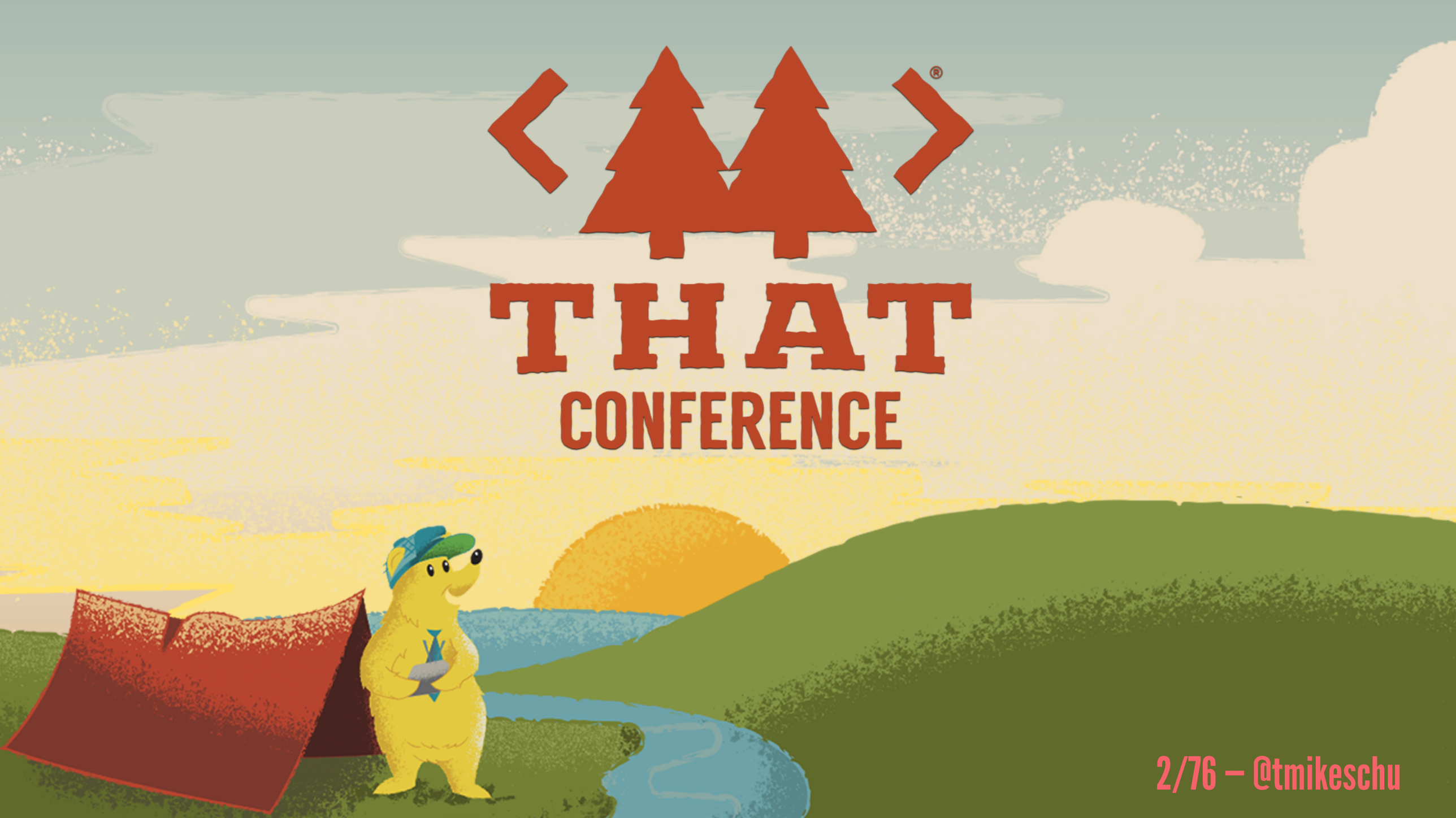
# GET OFF MY GRASS

## WEEDING OUT LOGIC WITH PRESENTERS

Mike Schutte
8/8/19
THAT Conference


THAT
CONFERENCE

# THANK YOU, THAT CONFERENCE SPONSORS!

CUNA MUTUAL GROUP

Northwestern Mutual®

NVISIA

spr

DataRobot

TREK

arms® BUSINESS SOLUTIONS

symplr®

okta

Microsoft

- @tmikeschu (    )

- @tmikeschu (🍵🎤👞)
- 

- @tmikeschu (🍵🎤👞)
- 
- 

- @tmikeschu (🍵🎤👞)
- 
- 
- 

- @tmikeschu (🍵🎤👞)
- 
- 
-   
-  

- @tmikeschu (🍵🎤👞)
- 🏡
- 🖼️
- 🎸🎤🎹
- 🏃🚲
- TED

Questions along the way?

