SWIM

The Team:

Morris Hopkins - Project Manager
Seungwoo Lee - Language Guru
Lev Brie - System Architect
Alexandros Sigaras - System Integrator
Michal Wolski - Verification and Validation

WHAT IS SWIM?

+

+

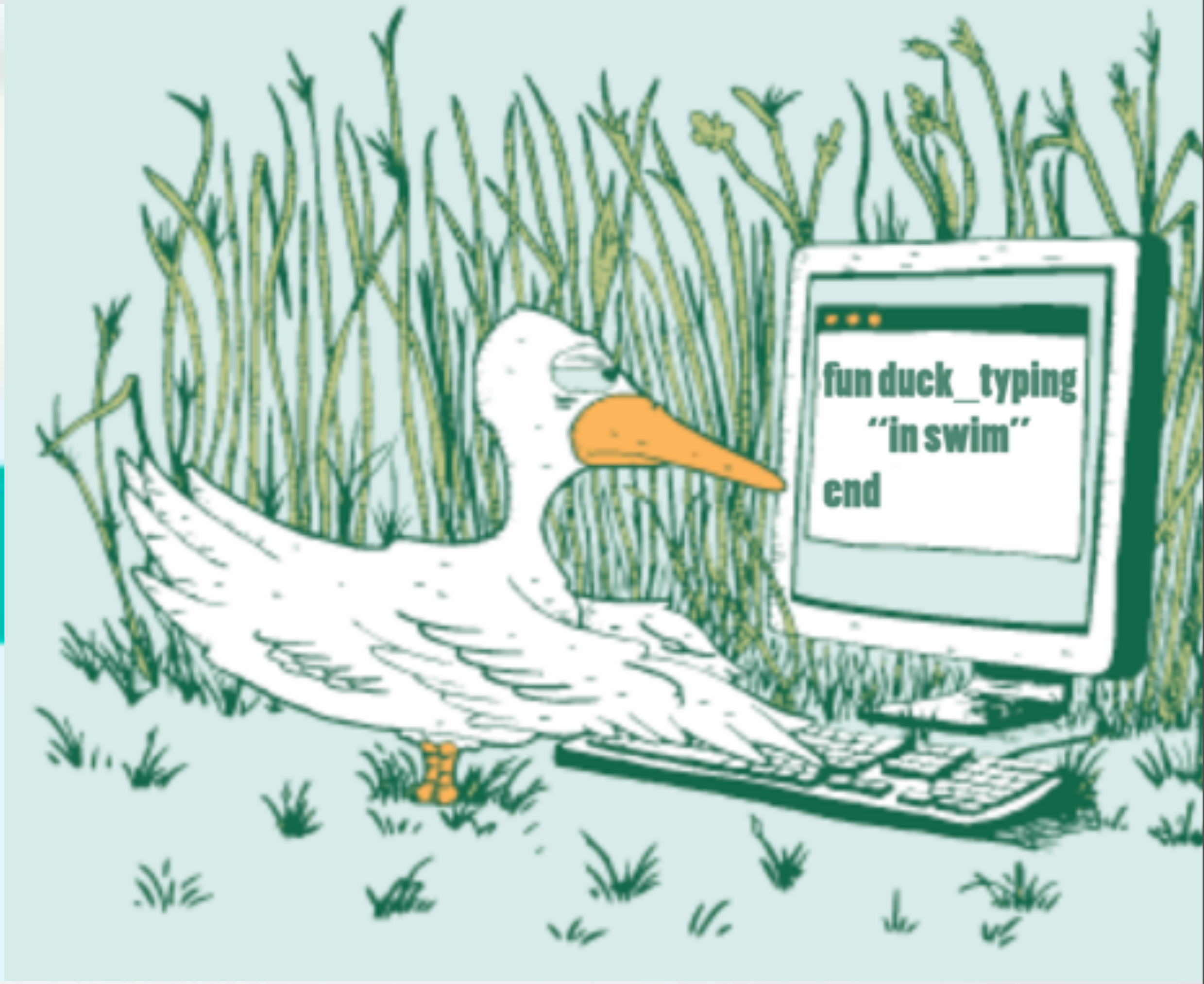
Easy to Learn

Easy Access to Data

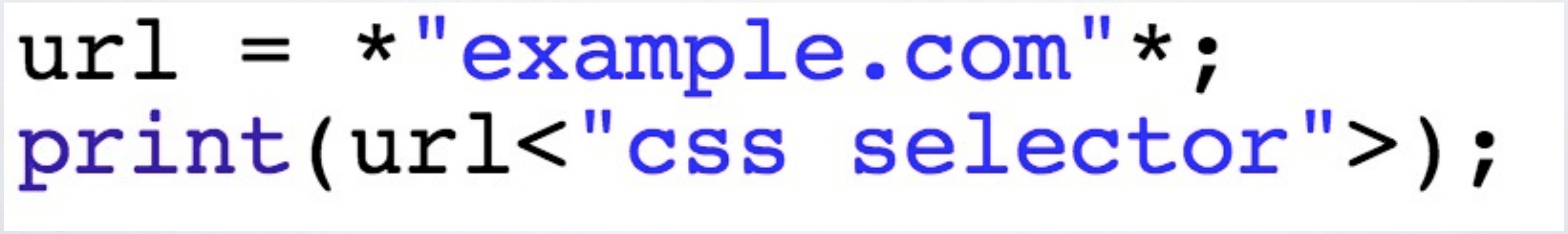
Easy Collection of Data

- ★ An object-oriented DSL for collecting data from the web
- ★ Enables access, collection and analysis of web documents out of the box.
- ★ Easy to learn for users with programming experience

PROPERTIES



Web-centric + *Duck Typed*



```
