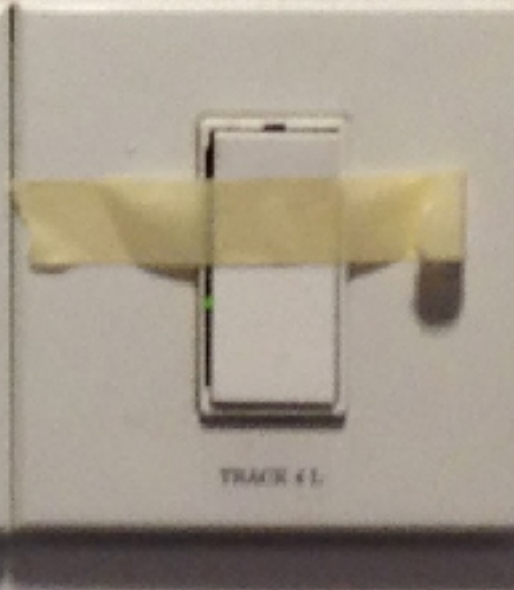
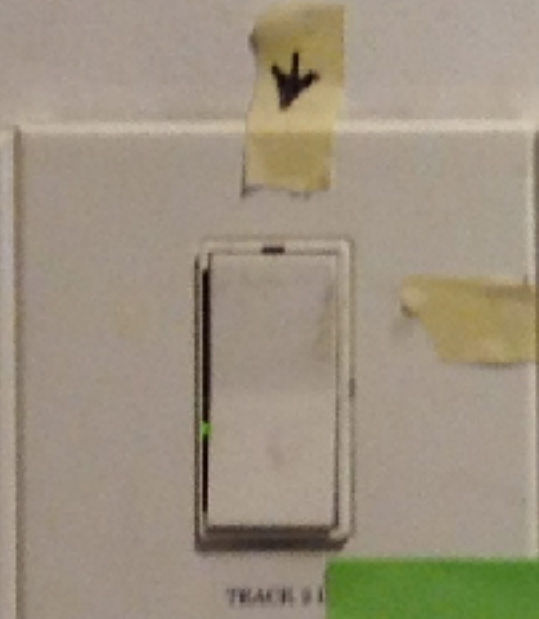
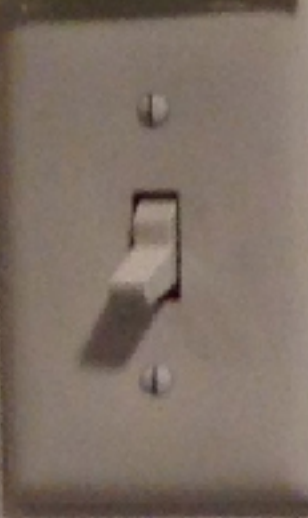


# Meta presentation

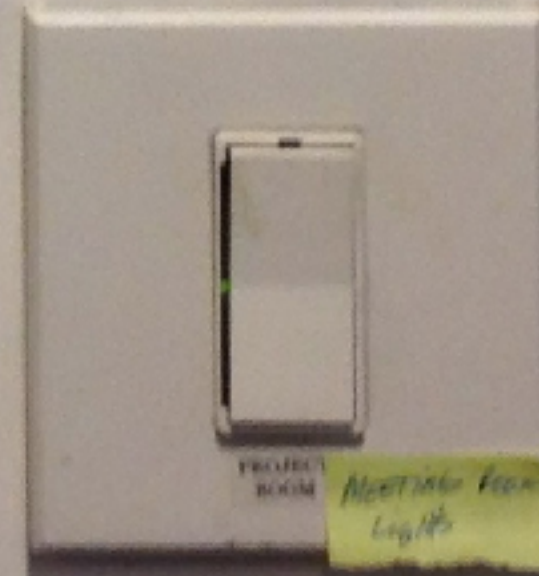
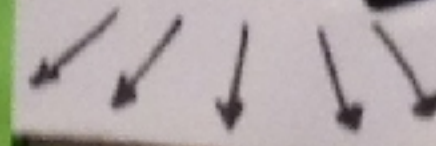
# Switchers

