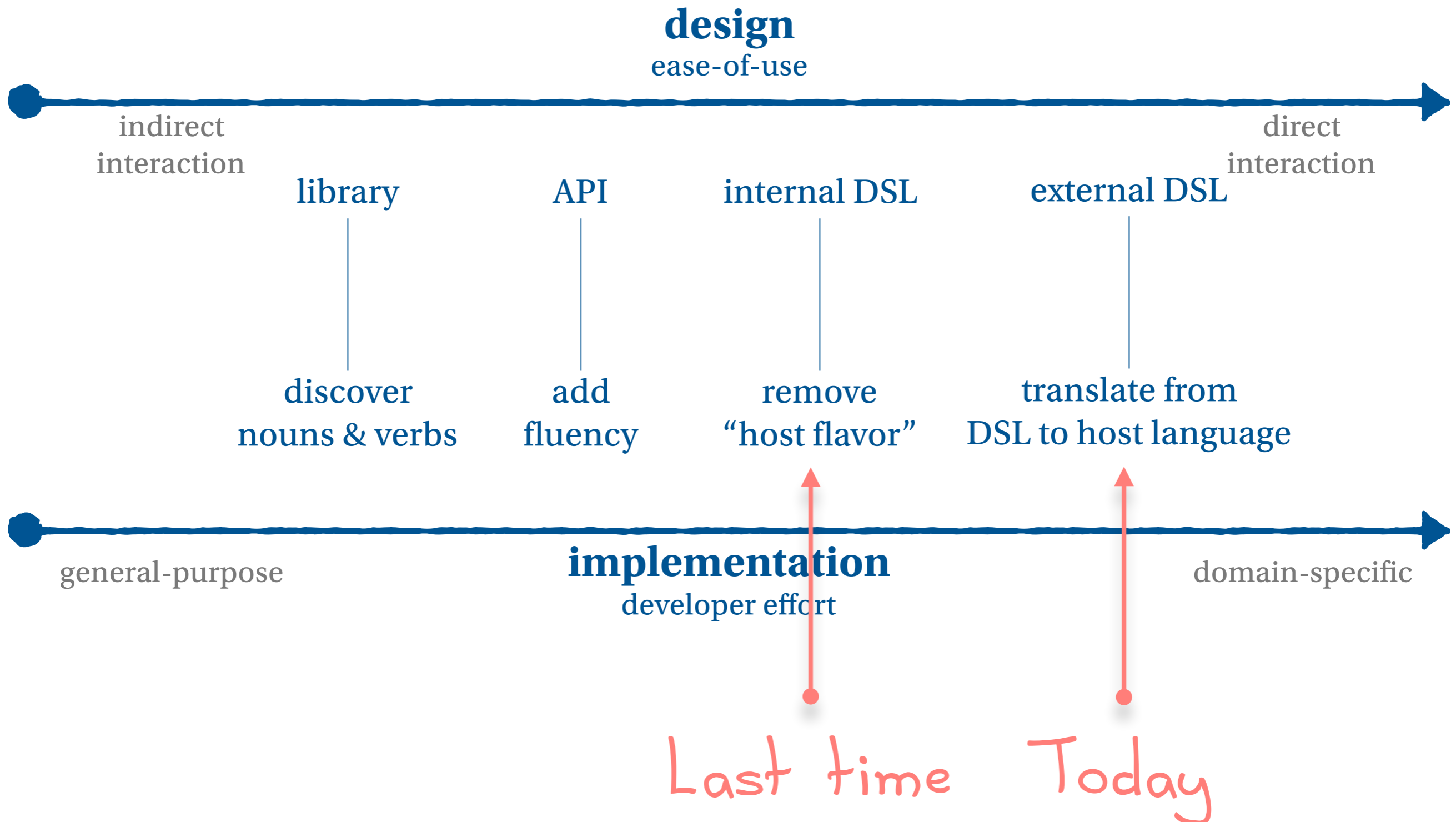


Today: implementation strategies

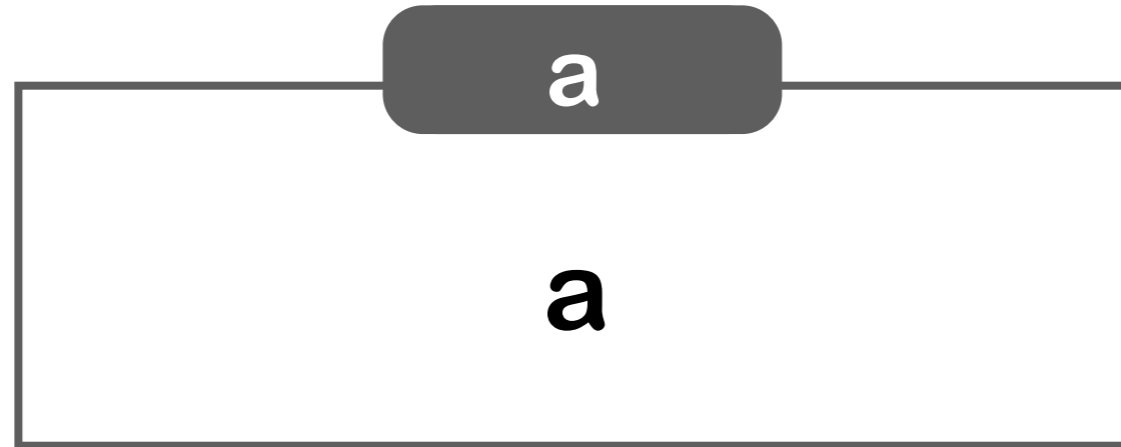


Regular expressions

for string-matching

Literal matching

one character matches

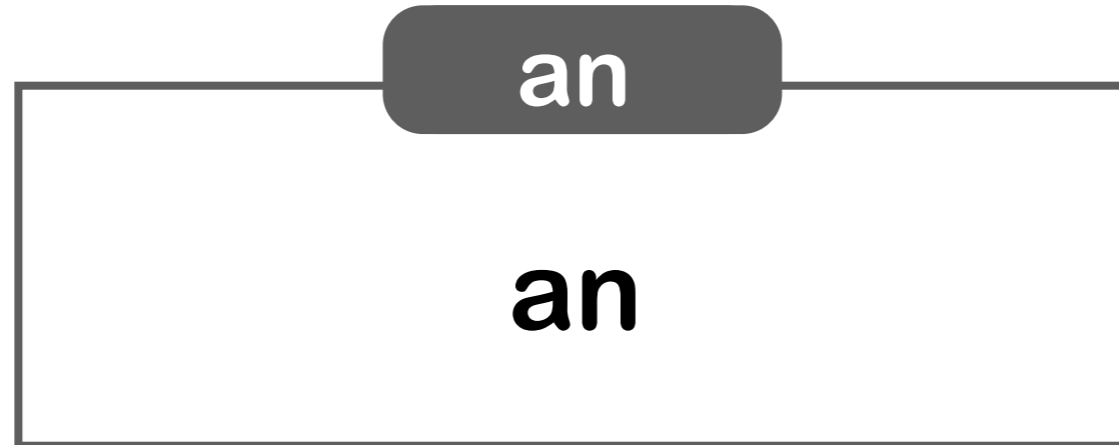


```
object Program extends App {  
  val pattern = ""a"".r  
  val string = "banana"  
  pattern.findAllIn(string) foreach println  
}
```

```
a  
a  
a
```