@tmikeschu #getOffMyGrass

# FWBAT

friendgineers will be able to...

- Explain what a presenter object is

- Explain what a presenter object is
- Explain why a presenter object is useful

- Explain what a presenter object is
- Explain why a presenter object is useful
- Detect when a presenter is the tool for the job

- Explain what a presenter object is
- Explain why a presenter object is useful
- Detect when a presenter is the tool for the job
- Expound upon the virtues of the Ruby language

# ROADMAP

# The View Layer

- The View Layer
- The Presenter Pattern

- The View Layer
- The Presenter Pattern
- Defining Dependencies

- The View Layer
- The Presenter Pattern
- Defining Dependencies
- Conditional Rendering

- The View Layer
- The Presenter Pattern
- Defining Dependencies
- Conditional Rendering
- Rendering Collections

- The View Layer
- The Presenter Pattern
- Defining Dependencies
- Conditional Rendering
- Rendering Collections
- Message passing

- The View Layer
- The Presenter Pattern
- Defining Dependencies
- Conditional Rendering
- Rendering Collections
- Message passing
- Testing

# THE VIEW LAYER

# THE VIEW LAYER

- MVC

# THE VIEW LAYER

- MVC
- Modern JS libraries

# THE VIEW LAYER

- MVC
- Modern JS libraries
- HTML

# THE VIEW LAYER

- MVC
- Modern JS libraries
- HTML
- JSON

# THE PRESENTER PATTERN

# THE PRESENTER PATTERN

- Controller gathers model data and

# THE PRESENTER PATTERN

- Controller gathers model data and
- Controller instantiates a presenter object with model data

# THE PRESENTER PATTERN

- Controller gathers model data and
- Controller instantiates a presenter object with model data
- View sends messages to the presenter for all logical needs

# CONTROLLER CONSTRUCTS VIEW CONSUMES

```ruby
# app/controllers/quidditch_controller.rb
class QuidditchController
  def index
    @presenter = QuidditchPresenter.present(players: Player, scores: Score)
  end
end
```

```ruby
# app/presenters/quidditch_presenter.rb
class QuidditchPresenter
  def self.present(players:, scores:)
    # ...
  end

  def leading_scorers
    # ...
  end
end
```

```
<!-- # app/views/quidditch/index.html.erb -->
<div>
  <% @presenter.leading_scorers.each do |leading_scorer| %>
    ...
  <% end %>
</div>
```

# NOT JUST SERVER-SIDE TEMPLATES!

```
class QuidditchContainer extends React.Component {
  state = { players: [], scores: [] };

  componentDidMount() {
    PlayerService.get().then(players => {
      this.setState({ players });
    });
    ScoresService.get().then(scores => {
      this.setState({ scores });
    });
  }
  render() {
    const { players, scores } = this.state;
    return <QuidditchScores {...quidditchPresenter({ players, scores })} />;
  }
}
```

```
function quidditchPresenter({ players, scores }) {
  return {
    leadingScorers: /* ... */
  };
}
```

```
function QuidditchScores({ leadingScorers }) {
  return (
    <div>
      {leadingScorers.map(leadingScorer => (
        <div>...</div>
      ))}
    </div>
  );
}
```

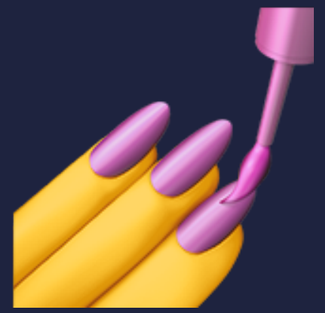
# WHAT

A presenter object encapsulates logic (e.g., boolean and transformation) needed for building a particular view.

# WHY

# CONTROLLER: INTEGRATION

# VIEW: CONTENT AND STYLE



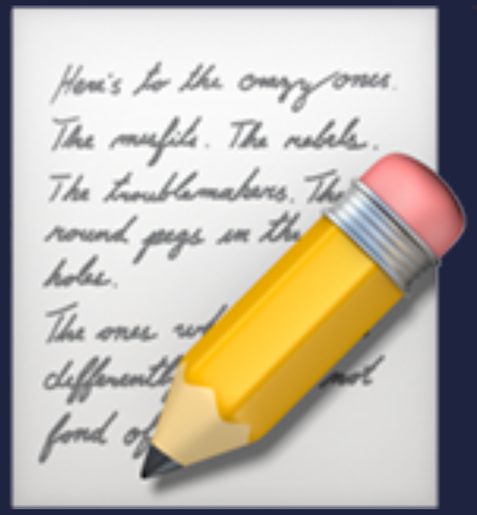
# VIEW: DECLARATIVE INTERFACE 

# VIEW: SIGNAL TO NOISE 

# VIEW: EXPLICIT DEPENDENCIES 

# UNIT TEST ✅

# LESSONS FROM THE FIELD



# CONTEXT

# CONTEXT

- Event highlights for each attendee of an event

# CONTEXT

- Event highlights for each attendee of an event
- Same general layout for all events

