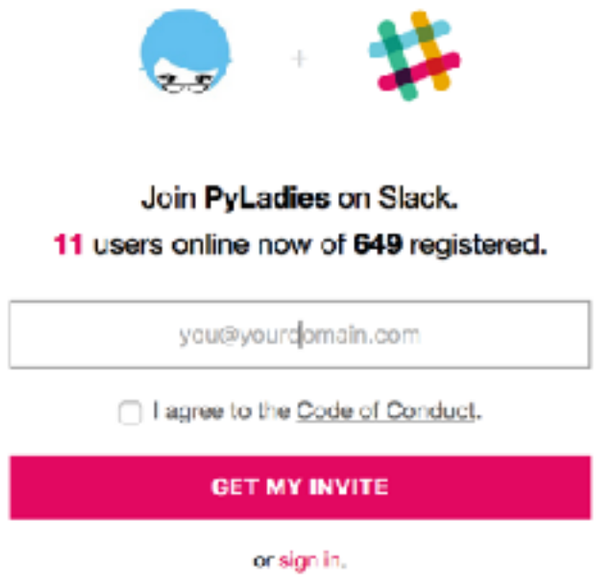
Also Donut.js in Portland

For the Gals

- PyLadies
- Girl Geek Dinners
- Women Who Code
- Girl Develop It

Online

- Many groups have chat equivalents online too
- Exercism.io & others have built-in communities
- Coding communities



CodeBuddies new to me, but seems to be a way to connect online with fellow coders/learners

Summary

-- 23:30

Summary

- Challenge Based
- Example Based
- Project Based
- Community
- There's no one way to continue learning and growing

Challenge based:
- Practicing with small exercises
- Test cases, Community feedback

Example based:
- Find a repository that is interesting
- Ask questions of the code
- Try and break it, then fix it

Project based:
- Break down any project idea into small pieces
- Have fun

Community - Lots of ways to connect
- Groups, Tech Events Conferences
- Mentor

Resources

- Me - @kkarinhawley, Honor (joinhonor.com)
- These slides at https://github.com/khawley/my-talks
- Challenge based
  - HackerRank.com
  - Exercism.io
- Example based
  - Git/GitHub Tutorials
    - My Gitbook - https://www.gitbook.com/book/khawley/git-github-basics-tutorial/details
    - Atlassian's tutorial - http://www.atlassian.com/git/tutorials/what-is-version-control
- Project based
  - CastleDice, the never ending - https://github.com/khawley/CastleDice