Sequential matching

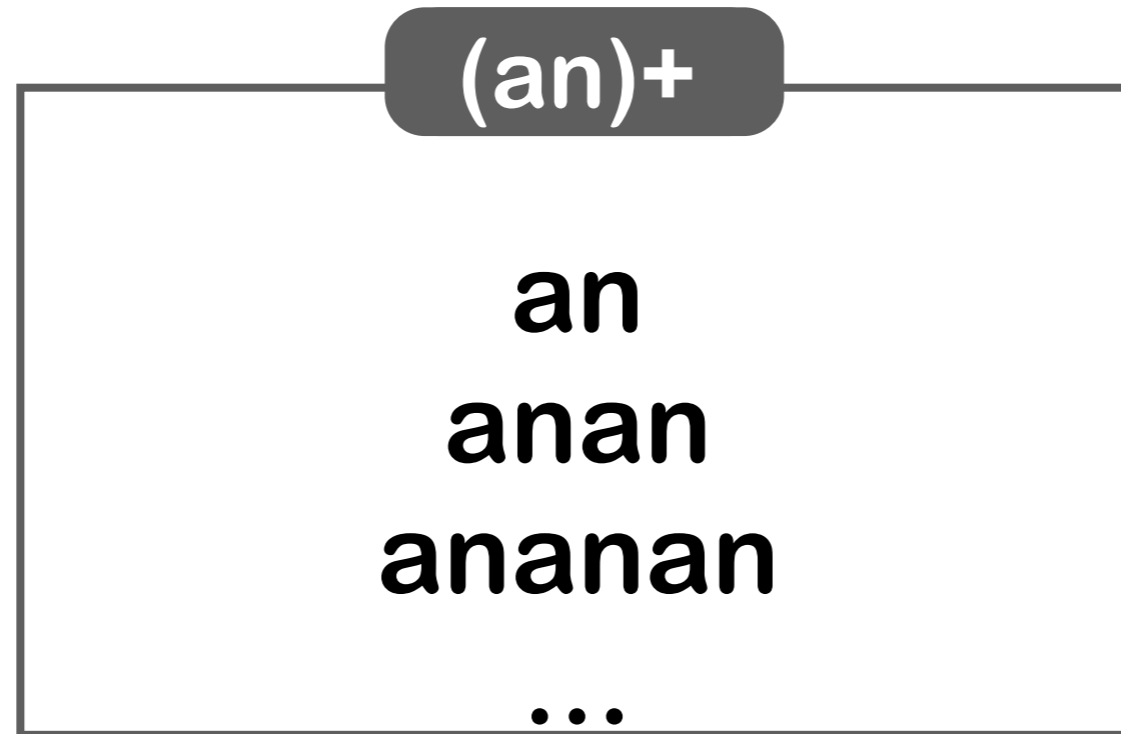
a sequence of characters matches



```
object Program extends App {  
  val pattern = ""an"".r  
  val string = "banana"  
  pattern.findAllIn(string) foreach println  
}
```

```
an  
an
```

Repetition: one or more



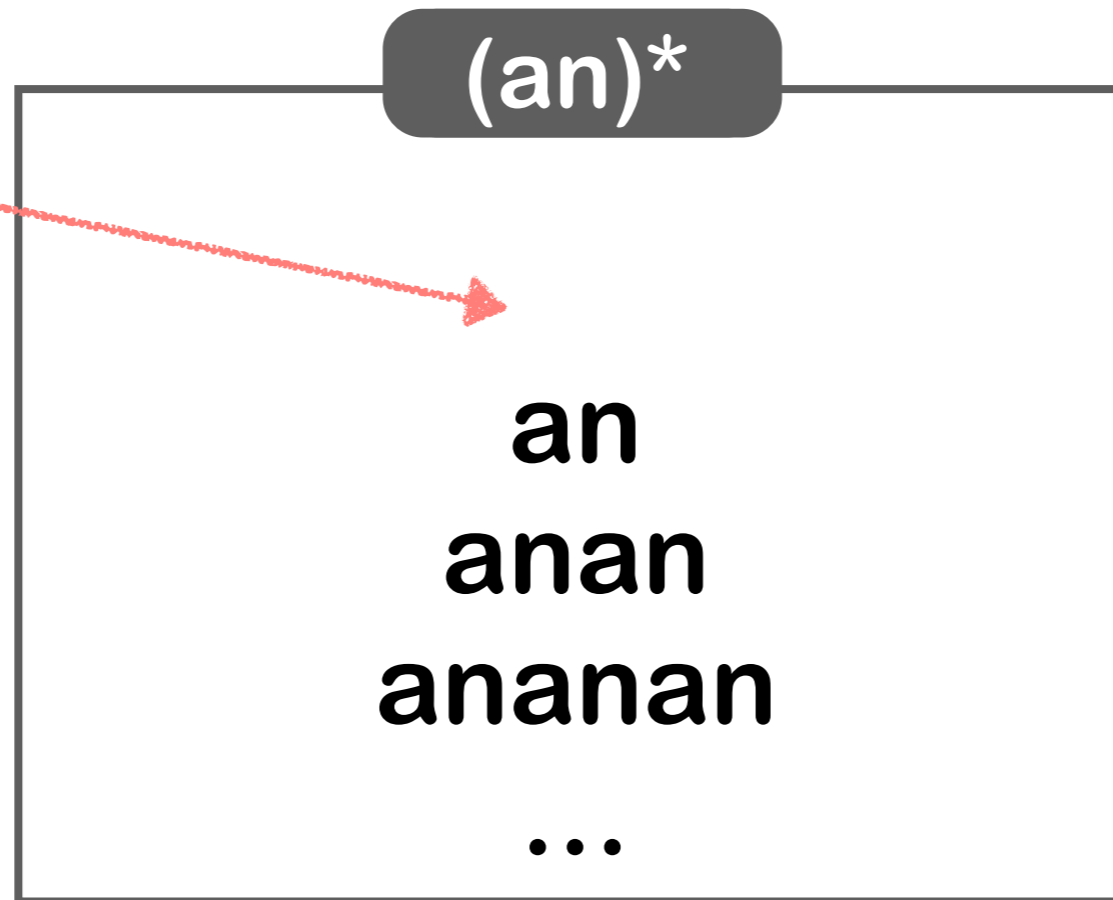
```
object Program extends App {  
  val pattern = """"(an)+""".r  
  val string = "banana"  
  pattern.findAllIn(string) foreach println  
}
```