# CONTEXT

- Event highlights for each attendee of an event
- Same general layout for all events
- Unique branding for each event

# CONTEXT

- Event highlights for each attendee of an event
- Same general layout for all events
- Unique branding for each event
- Various data sources

# CONTEXT

- Event highlights for each attendee of an event
- Same general layout for all events
- Unique branding for each event
- Various data sources
- Precedence: copy and paste previous event

- 84-line controller action, 0 helper methods

- 84-line controller action, 0 helper methods
- 23 instance variables used by view

- 84-line controller action, 0 helper methods
- 23 instance variables used by view
- 3 view directories with basically identical code

- 84-line controller action, 0 helper methods
- 23 instance variables used by view
- 3 view directories with basically identical code
- 7 &&, 11 | |

- 84-line controller action, 0 helper methods
- 23 instance variables used by view
- 3 view directories with basically identical code
- 7 &&, 11 ||
- 26 `if` statements, 20 ternary expressions

- 84-line controller action, 0 helper methods
- 23 instance variables used by view
- 3 view directories with basically identical code
- 7 &&, 11 ||
- 26 `if` statements, 20 ternary expressions
- 5 cases of manually iterated markup

- 84-line controller action, 0 helper methods
- 23 instance variables used by view
- 3 view directories with basically identical code
- 7 &&, 11 ||
- 26 `if` statements, 20 ternary expressions
- 5 cases of manually iterated markup
- 20+ Law of Demeter violations

- 84-line controller action, 0 helper methods
- 23 instance variables used by view
- 3 view directories with basically identical code
- 7 &&, 11 | |
- ~~26 if statements,~~ 20 ternary expressions
- 5 cases of manually iterated markup
- 20+ Law of Demeter violations

# POST-PRESENTER...

- ~~84~~ 12-line controller action, ~~0~~ 16 helper methods

- ~~84~~ 12-line controller action, ~~0~~ 16 helper methods
- ~~23~~ 1 instance variable used by view

- ~~84~~ 12-line controller action, ~~0~~ 16 helper methods
- ~~23~~ 1 instance variable used by view
- 1 view template with shared partials

- ~~84~~ 12-line controller action, ~~0~~ 16 helper methods
- ~~23~~ 1 instance variable used by view
- 1 view template with shared partials
- ~~7~~ 5 &&, ~~11~~ 1 | |

- ~~84~~ 12-line controller action, ~~0~~ 16 helper methods
- ~~23~~ 1 instance variable used by view
- 1 view template with shared partials
- ~~7~~ 5 `&&`, ~~11~~ 1 `||`
- 19 `if`, 6 `elsif`, ~~20~~ 0 ternaries

- ~~84~~ 12-line controller action, ~~0~~ 16 helper methods
- ~~23~~ 1 instance variable used by view
- 1 view template with shared partials
- ~~7~~ 5 `&&`, ~~11~~ 1 `||`
- 19 `if`, 6 `elsif`, ~~20~~ 0 ternaries
- ~~5~~ 0 cases of manually iterated markup

- ~~84~~ 12-line controller action, ~~0~~ 16 helper methods
- ~~23~~ 1 instance variable used by view
- 1 view template with shared partials
- ~~7~~ 5 `&&`, ~~11~~ 1 `||`
- 19 `if`, 6 `elsif`, ~~20~~ 0 ternaries
- ~~5~~ 0 cases of manually iterated markup
- ~~20+~~ 1 Law of Demeter violation