url = *"example.com"*;
print(url<"css selector">);
```

WHY SWIM?

★Data scraping has been done before

★ But... They've always used specialized libraries...

VS.

★ We wanted a DSL for data scraping that worked out of the box

WHY SWIM?

- ★ SWIM is for researchers
- ★ SWIM is for industry
- ★ SWIM is for general-purpose data analysis

Business *Commercial* *Research* *IT*

SWIM HAS...

★*An Expressive programming syntax*

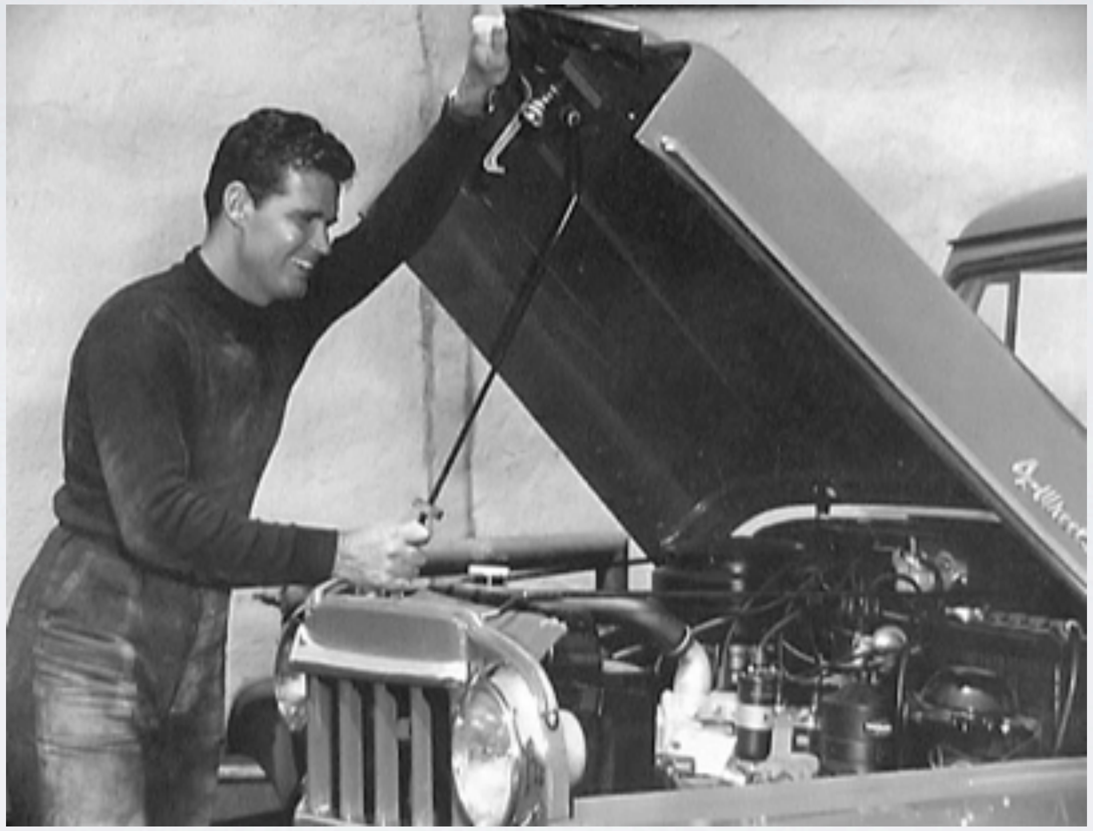
```
ars = *"arstechnica.com"*;
```

```
for each post in ars<".post"> do
  print(post);
end
```

```
print(ars<".post">);
```