anan

↑ finds the longest match that
doesn't overlap with any other match

Repetition: zero or more

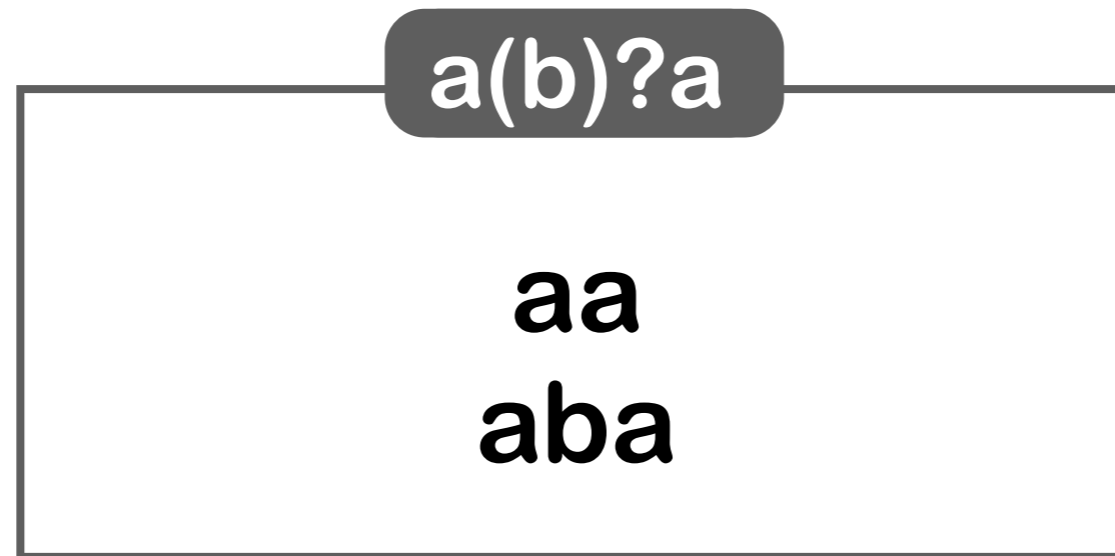
empty string
is a match



```
object Program extends App {  
  val pattern = """(an)*""".r  
  val string = "banana"  
  pattern.findAllIn(string) foreach println  
}
```

anan

Repetition: zero or one



```
object Program extends App {  
  val pattern = """a(b)?a""".r  
  val string = "aa"  
  pattern.findAllIn(string) foreach println  
}
```

aa

Alternatives

two ways to say: "this or that"

a|b

a

b

[ab]

a

b

```
object Program extends App {  
  val pattern = ""a|b"".r  
  val string = "banana"  
  pattern.findAllIn(string) foreach println  
}
```

b
a
a
a

```
object Program extends App {  
  val pattern = ""[ab]"".r  
  val string = "banana"  
  pattern.findAllIn(string) foreach println  
}
```

b
a
a
a

Negation

not these characters

[^ab]

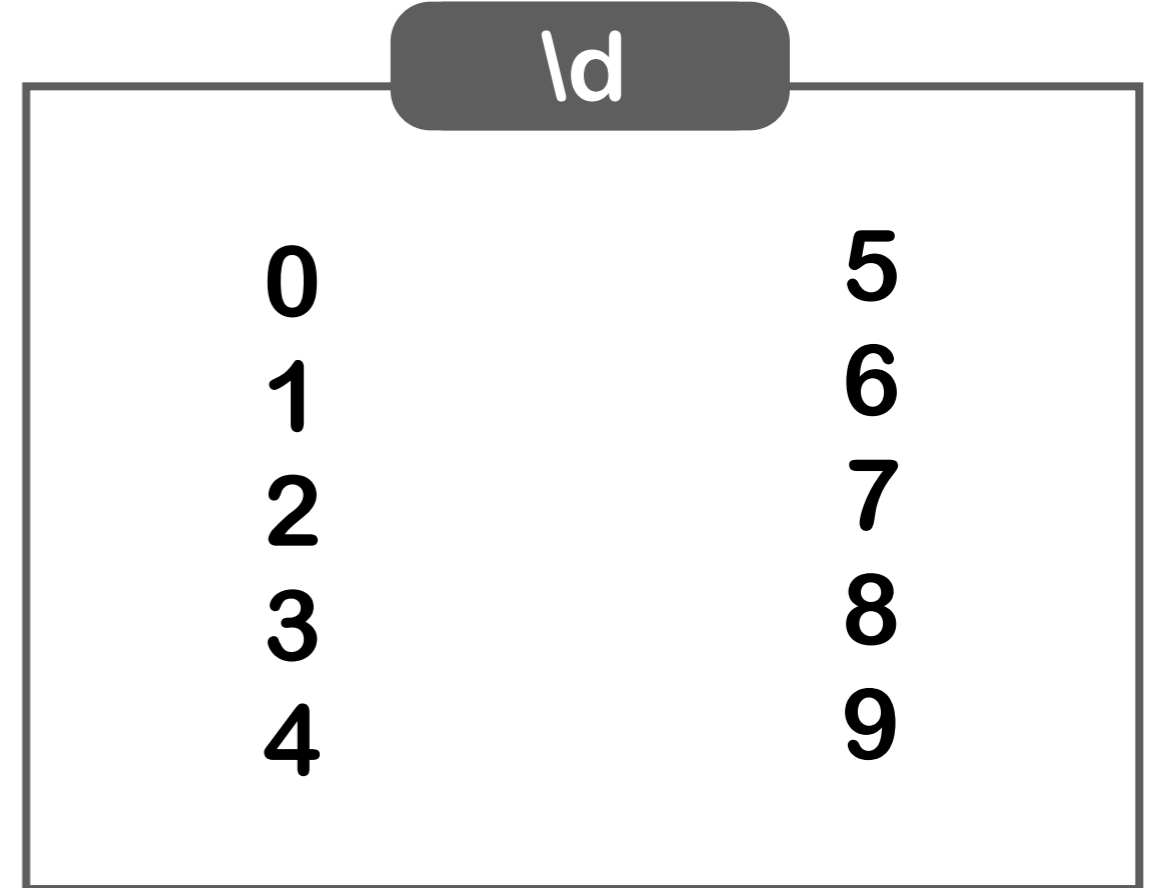
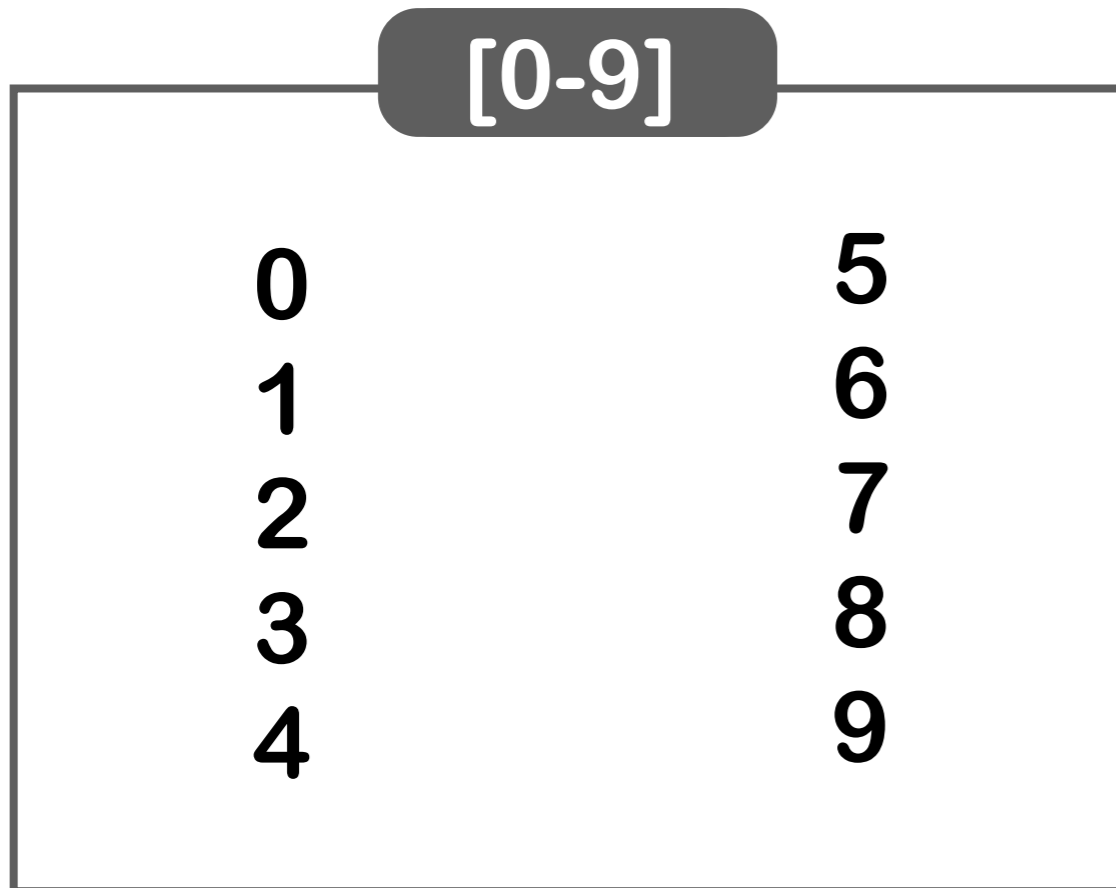
(anything but a or b)

```
object Program extends App {  
  val pattern = "[^ab]".r  
  val string = "banana"  
  pattern.findAllIn(string) foreach println  
}
```

```
n  
n
```

