Fluency

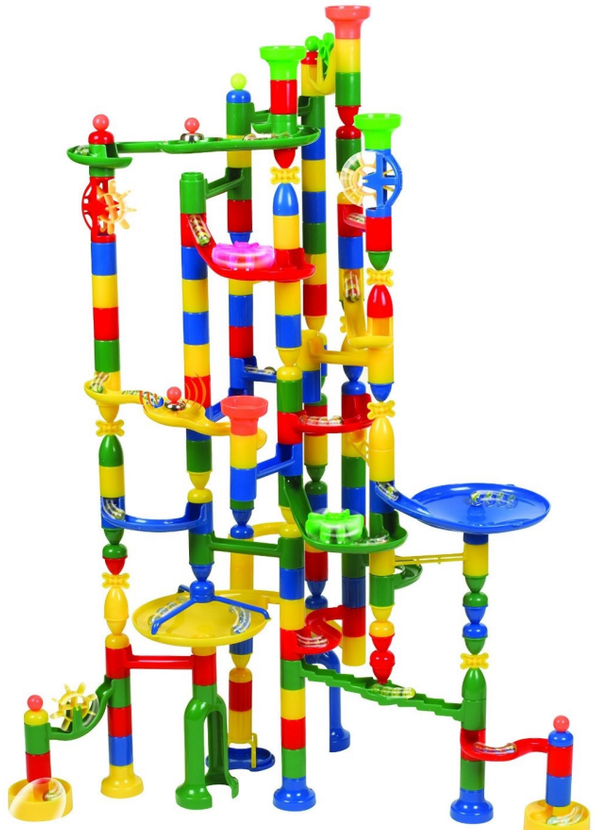
Words → Grammar

Today: Composition • “Combinators”

Assignment: APIs, syntax, operations, features

Is this a DSL?

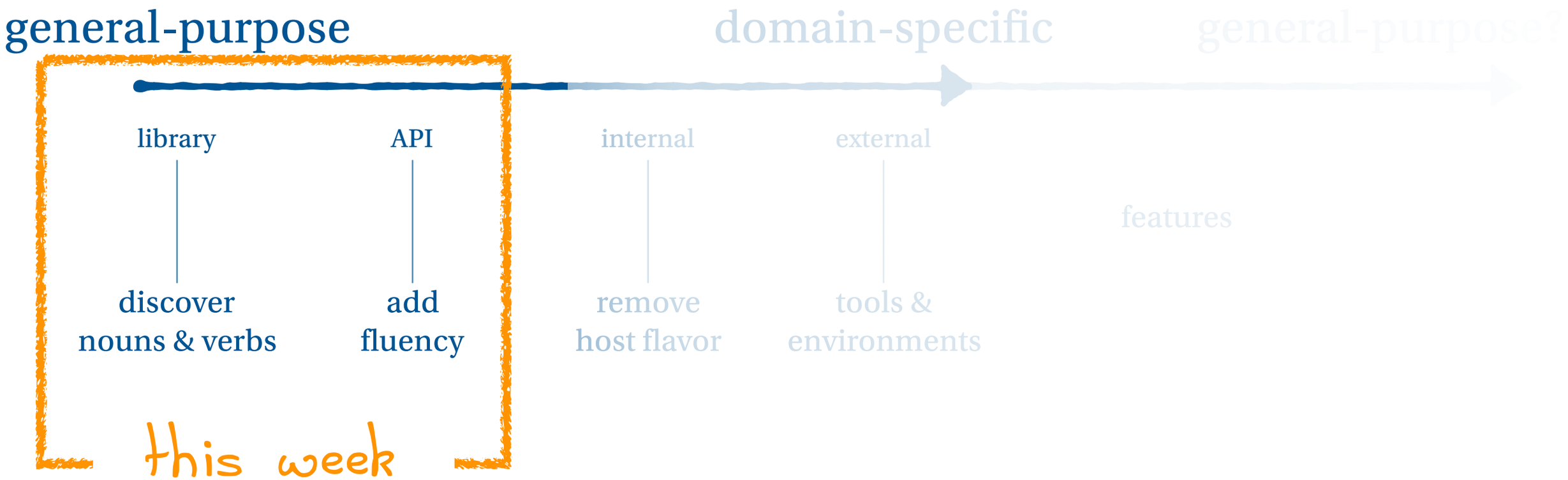
Marbulous



We should have good answers for all these questions

- Can it satisfy our definition of a programming language?
- What does a program in this language look like?
- What happens when a program runs?
- What should be easy in this language?
- What should be difficult?
- What should be impossible?

The evolution of a DSL?



Sound lab

work until 3:40pm

Fork the repository and submit a pull request soon after

It should be easy for users of your library to:

- Modify an existing sound file by reversing it and making it quieter.
- Play the resulting sound.
- Save that sound to a file.

```
$ python
Python 2.7.10 (default, Jun 10 2015, 19:42:47)
[GCC 4.2.1 Compatible Apple LLVM 6.1.0 (clang-602.0.53)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import sounds
>>> sounds.reverse('spam.wav')
>>> # edit sounds.py to make changes
>>> reload(sounds)
<module 'sounds' from 'sounds.pyc'>
```

Fluency: let’s talk

How did the original design make it harder for users to play with sounds?

How did you change the design to make it easier for users?

Programming as language design?

HW 1: Language design

Work in pairs

Only one partner needs to fork, then add the other partner to the fork

Readings about language design

Mini-essays that respond to the reading

several questions, ~500-word responses per question

Available later today (I'll announce on Piazza)

Choose your partner

Growing a Language (the paper, not the talk)