SWIM HAS...

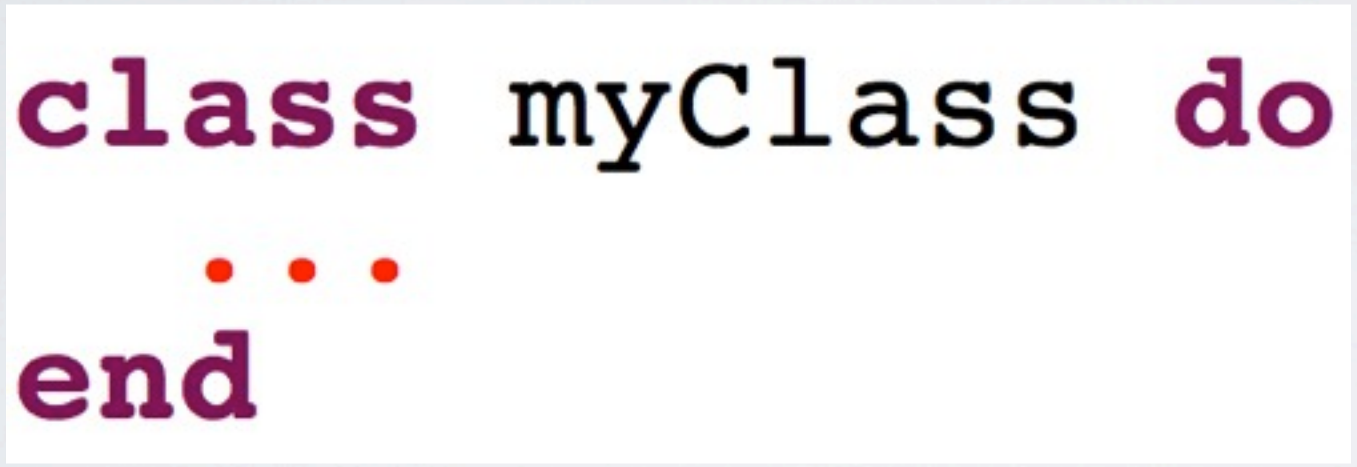
★*Tremendous Power Under The Hood*



for users with more specialized demands, SWIM provides an extensible class and function structure that allows for code reuse, makes use of several aspects of the functional programming paradigm - lambda functions