Character classes

multiple ways to say: any character in this range



```
object Program extends App {  
  val pattern = "[0-9]".r  
  val string = "abc123forty7"  
  pattern.findAllIn(string) foreach println  
}
```

```
1  
2  
3  
7
```

```
object Program extends App {  
  val pattern = "\\d".r  
  val string = "abc123forty7"  
  pattern.findAllIn(string) foreach println  
}
```

```
1  
2  
3  
7
```

More character classes

Decimals	\d	[0-9]
	\D	[^0-9]
Whitespace	\s	[\t\n\r\f\v]
	\S	[^ \t\n\r\f\v]
Alpha-Numeric	\w	[a-zA-Z0-9_]
	\W	[^a-zA-Z0-9_]
Backslash	\\	[\\]
Any Character	.	(everything)

Matching an entire string

```
object Program extends App {  
  val pattern = """"^\\w+\\d+$"""".r  
  val string = "abc123"  
  pattern.findAllIn(string) foreach println  
}
```

abc123

```
object Program extends App {  
  val pattern = """"^\\w+\\d+$"""".r  
  val string = "abc123a"  
  pattern.findAllIn(string) foreach println  
}
```

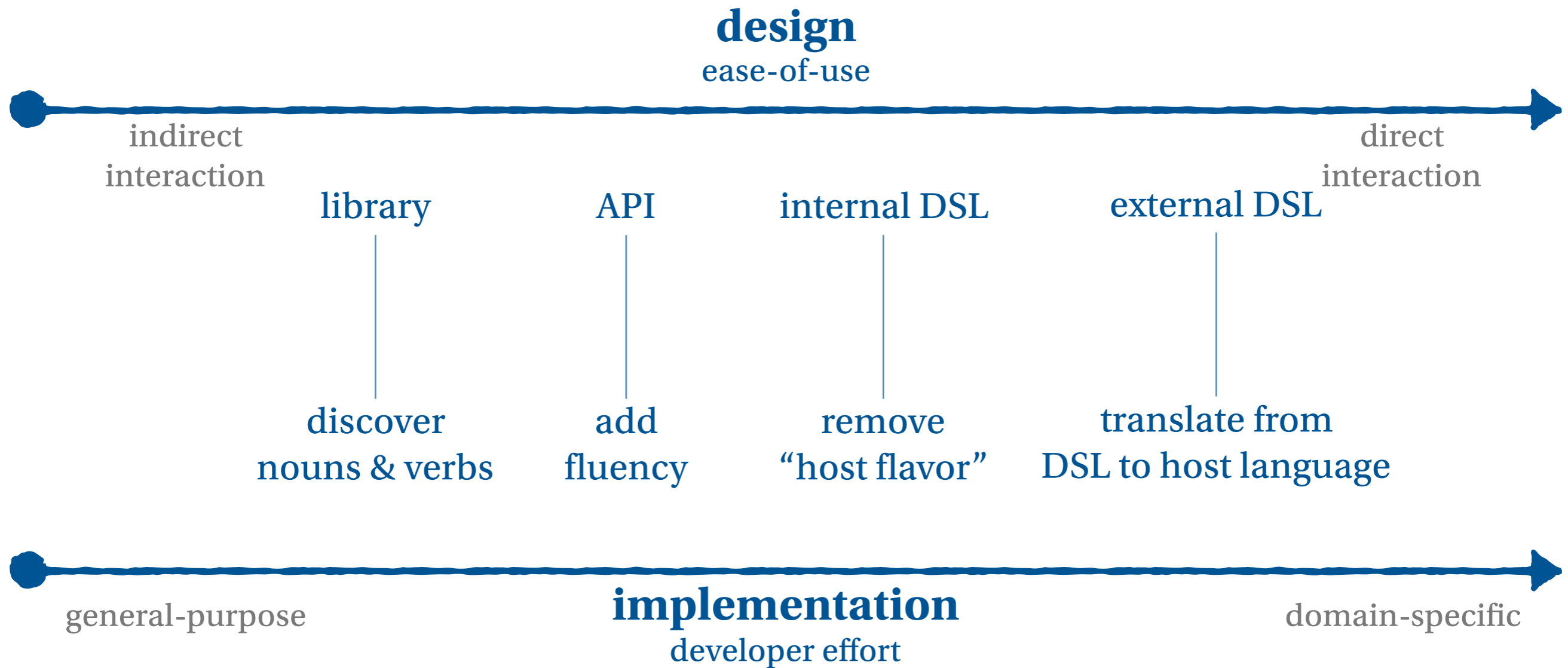
Groups and matching

Scala has some special magic that helps us pull matches out of strings

```
object Program extends App {  
  val pattern = """"^([a-z]+)(\d+)$$""".r  
  val string = "abc123"  
  val pattern(letters, numbers) = string  
  println(s"Letters: $letters")  
  println(s"Numbers: $numbers")  
}
```

```
Letters: abc  
Numbers: 123
```

DSL implementation strategies



Is it a DSL?