LIGHTS  
OVER WORK  
TABLES



These are not  
the light switches  
you're looking for.  
Look over there →

Don't touch  
these



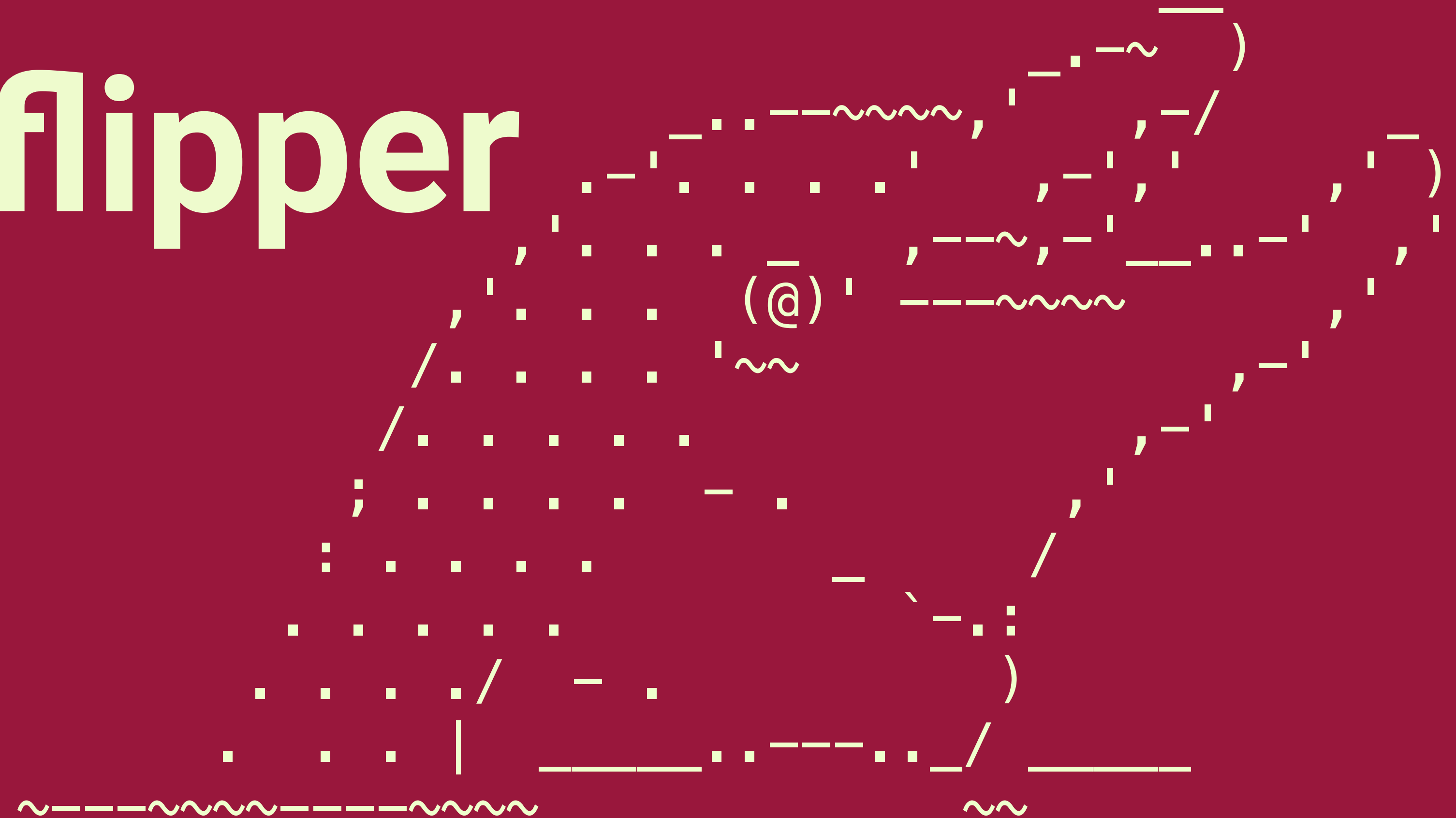
MAIN LIGHTS

PRESS  
← 2x

PRESS  
← 2x

CORR  
AREA

# flipper



```
require 'flipper/adapters/redis'  
client = Redis.new  
adapter = Flipper::Adapters::Redis.new(client)
```

```
require 'flipper/adapters/memory'  
adapter = Flipper::Adapters::Memory.new
```

```
require 'flipper/adapters/active_record'  
adapter = Flipper::Adapters::ActiveRecord.new
```

```
require 'flipper/adapters/memory'  
adapter = Flipper::Adapters::Memory.new
```

```
require 'flipper/adapters/memory'  
adapter = Flipper::Adapters::Memory.new  
  
flipper = Flipper.new(adapter)
```

```
# config/initializers/flipper.rb
require 'flipper/adapters/memory'
adapter = Flipper::Adapters::Memory.new

$flipper = Flipper.new(adapter)
```

```
# config/initializers/flipper.rb
require 'flipper/adapters/memory'
adapter = Flipper::Adapters::Memory.new
$flipper = Flipper.new(adapter)
```

```
# config/routes.rb
Rails.application.routes.draw do
  mount Flipper::UI.app($flipper) => '/flipper'
end
```

```
# config/initializers/flipper.rb
require 'flipper/adapters/memory'
adapter = Flipper::Adapters::Memory.new
$flipper = Flipper.new(adapter)
```

```
# config/routes.rb
Rails.application.routes.draw do
  mount Flipper::UI.app($flipper) => '/flipper'
end
```

```
# your_app/somewhere.rb
if $flipper.enabled?(:search)
  puts 'Yay!'
else
  puts 'Nay!'
end
```

```
# config/initializers/flipper.rb
require 'flipper/adapters/memory'
adapter = Flipper::Adapters::Memory.new
$flipper = Flipper.new(adapter)
```

```
# config/routes.rb
Rails.application.routes.draw do
  mount Flipper::UI.app($flipper) => '/flipper'
end
```

```