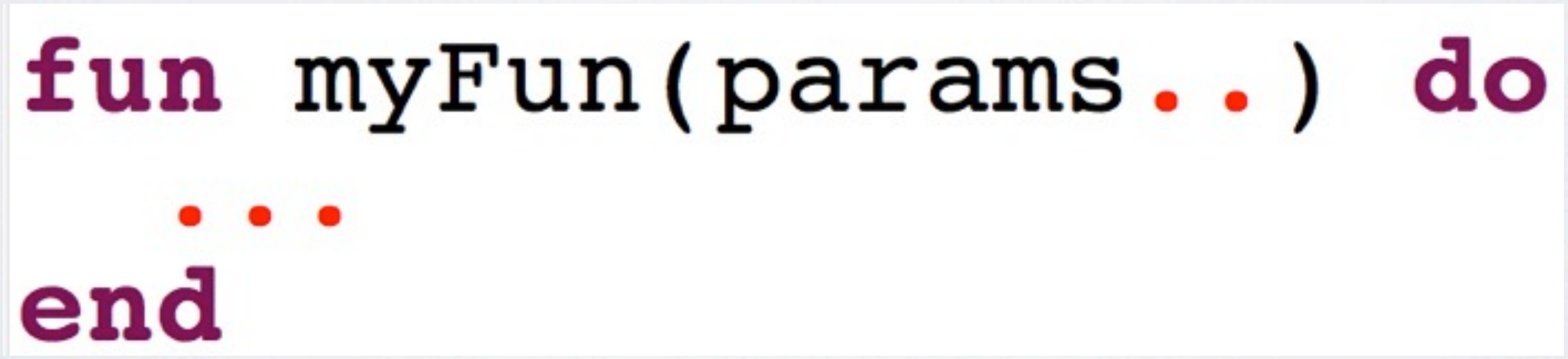
SYNTACTIC CONSTRUCTS

- *Classes*



```
class myClass do
   ...
end
```

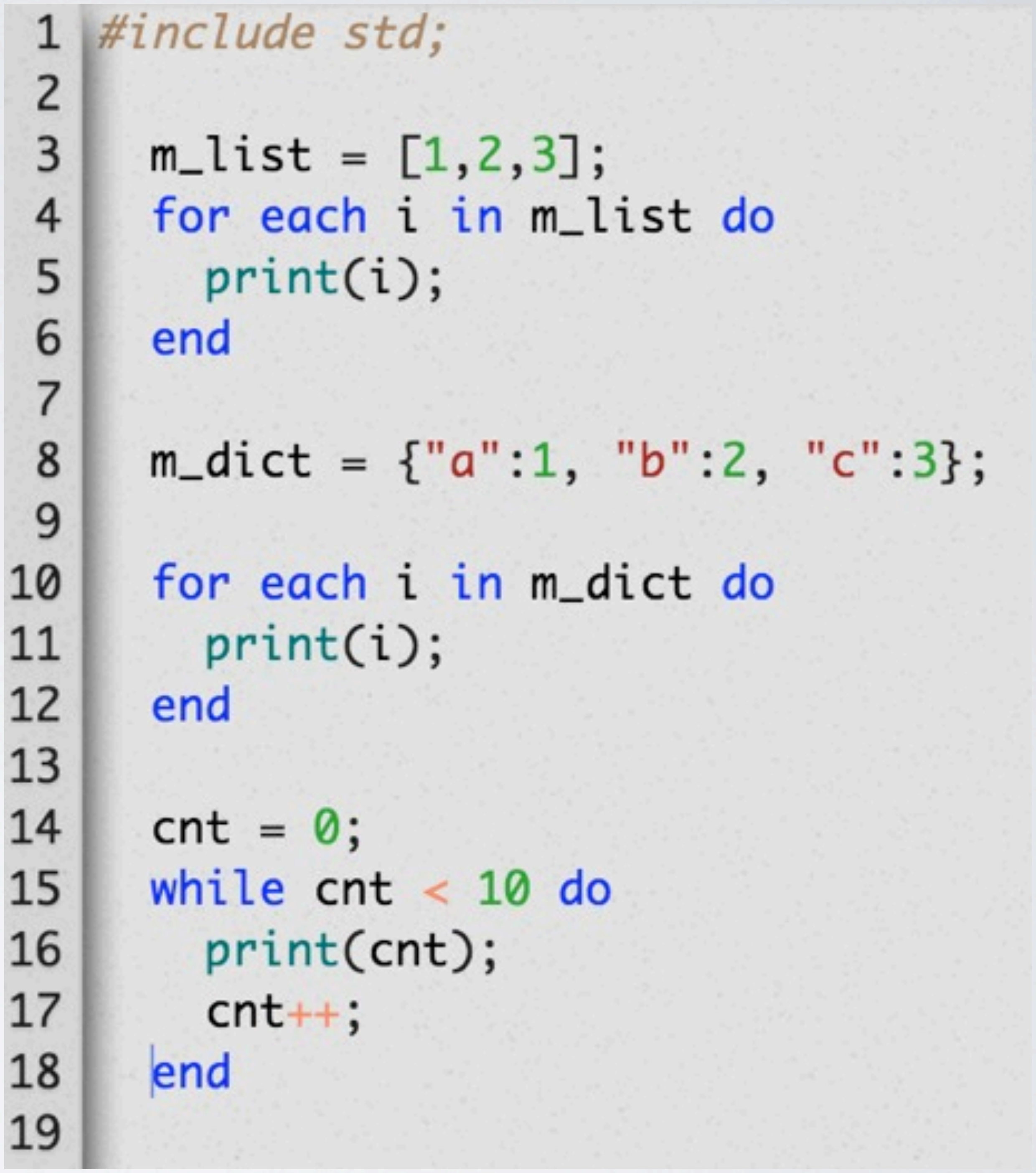
- *Functions*



```
fun myFun(params..) do
   ...
end
```