Programming Language

Describe something to a computer.

General-purpose

Allow a professional programmer to write an arbitrary program.

restrict focus

Domain-specific

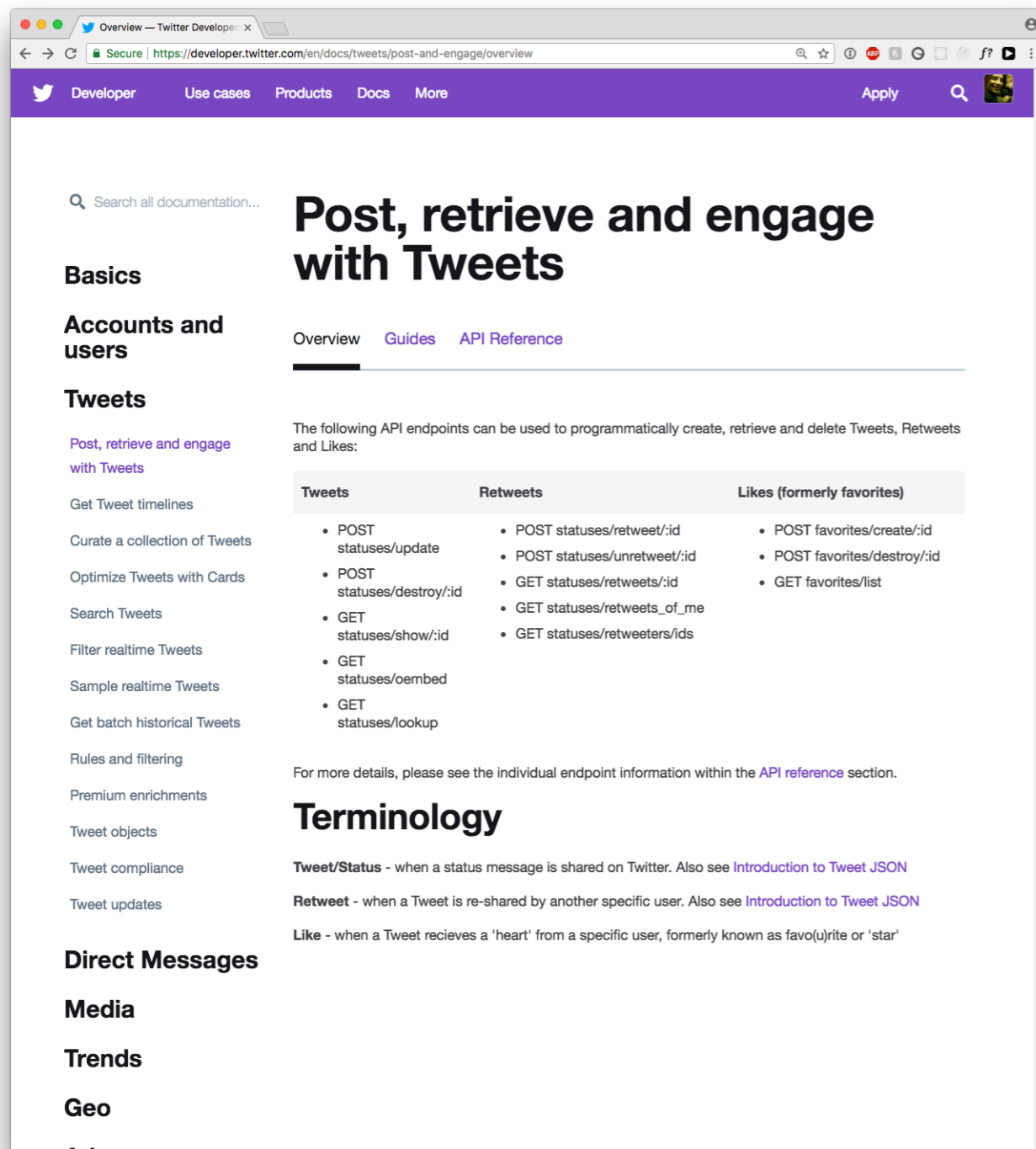
Allow a **domain expert** to interact directly with their domain.

We should have good answers for all these questions

1. Is it a programming language?
2. What is the focus? What does the *domain expert* describe?
3. What is easy, difficult, impossible in this language?

(relative to a general-purpose programming language)

Example: REST API



Overview — Twitter Developer x
Secure | https://developer.twitter.com/en/docs/tweets/post-and-engage/overview

Developer Use cases Products Docs More Apply

Search all documentation...

Post, retrieve and engage with Tweets

Overview Guides API Reference

The following API endpoints can be used to programmatically create, retrieve and delete Tweets, Retweets and Likes:

Tweets	Retweets	Likes (formerly favorites)
<ul style="list-style-type: none">POST statuses/updatePOST statuses/destroy/:idGET statuses/show/:idGET statuses/oembedGET statuses/lookup	<ul style="list-style-type: none">POST statuses/retweet/:idPOST statuses/unretweet/:idGET statuses/retweets/:idGET statuses/retweets_of_meGET statuses/retweeters/ids	<ul style="list-style-type: none">POST favorites/create/:idPOST favorites/destroy/:idGET favorites/list

For more details, please see the individual endpoint information within the [API reference](#) section.