# DEFINING DEPENDENCIES

```
<%=
  render "frontend/highlights/#{params[:event_slug].upcase}/index",
  locals: {
    activity_names: @activity_names,
    conf_activities: @conf_activities,
    conf_connections: @conf_connections,
    conf_general: @conf_general,
    conf_hearts: @conf_hearts,
    conf_messages: @conf_messages,
    dinners: @dinners,
    event_photos: @event_photos,
    faceoff: @faceoff,
    lunches: @lunches,
    photo_ids: @photo_ids,
    public_token: @public_token,
    session_list: @session_list,
    t19_registrant: @t19_registrant,
    t20_registrant: @t20_registrant,
    workshops: @workshops,
    sunday: @sunday,
    monday: @monday,
    tuesday: @tuesday,
    wednesday: @wednesday,
    thursday: @thursday,
    friday: @friday,
    saturday: @saturday
  }
%>
```

```
def show
  @presenter = Frontend::HighlightsPresenter.present(
    json_resources.merge({
      event_config: event_config,
      event_titles: event_titles,
      eventster: eventster,
      faceoff_data: faceoff_data,
      view_more_photos_url: view_more_photos_url,
    })
  )
  render :layout => 'layouts/highlights'
end
```

```
def show
  @presenter = Frontend::HighlightsPresenter.present(
    json_resources.merge({
      event_config: event_config,
      event_titles: event_titles,
      eventster: eventster,
      faceoff_data: faceoff_data,
      view_more_photos_url: view_more_photos_url,
    })
  )
  render :layout => 'layouts/highlights'
end
```

```
def show
  @presenter = Frontend::HighlightsPresenter.present(
    json_resources.merge({
      event_config: event_config,
      event_titles: event_titles,
      eventster: eventster,
      faceoff_data: faceoff_data,
      view_more_photos_url: view_more_photos_url,
    })
  )
  render :layout => 'layouts/highlights'
end
```

```
def self.present(
  activities:,
  connections:,
  event_config:,
  event_titles:,
  eventster:,
  faceoff_data:,
  hearts:,
  view_more_photos_url:
)
  # initialize/construct
end
```

# CONDITIONAL RENDERING

# SEPARATE THE NEED FOR A TEST FROM THE TEST ITSELF

# SEPARATE THE STATEMENT FROM THE EXPRESSION

```
<% if locals[:workshops] > 0 || locals[:dinners] > 0 || locals[:lunches] > 0 || locals[:sunday].any? || locals[:monday].any? || locals[:tuesday].any? || locals[:wednesday].any? || locals[:thursday].any? || locals[:friday].any? || locals[:saturday].any? %>
  <!-- ... -->
<% end %>
```

```
<% if locals[:workshops] > 0 ||
      locals[:dinners] > 0 ||
      locals[:lunches] > 0 ||
      locals[:sunday].any? ||
      locals[:monday].any? ||
      locals[:tuesday].any? ||
      locals[:wednesday].any? ||
      locals[:thursday].any? ||
      locals[:friday].any? ||
      locals[:saturday].any? %>
  <!-- ... -->
<% end %>
```

VS.

```ruby
# presenter
def show_activities?
  @show_activities ||= went_to_workshops? || went_to_dinners? || went_to_lunches?
end
```

```erb
<!-- view -->
<% if @presenter.show_activities? %>
  <!-- ... -->
<% end %>
```

# ITERATION LOGIC

```
<div class="flex flex-wrap tc-ns">
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      54
    </div>
    <div class="f6 ph3">
      Countries represented
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      105
    </div>
    <div class="f6 ph3">
      Speakers
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      44
    </div>
    <div class="f6 ph3">
      Discovery Sessions
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      23
    </div>
    <div class="f6 ph3">
      Exhibits + social spaces
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      1,411
    </div>
    <div class="f6 ph3">
      Conversations with Gigi
    </div>
  </div>
</div>
```

```
<div class="flex flex-wrap tc-ns">
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      54
    </div>
    <div class="f6 ph3">
      Countries represented
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      105
    </div>
    <div class="f6 ph3">
      Speakers
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      44
    </div>
    <div class="f6 ph3">
      Discovery Sessions
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      23
    </div>
    <div class="f6 ph3">
      Exhibits + social spaces
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      1,411
    </div>
    <div class="f6 ph3">
      Conversations with Gigi
    </div>
  </div>
</div>
```

```
<div class="flex flex-wrap tc-ns">
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      54
    </div>
    <div class="f6 ph3">
      Countries represented
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      105
    </div>
    <div class="f6 ph3">
      Speakers
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      44
    </div>
    <div class="f6 ph3">
      Discovery Sessions
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      23
    </div>
    <div class="f6 ph3">
      Exhibits + social spaces
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      1,411
    </div>
    <div class="f6 ph3">
      Conversations with Gigi
    </div>
  </div>
</div>
```

```
<div class="flex flex-wrap tc-ns">
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      54
    </div>
    <div class="f6 ph3">
      Countries represented
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      105
    </div>
    <div class="f6 ph3">
      Speakers
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      44
    </div>
    <div class="f6 ph3">
      Discovery Sessions
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      23
    </div>
    <div class="f6 ph3">
      Exhibits + social spaces
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      1,411
    </div>
    <div class="f6 ph3">
      Conversations with Gigi
    </div>
  </div>
</div>
```

```
<div class="flex flex-wrap tc-ns">
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      54
    </div>
    <div class="f6 ph3">
      Countries represented
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      105
    </div>
    <div class="f6 ph3">
      Speakers
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      44
    </div>
    <div class="f6 ph3">
      Discovery Sessions
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      23
    </div>
    <div class="f6 ph3">
      Exhibits + social spaces
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      1,411
    </div>
    <div class="f6 ph3">
      Conversations with Gigi
    </div>
  </div>
</div>
```

```
<div class="flex flex-wrap tc-ns">
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      54
    </div>
    <div class="f6 ph3">
      Countries represented
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      105
    </div>
    <div class="f6 ph3">
      Speakers
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      44
    </div>
    <div class="f6 ph3">
      Discovery Sessions
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      23
    </div>
    <div class="f6 ph3">
      Exhibits + social spaces
    </div>
  </div>
  <div class="w-50 w-20-ns mb3 mb0-ns">
    <div class="f2 f1-l pv2">
      1,411
    </div>
    <div class="f6 ph3">
      Conversations with Gigi
    </div>
  </div>
</div>
```