SYNTACTIC CONSTRUCTS

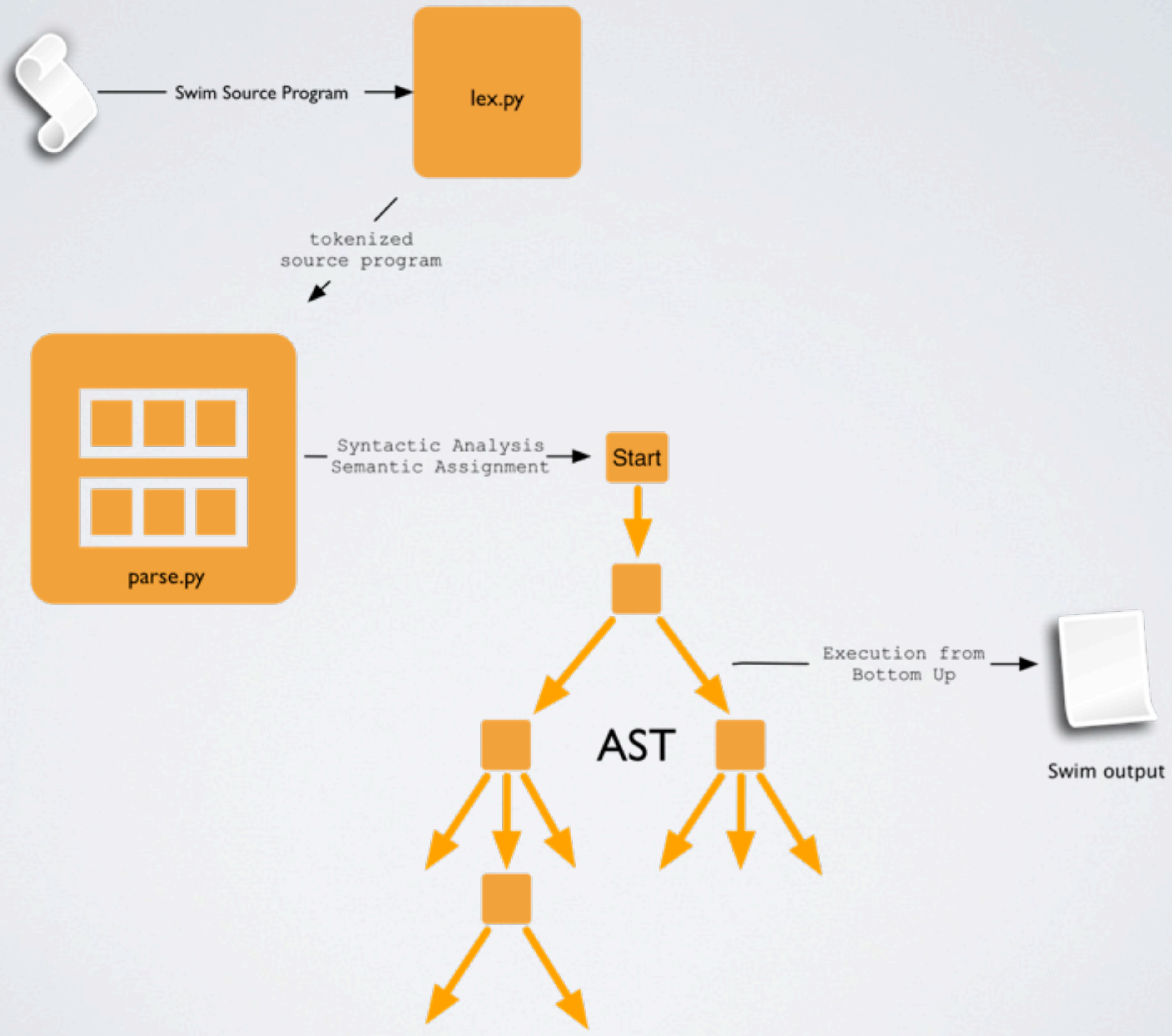
- ***Built-in Data Types***
  - *boolean*
  - *number*
- ***Derived Data Types***
  - *Lists*
  - *Dictionaries*
  - *URLs*
  - *Strings*
- ***Conditionals***
  - *if ... elif ... else*