Terminology

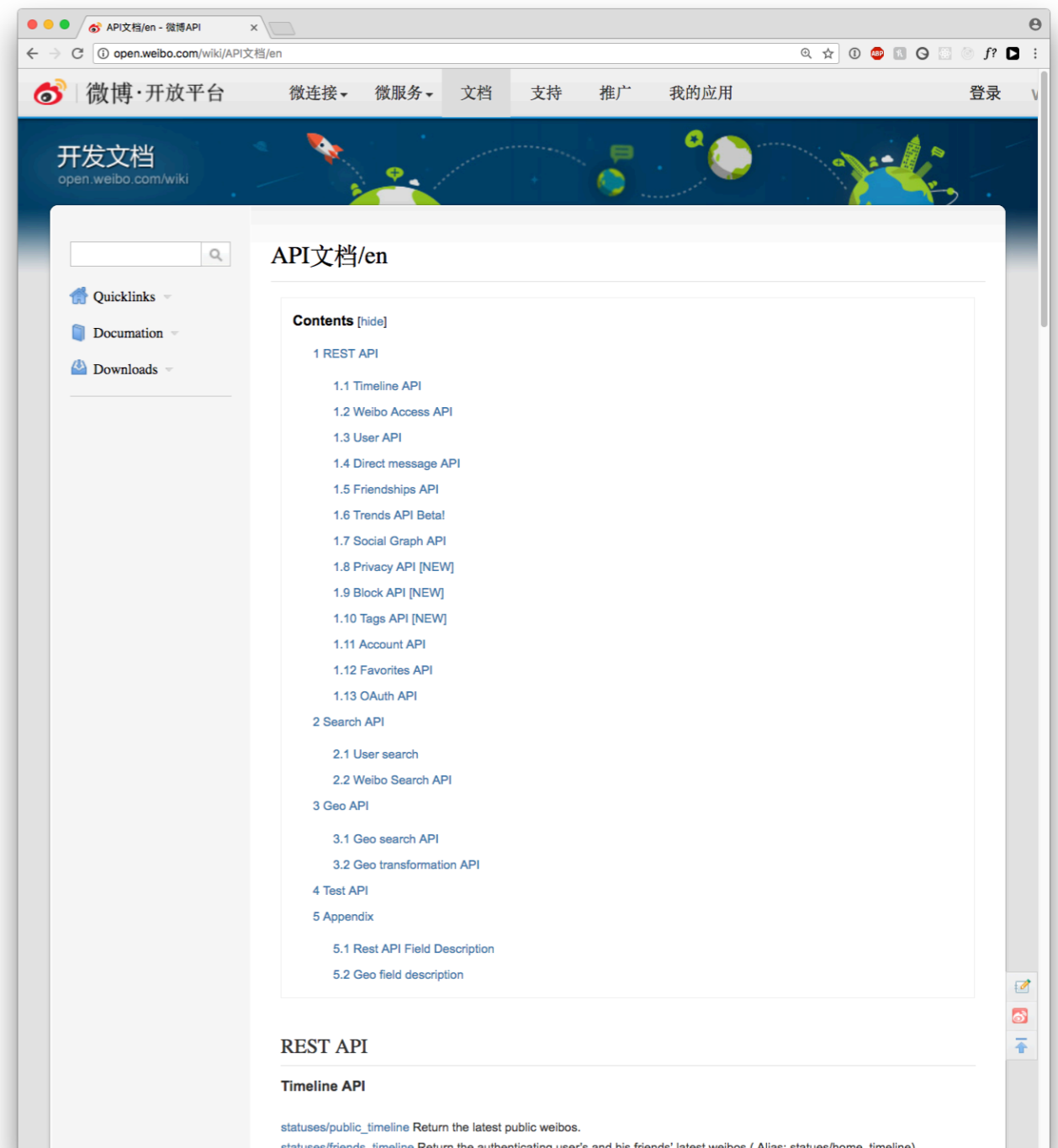
Tweet/Status - when a status message is shared on Twitter. Also see [Introduction to Tweet JSON](#)

Retweet - when a Tweet is re-shared by another specific user. Also see [Introduction to Tweet JSON](#)

Like - when a Tweet receives a 'heart' from a specific user, formerly known as favo(u)rite or 'star'

Basics
Accounts and users
Tweets
Direct Messages
Media
Trends
Geo

twitter



API文档/en - 微博API
open.weibo.com/wiki/API文档/en

微博·开放平台 微连接 微服务 文档 支持 推广 我的应用 登录

API文档/en

Contents [hide]

- 1 REST API
 - 1.1 Timeline API
 - 1.2 Weibo Access API
 - 1.3 User API
 - 1.4 Direct message API
 - 1.5 Friendships API
 - 1.6 Trends API Beta!
 - 1.7 Social Graph API
 - 1.8 Privacy API [NEW]
 - 1.9 Block API [NEW]
 - 1.10 Tags API [NEW]
 - 1.11 Account API
 - 1.12 Favorites API
 - 1.13 OAuth API
- 2 Search API
 - 2.1 User search
 - 2.2 Weibo Search API
- 3 Geo API
 - 3.1 Geo search API
 - 3.2 Geo transformation API
- 4 Test API
- 5 Appendix
 - 5.1 Rest API Field Description
 - 5.2 Geo field description

REST API

Timeline API

`statuses/public_timeline` Return the latest public weibos.
`statuses/friends_timeline` Return the authenticating user's and his friends' latest weibos (Alias: statuses/home_timeline)

weibo

Example: internal DSL w/ chaining

Stacked-to-Grouped Bars - bl... x

bl.ocks.org/mbostock/3943967

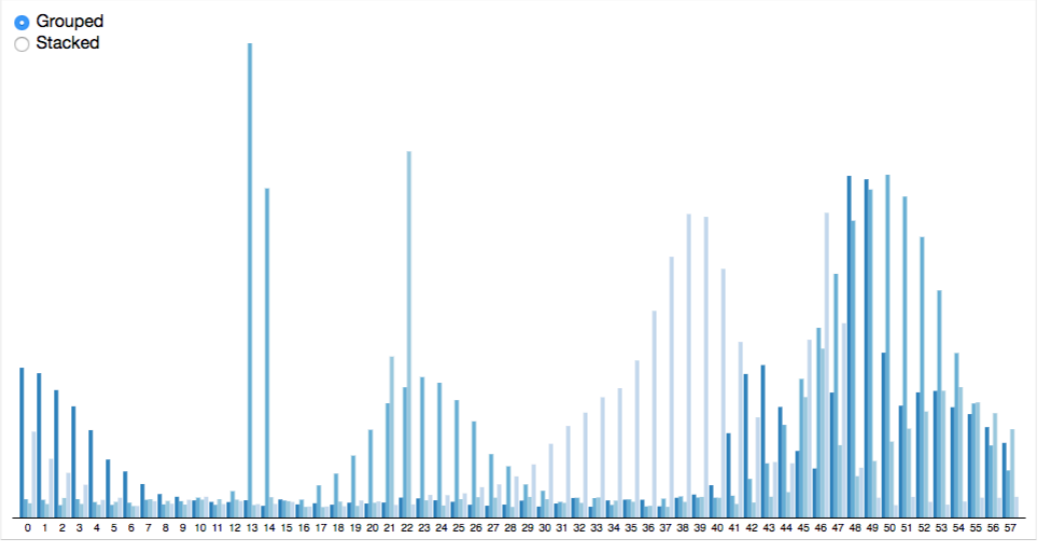
Like visualization and creative coding? Try interactive JavaScript notebooks in **Observable!**