VS.

```
<div class="flex flex-wrap tc-ns">
  <% @presenter.at_a_glance_stats.each do |(label, value)| %>
    <div class="w-50 w-20-ns mb3 mb0-ns">
      <div class="f2 f1-l pv2">
        <%= value %>
      </div>
      <div class="f6 ph3">
        <%= label %>
      </div>
    </div>
  <% end %>
</div>
```

```
<div class="flex flex-wrap tc-ns">
  <% @presenter.at_a_glance_stats.each do |(label, value)| %>
    <div class="w-50 w-20-ns mb3 mb0-ns">
      <div class="f2 f1-l pv2">
        <%= value %>
      </div>
      <div class="f6 ph3">
        <%= label %>
      </div>
    </div>
  <% end %>
</div>
```

# LIMIT ELEMENT INTERFACE TO SIMPLE MESSAGE PASSING

```
<div class="w-third w-20-ns bg-near-black square cover hide-child relative bg-center"
     style="background-image:url(<%= crushinate(
        locals[:event_photos][4].present? ? locals[:event_photos][4]['url']
                                          : 'https://live.staticflickr.com/65535/33746025918_03319ba6a1_o_d.jpg',
        w: 550, quality:90
      ) %>)">
  <a href="<%= locals[:event_photos][4].present? ? locals[:event_photos][4]['url']
                                                 : 'https://live.staticflickr.com/65535/33746025918_03319ba6a1_o_d.jpg' %>"
     class="db f6 white child v-mid bg-black-50 w-100 h-100 absolute absolute--fill lightbox">
    <div class="vertical-align">
      <img src="<%= image_path 'highlights/expand.svg' %>" />
    </div>
  </a>
</div>
```

# (* 10)

```
<div class="w-third w-20-ns bg-near-black square cover hide-child relative bg-center"
    style="background-image:url(<%= crushinate(
       locals[:event_photos][4].present? ? locals[:event_photos][4]['url']
                                         : 'https://live.staticflickr.com/65535/33746025918_03319ba6a1_o_d.jpg',
       w: 550, quality:90
     ) %>)">
  <a href="<%= locals[:event_photos][4].present? ? locals[:event_photos][4]['url']
                                                 : 'https://live.staticflickr.com/65535/33746025918_03319ba6a1_o_d.jpg' %>"
    class="db f6 white child v-mid bg-black-50 w-100 h-100 absolute absolute--fill lightbox">
    <div class="vertical-align">
     <img src="<%= image_path 'highlights/expand.svg' %>" />
    </div>
  </a>
</div>
```

# (* 10)

VS.

```
<% @presenter.tile_photos.each do |tile_photo| %>
  <div class="<%= tile_photo.css_classes %>"
       style="background-image:url(<%= crushinate(tile_photo.id, w: 550, quality:90) %>)">
     <a href="<%= tile_photo.id %>"
        class="db f6 white child v-mid bg-black-50 w-100 h-100 absolute absolute--fill lightbox">
        <div class="vertical-align">
          <img src="<%= image_path 'highlights/expand.svg' %>" />
        </div>
     </a>
  </div>
<% end %>
```

```
<% @presenter.tile_photos.each do |tile_photo| %>
  <div class="<%= tile_photo.css_classes %>"
       style="background-image:url(<%= crushinate(tile_photo.id, w: 550, quality:90) %>)">
     <a href="<%= tile_photo.id %>"
        class="db f6 white child v-mid bg-black-50 w-100 h-100 absolute absolute--fill lightbox">
        <div class="vertical-align">
          <img src="<%= image_path 'highlights/expand.svg' %>" />
        </div>
     </a>
  </div>
<% end %>
```

```
<% @presenter.tile_photos.each do |tile_photo| %>
  <div class="<%= tile_photo.css_classes %>"
       style="background-image:url(<%= crushinate(tile_photo.id, w: 550, quality:90) %>)">
      <a href="<%= tile_photo.id %>"
         class="db f6 white child v-mid bg-black-50 w-100 h-100 absolute absolute--fill lightbox">
         <div class="vertical-align">
           <img src="<%= image_path 'highlights/expand.svg' %>" />
         </div>
      </a>
  </div>
<% end %>
```

```
<% @presenter.tile_photos.each do |tile_photo| %>
  <div class="<%= tile_photo.css_classes %>"
       style="background-image:url(<%= crushinate(tile_photo.id, w: 550, quality:90) %>)">
     <a href="<%= tile_photo.id %>"
        class="db f6 white child v-mid bg-black-50 w-100 h-100 absolute absolute--fill lightbox">
        <div class="vertical-align">
          <img src="<%= image_path 'highlights/expand.svg' %>" />
        </div>
     </a>
  </div>
<% end %>
```