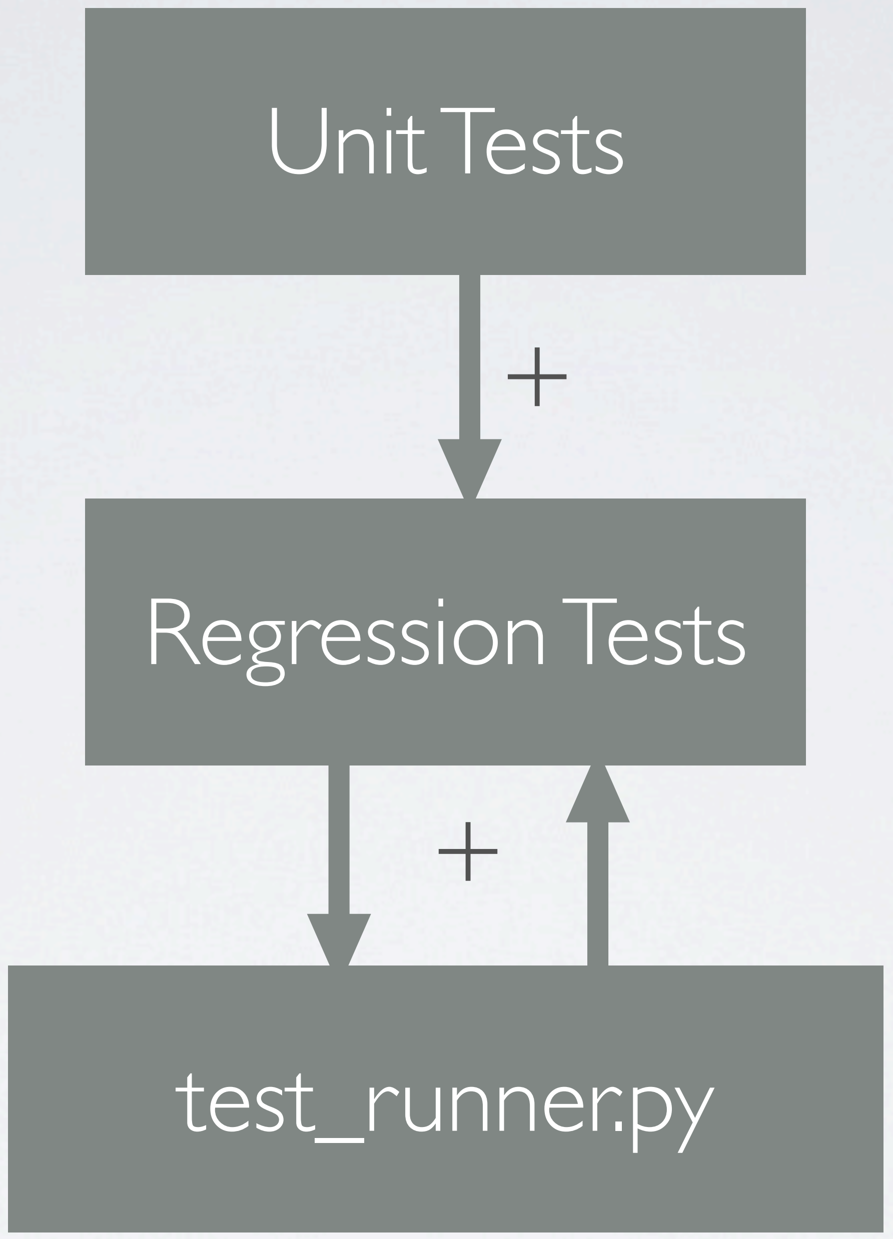
```
#include std;

  m_list = [1,2,3];
  for each i in m_list do
    print(i);
  end

  m_dict = {"a":1, "b":2, "c":3};

  for each i in m_dict do
    print(i);
  end

  cnt = 0;
  while cnt < 10 do
    print(cnt);
    cnt++;
  end