Mike Bostock's Block 3943967
Updated December 7, 2016

Popular / About

Stacked-to-Grouped Bars

Grouped
 Stacked



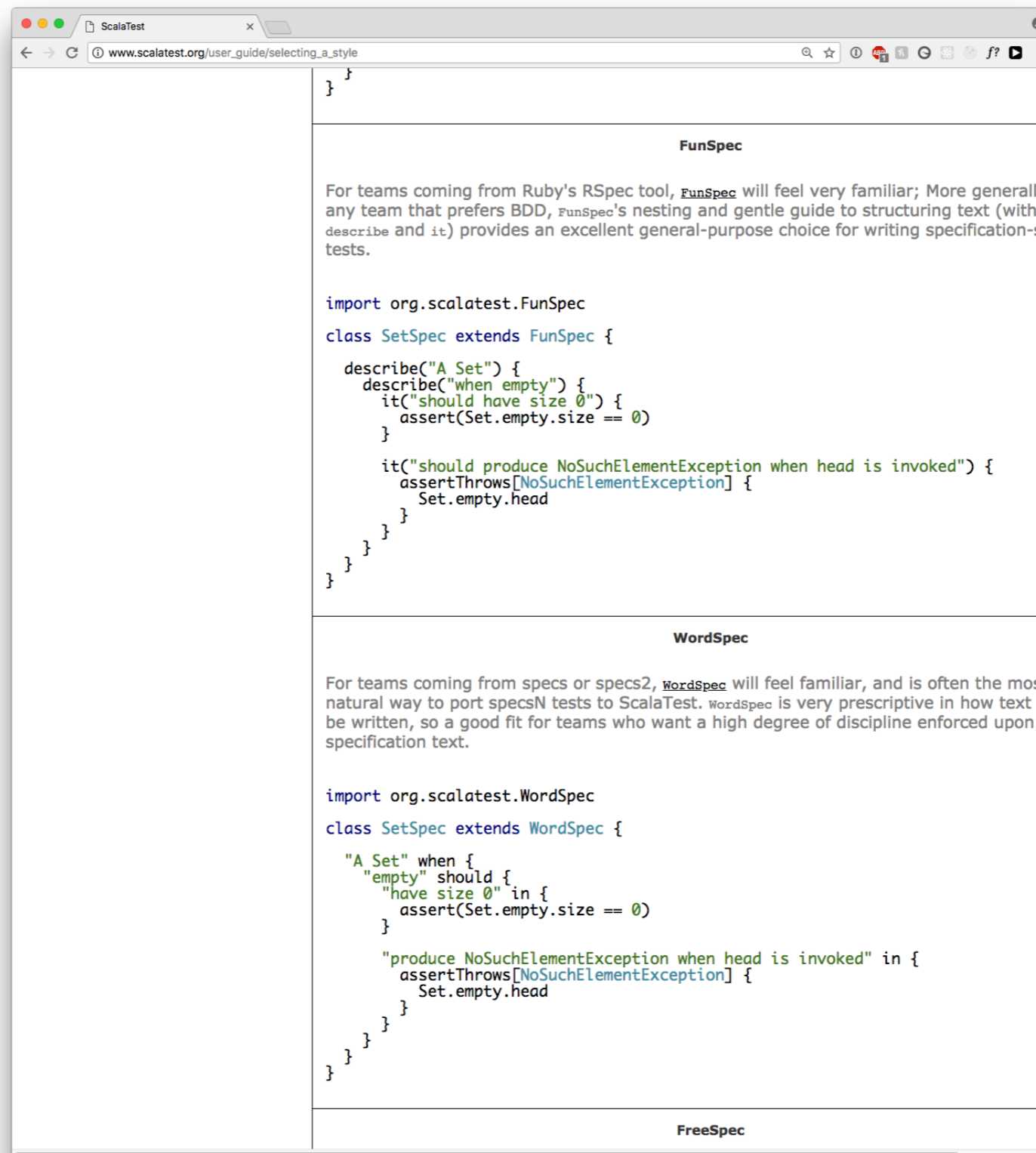
Switch between stacked and grouped layouts using sequenced transitions! Animations preserve object constancy and allow the user to follow the data across views. Animation design by [Heer and Robertson](#). Inspired by [Byron and Wattenberg](#). [Open](#)

index.html

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>
form {
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  position: absolute;
  left: 10px;
  top: 10px;
}
label {
  display: block;
}
</style>
<form>
  <label><input type="radio" name="mode" value="grouped"> Grouped</label>
  <label><input type="radio" name="mode" value="stacked" checked> Stacked</label>
</form>
<svg width="960" height="500"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
</script>
```

D3

Example: fluent internal DSL



The screenshot shows a web browser window with the URL `www.scalatest.org/user_guide/selecting_a_style`. The page content is as follows:

FunSpec

For teams coming from Ruby's RSpec tool, `FunSpec` will feel very familiar; More generally any team that prefers BDD, `FunSpec`'s nesting and gentle guide to structuring text (with `describe` and `it`) provides an excellent general-purpose choice for writing specification-s tests.

```
import org.scalatest.FunSpec
class SetSpec extends FunSpec {
  describe("A Set") {
    describe("when empty") {
      it("should have size 0") {
        assert(Set.empty.size == 0)
      }
      it("should produce NoSuchElementException when head is invoked") {
        assertThrows[NoSuchElementException] {
          Set.empty.head
        }
      }
    }
  }
}
```

WordSpec

For teams coming from specs or specs2, `WordSpec` will feel familiar, and is often the most natural way to port specsN tests to ScalaTest. `WordSpec` is very prescriptive in how text i be written, so a good fit for teams who want a high degree of discipline enforced upon specification text.

```
import org.scalatest.WordSpec
class SetSpec extends WordSpec {
  "A Set" when {
    "empty" should {
      "have size 0" in {
        assert(Set.empty.size == 0)
      }
      "produce NoSuchElementException when head is invoked" in {
        assertThrows[NoSuchElementException] {
          Set.empty.head
        }
      }
    }
  }
}
```

FreeSpec

ScalaTest

Example: external DSL

WebGraphviz is [Graphviz](#) in the Browser

Enter your graphviz data into the Text Area:
(Your Graphviz data is private and never harvested)

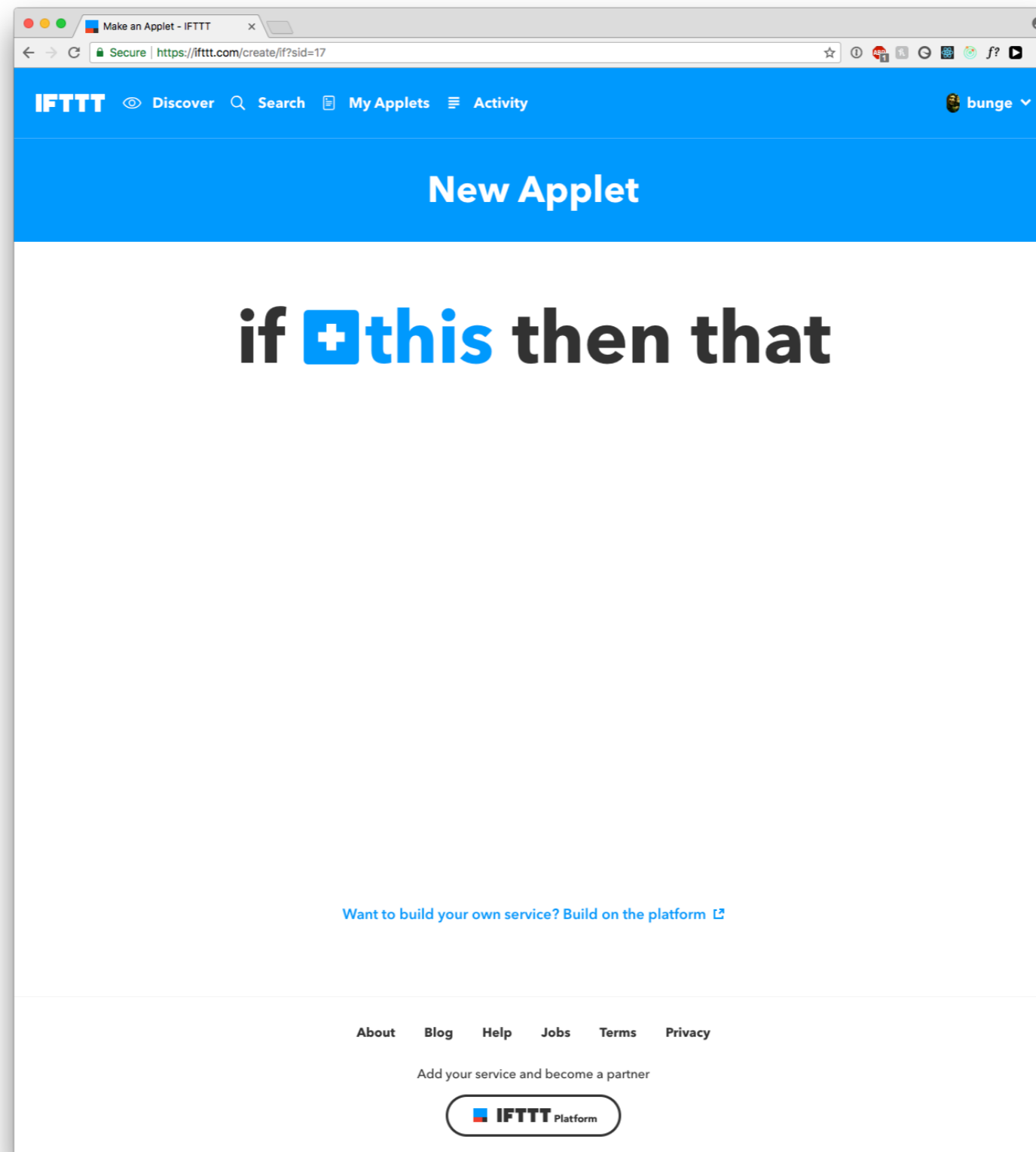
Sample 1 Sample 2 Sample 3 Sample 4 Sample 5