# TESTING

# MANY PRESENTER METHODS MIGHT NOT NEED UNIT TESTS

```ruby
def show_hearts?
  hearts_enabled? && heart_count > 0
end

def show_more_hearts?
  heart_count > 10
end

def heart_count
  @heart_count ||= hearts.count
end
```

Non-trivial computations or transformations are simple to test

Non-trivial computations or transformations are simple to test

- (i.e., no need for spies, mocks, or stubs)

```ruby
def workshop_days
  default_days = %i(Sunday Monday Tuesday Wednesday Thursday Friday Saturday)
  start = default_days.index(first_day.capitalize.to_sym)
  days = default_days.rotate(start)
  abbreviations = {
    Sun: :Sunday,
    Mon: :Monday,
    Tue: :Tuesday,
    Wed: :Wednesday,
    Thu: :Thursday,
    Fri: :Friday,
    Sat: :Saturday,
  }

  @workshop_days ||= activities
    .group_by { |raw| raw.values.first.second.slice(0, 3) }
    .map { |day, activities| WorkshopDay.new(abbreviations.fetch(day.to_sym), activities) }
    .sort_by { |workshop_day| days.index(workshop_day.to_s.to_sym) }
end
```

```ruby
def workshop_days
  default_days = %i(Sunday Monday Tuesday Wednesday Thursday Friday Saturday)
  start = default_days.index(first_day.capitalize.to_sym)
  days = default_days.rotate(start)
  abbreviations = {
    Sun: :Sunday,
    Mon: :Monday,
    Tue: :Tuesday,
    Wed: :Wednesday,
    Thu: :Thursday,
    Fri: :Friday,
    Sat: :Saturday,
  }

  @workshop_days ||= activities
    .group_by { |raw| raw.values.first.second.slice(0, 3) }
    .map { |day, activities| WorkshopDay.new(abbreviations.fetch(day.to_sym), activities) }
    .sort_by { |workshop_day| days.index(workshop_day.to_s.to_sym) }
end
```

```ruby
def workshop_days
  default_days = %i(Sunday Monday Tuesday Wednesday Thursday Friday Saturday)
  start = default_days.index(first_day.capitalize.to_sym)
  days = default_days.rotate(start)
  abbreviations = {
    Sun: :Sunday,
    Mon: :Monday,
    Tue: :Tuesday,
    Wed: :Wednesday,
    Thu: :Thursday,
    Fri: :Friday,
    Sat: :Saturday,
  }

  @workshop_days ||= activities
    .group_by { |raw| raw.values.first.second.slice(0, 3) }
    .map { |day, activities| WorkshopDay.new(abbreviations.fetch(day.to_sym), activities) }
    .sort_by { |workshop_day| days.index(workshop_day.to_s.to_sym) }
end
```