```

SWIM ARCHITECTURE

BLOCK DIAGRAM OF THE SWIM INTERPRETER



TESTING



DEVELOPMENT ENVIRONMENT

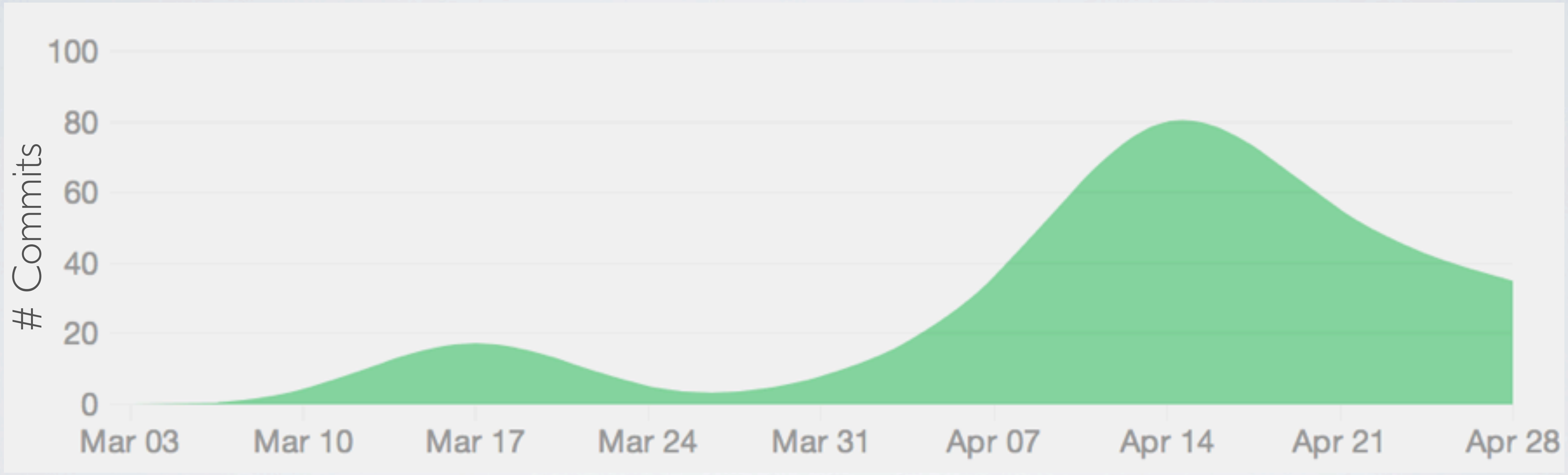
- ★ *Software Development Environment:*
- ★ *Lexing and Parsing: PLY/Python*
- ★ *Version Control: Git with Github hosting*
- ★ *Text Editor: Sublime Text 2*
- ★ *REPL: Custom SWIM REPL*
- ★ *Live Online Editor: swimco.de*
- ★ *Test Suite: Python runner for test suite of SWIM files with integrated regression testing*

PROJECT MANAGEMENT

- ★ *Mentoring Sessions with Prof. Aho*
- ★ *Weekly Scheduled Meetings + Additional Meetings*
- ★ *Google Drive for Documentation Management*
- ★ *Google+, Skype, Google Hangouts, Email, Text, etc. for coordinating meetings/remote work*