# your_app/somewhere.rb
if $flipper.enabled?(:search)
  puts 'Yay!'
else
  puts 'Nay!'
end
```

# Feature class

```
class Feature
```

```
end
```

```
class Feature
  def flipper
  end
end
```

```
class Feature  
  def flipper  
    @flipper ||= Flipper.new(adapter)  
  end  
end
```

```
require 'flipper/adapters/memory'

class Feature

  def flipper
    @flipper ||= begin
      adapter = Flipper::Adapters::Memory.new
      Flipper.new(adapter)
    end
  end

end

end
```

```
require 'flipper/adapters/memory'

class Feature

  def flipper
    @flipper ||= begin
      adapter = Flipper::Adapters::Memory.new
      Flipper.new(adapter)
    end
  end

end

end
```

```
Feature.new.flipper.enabled?(:search)
```

```
require 'flipper/adapters/memory'

class Feature

  def flipper
    @flipper ||= begin
      adapter = Flipper::Adapters::Memory.new
      Flipper.new(adapter)
    end
  end

end

end
```

```
require 'flipper/adapters/memory'

class Feature
  include Singleton

  def flipper
    @flipper ||= begin
      adapter = Flipper::Adapters::Memory.new
      Flipper.new(adapter)
    end
  end
end
```

```
require 'flipper/adapters/memory'

class Feature
  include Singleton

  class << self
    delegate :enabled?, to: 'instance.flipper'
  end

  def flipper
    @flipper ||= begin
      adapter = Flipper::Adapters::Memory.new
      Flipper.new(adapter)
    end
  end
end
```

```
class Feature  
  # code here  
end
```

```
class Feature  
  # code here  
end
```

```
Feature.new.flipper.enabled?(:search)
```

```
class Feature  
  # code here  
end
```

```
# your_app/somewhere.rb  
Feature.enabled?(:search)
```

# Meta plan

# Code highlight Animations

# Browser code

Title

Write

Preview

```
```ruby
class Feature
  # code here
end
```
```

Attach files by dragging & dropping, [selecting](#)