# REMINDER

Presenters are dependent on some kind of orchestrator for their data

# REMINDER

Presenters are dependent on some kind of orchestrator for their data

- (e.g., controller, container component)

```ruby
# spec/support/presenter_factory.rb
module PresenterFactory
  def self.make_highlights_presenter(options = {})
    titles = OpenStruct.new(
      asset_slug: "TEDSpaceship3000",
      full_title: "TEDSpaceship 3000: A Galaxy Far Far Away",
      title: "TEDSpaceship 3000"
    )

    eventster = OpenStruct.new(
      badge: OpenStruct.new(firstname: "Voldemort")
    )

    Frontend::HighlightsPresenter.present({
      activities: [],
      connections: { old: [], new: [] },
      event_titles: titles,
      event_config: {},
      eventster: eventster,
      faceoff_data: { events: [], interesting: [], personal: [] },
      hearts: [],
      view_more_photos_url: "www.com",
    }.merge(options))
  end
end
```

```ruby
# spec/support/presenter_factory.rb
module PresenterFactory
  def self.make_highlights_presenter(options = {})
    titles = OpenStruct.new(
      asset_slug: "TEDSpaceship3000",
      full_title: "TEDSpaceship 3000: A Galaxy Far Far Away",
      title: "TEDSpaceship 3000"
    )

    eventster = OpenStruct.new(
      badge: OpenStruct.new(firstname: "Voldemort")
    )

    Frontend::HighlightsPresenter.present({
      activities: [],
      connections: { old: [], new: [] },
      event_titles: titles,
      event_config: {},
      eventster: eventster,
      faceoff_data: { events: [], interesting: [], personal: [] },
      hearts: [],
      view_more_photos_url: "www.com",
    }.merge(options))
  end
end
```

```ruby
# spec/support/presenter_factory.rb
module PresenterFactory
  def self.make_highlights_presenter(options = {})
    titles = OpenStruct.new(
      asset_slug: "TEDSpaceship3000",
      full_title: "TEDSpaceship 3000: A Galaxy Far Far Away",
      title: "TEDSpaceship 3000"
    )

    eventster = OpenStruct.new(
      badge: OpenStruct.new(firstname: "Voldemort")
    )

    Frontend::HighlightsPresenter.present({
      activities: [],
      connections: { old: [], new: [] },
      event_titles: titles,
      event_config: {},
      eventster: eventster,
      faceoff_data: { events: [], interesting: [], personal: [] },
      hearts: [],
      view_more_photos_url: "www.com",
    }.merge(options))
  end
end
```

```ruby
# spec/support/presenter_factory.rb
module PresenterFactory
  def self.make_highlights_presenter(options = {})
    titles = OpenStruct.new(
      asset_slug: "TEDSpaceship3000",
      full_title: "TEDSpaceship 3000: A Galaxy Far Far Away",
      title: "TEDSpaceship 3000"
    )

    eventster = OpenStruct.new(
      badge: OpenStruct.new(firstname: "Voldemort")
    )

    Frontend::HighlightsPresenter.present({
      activities: [],
      connections: { old: [], new: [] },
      event_titles: titles,
      event_config: {},
      eventster: eventster,
      faceoff_data: { events: [], interesting: [], personal: [] },
      hearts: [],
      view_more_photos_url: "www.com",
    }.merge(options))
  end
end
```

```ruby
describe "#workshop_days" do
  let(:activities) {
    JSON.parse(
      Rails.root.join(
        *%w[spec fixtures frontend highlights s3_resources.json]
      ).read,
      symbolize_names: true
    ).fetch(:activities)
  }

  let(:subject) { PresenterFactory.make_highlights_presenter(activities: activities) }

  it "returns an ordered collection of days and their activities" do
    actual = subject.workshop_days
    expect(actual.first.to_s).to eq("Tuesday")
    expect(actual.first.activities.count).to eq(2)
    expect(actual.second.to_s).to eq("Friday")
    expect(actual.second.activities.count).to eq(1)
  end
  # ...
```

```ruby
describe "#workshop_days" do
  let(:activities) {
    JSON.parse(
      Rails.root.join(
        *%w[spec fixtures frontend highlights s3_resources.json]
      ).read,
      symbolize_names: true
    ).fetch(:activities)
  }

  let(:subject) { PresenterFactory.make_highlights_presenter(activities: activities) }

  it "returns an ordered collection of days and their activities" do
    actual = subject.workshop_days
    expect(actual.first.to_s).to eq("Tuesday")
    expect(actual.first.activities.count).to eq(2)
    expect(actual.second.to_s).to eq("Friday")
    expect(actual.second.activities.count).to eq(1)
  end
  # ...
```

```ruby
describe "#workshop_days" do
  let(:activities) {
    JSON.parse(
      Rails.root.join(
        *%w[spec fixtures frontend highlights s3_resources.json]
      ).read,
      symbolize_names: true
    ).fetch(:activities)
  }

  let(:subject) { PresenterFactory.make_highlights_presenter(activities: activities) }

  it "returns an ordered collection of days and their activities" do
    actual = subject.workshop_days
    expect(actual.first.to_s).to eq("Tuesday")
    expect(actual.first.activities.count).to eq(2)
    expect(actual.second.to_s).to eq("Friday")
    expect(actual.second.activities.count).to eq(1)
  end
  # ...
```

```ruby
describe "#workshop_days" do
  let(:activities) {
    JSON.parse(
      Rails.root.join(
        *%w[spec fixtures frontend highlights s3_resources.json]
      ).read,
      symbolize_names: true
    ).fetch(:activities)
  }

  let(:subject) { PresenterFactory.make_highlights_presenter(activities: activities) }

  it "returns an ordered collection of days and their activities" do
    actual = subject.workshop_days
    expect(actual.first.to_s).to eq("Tuesday")
    expect(actual.first.activities.count).to eq(2)
    expect(actual.second.to_s).to eq("Friday")
    expect(actual.second.activities.count).to eq(1)
  end
  # ...
```

```ruby
describe "#workshop_days" do
  let(:activities) {
    JSON.parse(
      Rails.root.join(
        *%w[spec fixtures frontend highlights s3_resources.json]
      ).read,
      symbolize_names: true
    ).fetch(:activities)
  }

  let(:subject) { PresenterFactory.make_highlights_presenter(activities: activities) }

  it "returns an ordered collection of days and their activities" do
    actual = subject.workshop_days
    expect(actual.first.to_s).to eq("Tuesday")
    expect(actual.first.activities.count).to eq(2)
    expect(actual.second.to_s).to eq("Friday")
    expect(actual.second.activities.count).to eq(1)
  end
  # ...
```

```
  context "when first_day is configured" do
    let(:subject) {
      PresenterFactory.make_highlights_presenter(
        activities: activities,
        event_config: {
          first_day: "friday"
        }
      )
    }

    it "orders relative to that day" do
      actual = subject.workshop_days
      expect(actual.first.to_s).to eq("Friday")
      expect(actual.first.activities.count).to eq(1)
      expect(actual.second.to_s).to eq("Tuesday")
      expect(actual.second.activities.count).to eq(2)
    end
  end
end
```

```
  context "when first_day is configured" do
    let(:subject) {
      PresenterFactory.make_highlights_presenter(
        activities: activities,
        event_config: {
          first_day: "friday"
        }
      )
    }

    it "orders relative to that day" do
      actual = subject.workshop_days
      expect(actual.first.to_s).to eq("Friday")
      expect(actual.first.activities.count).to eq(1)
      expect(actual.second.to_s).to eq("Tuesday")
      expect(actual.second.activities.count).to eq(2)
    end
  end
end
```

```
  context "when first_day is configured" do
    let(:subject) {
      PresenterFactory.make_highlights_presenter(
        activities: activities,
        event_config: {
          first_day: "friday"
        }
      )
    }

    it "orders relative to that day" do
      actual = subject.workshop_days
      expect(actual.first.to_s).to eq("Friday")
      expect(actual.first.activities.count).to eq(1)
      expect(actual.second.to_s).to eq("Tuesday")
      expect(actual.second.activities.count).to eq(2)
    end
  end
end
```

```
  context "when first_day is configured" do
    let(:subject) {
      PresenterFactory.make_highlights_presenter(
        activities: activities,
        event_config: {
          first_day: "friday"
        }
      )
    }

    it "orders relative to that day" do
      actual = subject.workshop_days
      expect(actual.first.to_s).to eq("Friday")
      expect(actual.first.activities.count).to eq(1)
      expect(actual.second.to_s).to eq("Tuesday")
      expect(actual.second.activities.count).to eq(2)
    end
  end
end
```