PROJECT MANAGEMENT

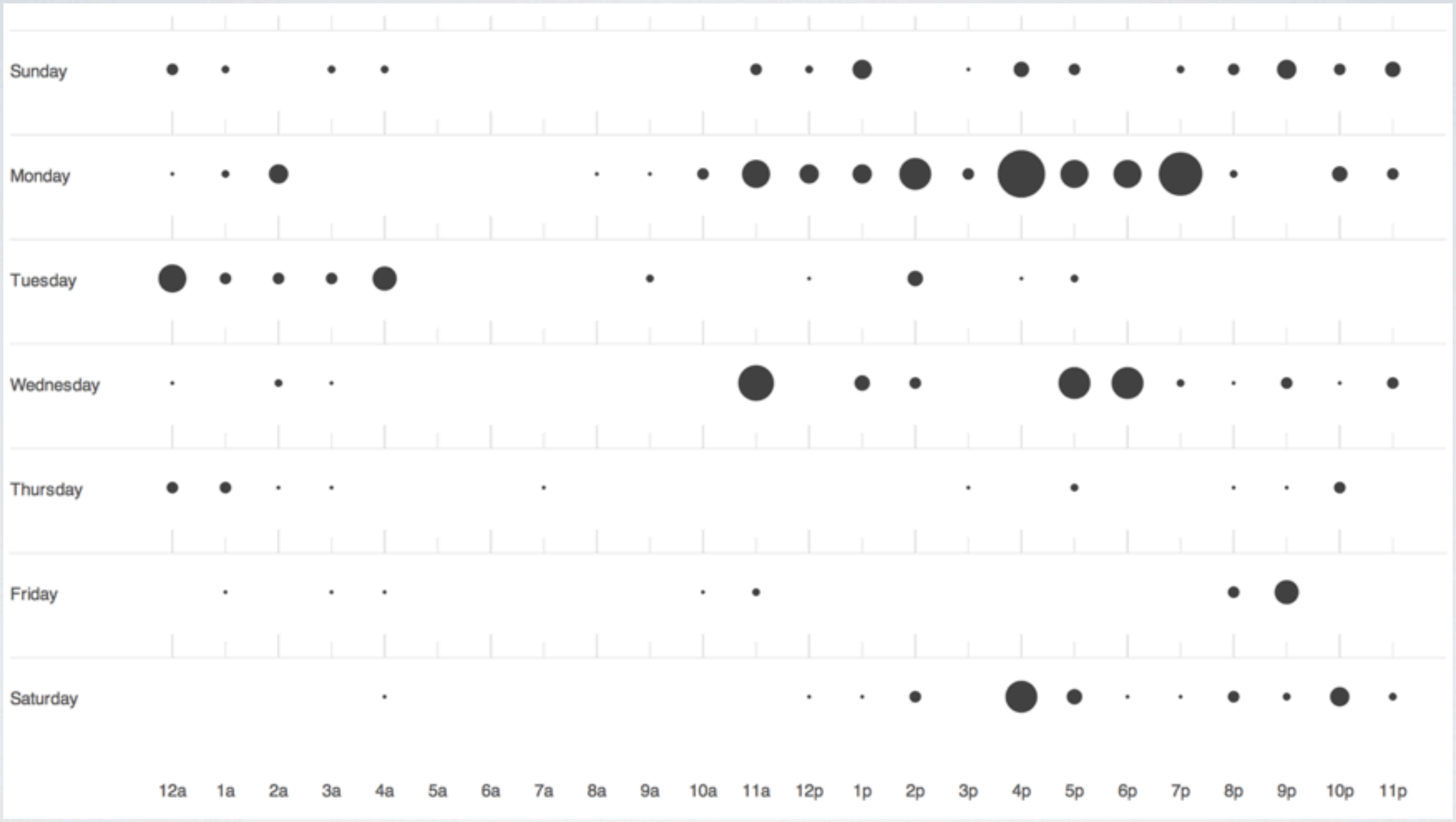
Github Commits By Date



Date

PROJECT MANAGEMENT

# Day of the Week



Time of Day

CONCLUSION

What we would do differently

- Plan for scoping right at the beginning

What worked well

- AST
- Version Control
- SWIM REPL

What we learned

- Writing your own programming language can be tricky.
- The more you take care of right at the beginning, the better!
- Create a regression test for everything! Problems will reappear over and over again.
- ***MAKE IT MODULAR!!! KEEP IT DRY!***

CONCLUSION

Let's SWIM!