```
digraph finite_state_machine {
  rankdir=LR;
  size="8,5"
  node [shape = doublecircle]; LR_0 LR_3 LR_4 LR_8;
  node [shape = circle];
  LR_0 -> LR_2 [ label = "SS(B)" ];
  LR_0 -> LR_1 [ label = "SS(S)" ];
  LR_1 -> LR_3 [ label = "S($end)" ];
  LR_2 -> LR_6 [ label = "SS(b)" ];
  LR_2 -> LR_5 [ label = "SS(a)" ];
  LR_2 -> LR_4 [ label = "S(A)" ];
  LR_5 -> LR_7 [ label = "S(b)" ];
  LR_5 -> LR_5 [ label = "S(a)" ];
  LR_6 -> LR_6 [ label = "S(b)" ];
  LR_6 -> LR_5 [ label = "S(a)" ];
  LR_6 -> LR_8 [ label = "S(b)" ];
  LR_7 -> LR_8 [ label = "S(b)" ];
  LR_7 -> LR_5 [ label = "S(a)" ];
  LR_8 -> LR_6 [ label = "S(a)" ];
  LR_8 -> LR_5 [ label = "S(b)" ];
  LR_8 -> LR_5 [ label = "S(a)" ];
}
```

Generate Graph!

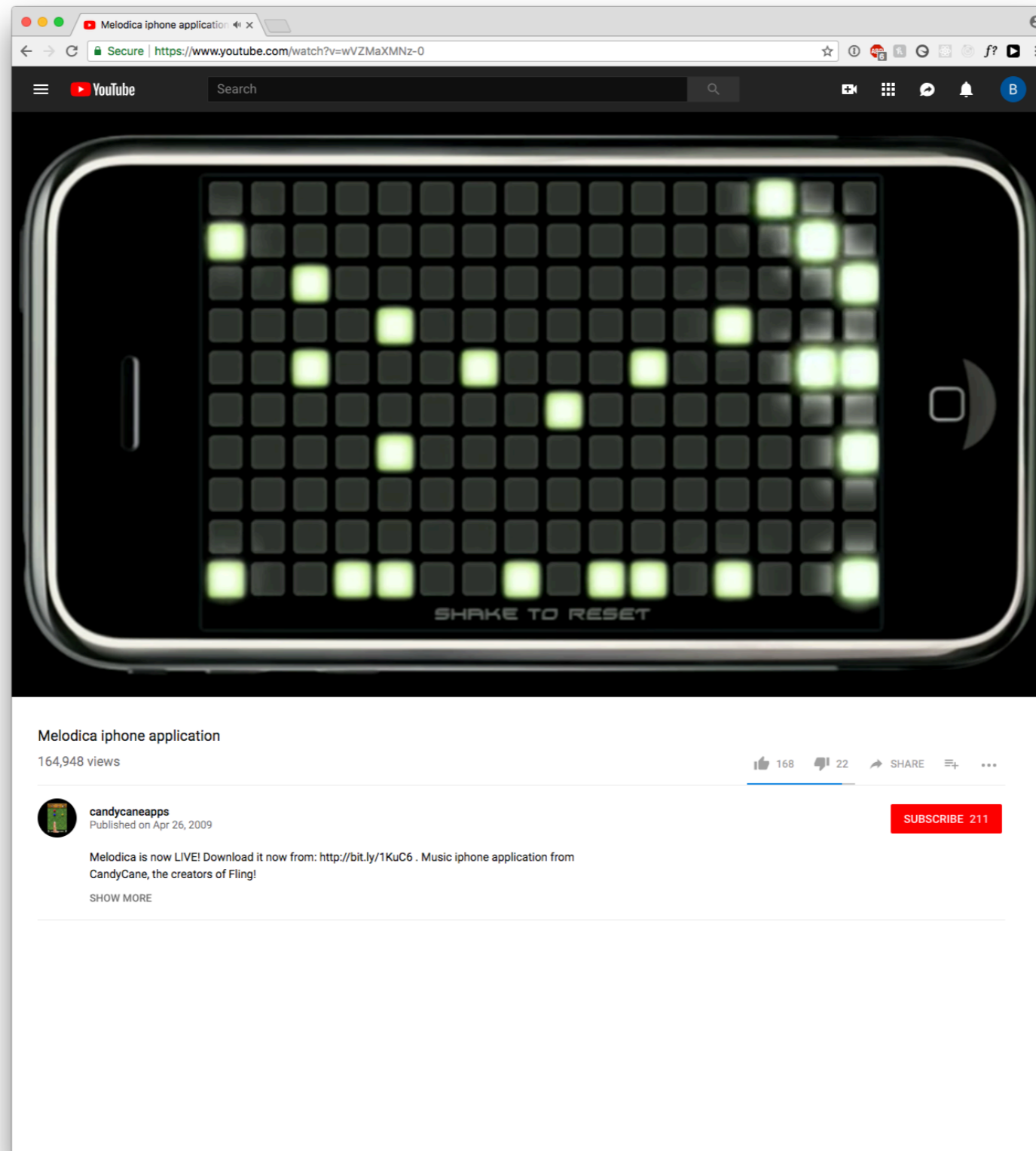
Graphviz

Example



If This Then That (IFTTT)

Example



Melodica