 Styling with Markdown is supported

Title

Write

Preview

```
class Feature
  # code here
end
```

 Styling with Markdown is supported

```
class Feature
  # code here
end
```

```
class Feature  
  # code here  
end
```



**highlight**

```
brew install highlight
```

highlight

```
highlight  
  --out-format=rtf  
  --no-trailing-nl  
  --style=monokai  
  --font=Menlo  
  --font-size=48  
  --syntax=ruby
```

```
highlight  
  --out-format=rtf  
  --no-trailing-nl  
  --style=monokai  
  --font=Menlo  
  --font-size=48  
  --syntax=ruby
```

pbpaste

pbcopy

```
pbpaste | highlight --out-format=rtf --no-trailing-nl  
--style=molokai --font=Menlo --font-size=48  
--syntax=ruby | pbcopy
```



-  Chrome History  
by David Ferguson
-  Convert  
by Dean Jackson
-  Dash  
by Kapeli
-  Google Translate  
by Thomas Hempel
-  Keyword to Script  
by Unknown
-  Lorem Ipsum  
by Jeremy Pippin
-  Lorem Ipsum  
by Brian Gonzalez
-  shortcuts  
by Unknown

## Keyword to Script








rb



rb



```
class Feature  
  # code here  
end
```

# Magic Move

```
class Feature
```

```
end
```

```
class Feature
  def flipper
  end
end
```

Untitled.key — Edited

View

46%

+

Add Slide

Play

Keynote Live

Table

Chart

T

Text

Shape

Media

Comment

Collaborate

Format

Animate

Document

26

27

28

29

30

31

32

33

34

35

36

37

38

39

0

1000

0

1000

Transitions

No Transition Effect

Add an Effect

✓ None

Recent Effects

Magic Move

Magic Move

Preview

Appear & Move

Clothesline

Confetti

Dissolve

Drop

Droplet

Fade Through Color

Grid

Iris

Move In

Push

Reveal

Switch

Wipe

Flip, Spin & Scale

Blinds

Color Planes

Cube

Doorway

Fall

Flip

Flop

Mosaic

Page Flip

Pivot

Reflection

Build Order

Edited ▾



Shape



Media



Comment



Collaborate



Format



Animate



Document

200

## Transitions

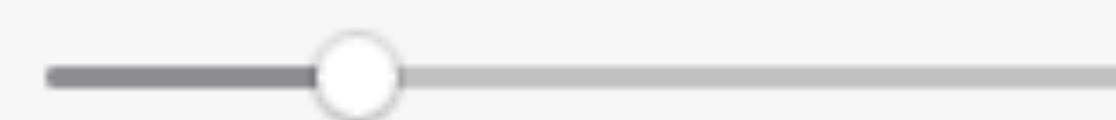


### Magic Move

Change

Preview ▶

#### Duration



2.00 s



☒ Fade Unmatched Objects

#### Match

By Object

By Word

By Character

```
class Feature  
  def flipper  
    @flipper ||= Flipper.new(adapter)  
  end  
end
```

```
require 'flipper/adapters/memory'

class Feature
  include Singleton

  class << self
    delegate :enabled?, to: 'instance.flipper'
  end

  def flipper
    @flipper ||= begin
      adapter = Flipper::Adapters::Memory.new
      Flipper.new(adapter)
    end
  end
end
```

```
class Feature  
  def flipper  
    @flipper ||= Flipper.new(adapter)  
  end  
end
```

```
require 'flipper/adapters/memory'

class Feature
  include Singleton

  class << self
    delegate :enabled?, to: 'instance.flipper'
  end

  def flipper
    @flipper ||= begin
      adapter = Flipper::Adapters::Memory.new
      Flipper.new(adapter)
    end
  end
end
```

```
class Feature  
  def flipper  
    @flipper ||= Flipper.new(adapter)  
  end  
end
```

# Continuity



**Last note**

# Curiosity Killed

[www.colourlovers.com](http://www.colourlovers.com)

# Thank you

**Anton K.**  
**@antulik**