

# RUBY

bits

DSL PART 1

PRESS START

# DOMAIN SPECIFIC LANGUAGES

## DSL

a language that has terminology and constructs designed for a specific domain

## External DSL

a standalone DSL

## Internal DSL

a DSL implemented within another programming language



# OUR SAMPLE DSL

```
tweet_as 'markkendall' do
  mention 'codeschool'
  text 'I made a DSL!'
  hashtag 'hooray'
  hashtag 'ruby'
  link 'http://codeschool.com'
end
```

..... valid Ruby syntax  
with custom terminology



@codeschool I made a DSL! #hooray #ruby http://codeschool.com

**RUBY**  
BITS

# FIRST IMPLEMENTATION

```
tweet_as 'markkendall' do
  mention 'codeschool'
  text 'I made a DSL!'
  hashtag 'hooray'
  hashtag 'ruby'
  link 'http://codeschool.com'
end
```

```
def tweet_as(user)
  @user = user
  @tweet = []
  yield
  submit_to_twitter
end
```

```
def text(str)
  @tweet << str
end

def mention(user)
  @tweet << "@" + user
end

def hashtag(str)
  @tweet << "#" + str
end

def link(str)
  @tweet << str
end
```

RUBY  
BITS

# FIRST IMPLEMENTATION

```
tweet_as 'markkendall' do
  mention 'codeschool'
  text 'I made a DSL!'
  hashtag 'hooray'
  hashtag 'ruby'
  link 'http://codeschool.com'
end
```



We polluted our  
namespace!!

```
def tweet_as(user)
  @user = user
  @tweet = []
  yield
  submit_to_twitter
end

def submit_to_twitter
  tweet_text = @tweet.join(' ')
  puts "#{@user}: #{tweet_text}"
end

def text(str)
  @tweet << str
end

...
```

# NAMESPACE POLLUTION

```
def tweet_as(user)
  tweet = Tweet.new(user)
  yield tweet
  tweet.submit_to_twitter
end
```

new block parameter

implementation  
didn't change

```
class Tweet
  def initialize(user)
    @user = user
    @tweet = []
  end

  def submit_to_twitter
    tweet_text = @tweet.join(' ')
    puts "#{@user}: #{tweet_text}"
  end

  def text(str)
    @tweet << str
  end

  ...
end
```



# NAMESPACE POLLUTION

```
def tweet_as(user)
  tweet = Tweet.new(user)
  yield tweet
  tweet.submit_to_twitter
end
```

```
tweet_as 'markkendall' do |tweet|
  tweet.mention 'codeschool'
  tweet.text 'I made a DSL!'
  tweet.hashtag 'hooray'
  tweet.hashtag 'ruby'
  tweet.link 'http://codeschool.com'
end
```



we made our  
DSL syntax ugly



# INSTANCE\_EVAL

```
def tweet_as(user, &block)
  tweet = Tweet.new(user)
  tweet.instance_eval(&block)
  tweet.submit_to_twitter
end
```

..... capture the block,  
..... pass it to instance\_eval

```
tweet_as 'markkendall' do
  mention 'codeschool'
  text 'I made a DSL!'
  hashtag 'hooray'
  hashtag 'ruby'
  link 'http://codeschool.com'
end
```



our original, clean syntax  
is back!

**RUBY**  
BITS



# METHOD CHAINING

```
tweet_as 'markkendall' do
  mention 'codeschool'
  text('I made a DSL!').hashtag('hooray').hashtag('ruby')
  link 'http://codeschool.com'
end
```

• Ruby requires these parents

```
def text(str)
  @tweet << str
  self
end
```

```
def mention(user)
  @tweet << "@" + user
  self
end
```

```
def hashtag(str)
  @tweet << "#" + str
  self
end
```

• return self

**RUBY**  
BITS