

Fluency

Today's theme:
programming as
language design

Is this a DSL?

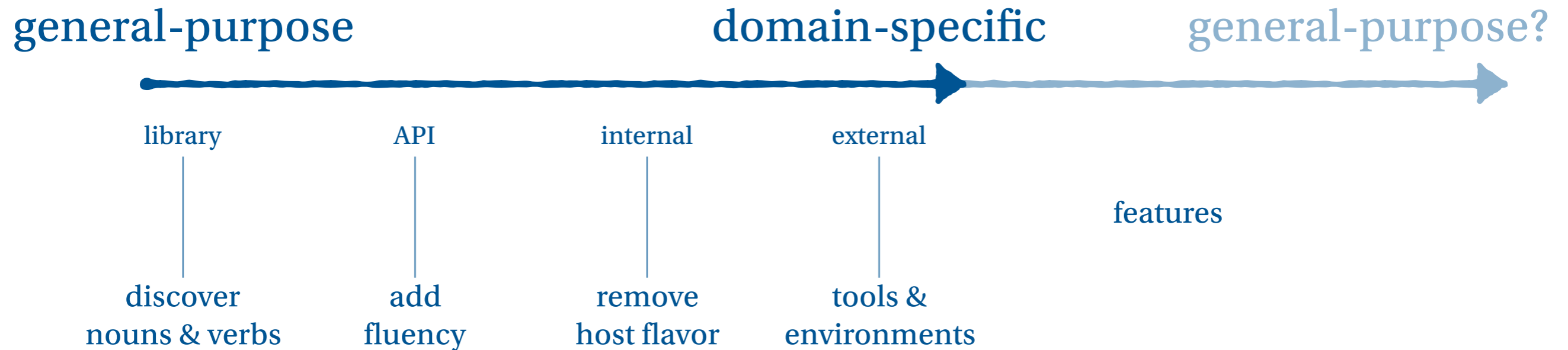
Marbulous



We should have good answers for all these questions

1. Can it satisfy our definition of a **programming language**?
2. What does a program in this language **look** like? (syntax)
3. What happens when a program **runs**? (semantics)
4. What should be **easy, difficult, impossible** in this language?

The evolution of a DSL?



We are here

Sound lab

Fork the repository and submit a pull request immediately 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.

```
def double(n):  
    return n * 2
```

```
values = [1,2,3]
```

```
map(double, values)  
filter(lambda x: x % 2 == 0, values)
```

```
# List comprehensions
```

```
[v * (v+1) for v in values]  
[v / 2 for v in values if v % 2 == 0]
```

Python
quick-reference

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
assignment includes recommendations for the writing

Available later today (I'll announce on Piazza)

Choose your partner

Growing a Language (the paper, not the talk)