# REVIEW

# The View Layer

- The View Layer
- The Presenter Pattern

- The View Layer
- The Presenter Pattern
- Defining Dependencies

- The View Layer
- The Presenter Pattern
- Defining Dependencies
- Conditional Rendering

- The View Layer
- The Presenter Pattern
- Defining Dependencies
- Conditional Rendering
- Rendering Collections

- The View Layer
- The Presenter Pattern
- Defining Dependencies
- Conditional Rendering
- Rendering Collections
- Message passing

- The View Layer
- The Presenter Pattern
- Defining Dependencies
- Conditional Rendering
- Rendering Collections
- Message passing
- Testing

# FWBAT

# WHAT IS A PRESENTER OBJECT?

a) a laser pointer

b) an object that encapsulates logic needed for building a view

c) the view layer of an application

# WHAT IS A PRESENTER OBJECT?

a) a laser pointer

b) an object that encapsulates logic needed for building a view ✅

c) the view layer of an application

# WHY IS A PRESENTER OBJECT USEFUL?

a) controller focuses on integration

b) view focuses on content and styling

c) explicitly defines the view's dependencies

d) easy to unit test

e) all the above

# WHY IS A PRESENTER OBJECT USEFUL?

a) controller focuses on integration

b) view focuses on content and styling

c) explicitly defines the view's dependencies

d) easy to unit test

e) all the above ✅

# WHEN MIGHT A PRESENTER BE THE TOOL FOR THE JOB?

a) computational logic in a controller (-like object)

b) repetitive markup

c) complex logic in the view

d) multiple objects used in the view

e) all the above

# WHEN MIGHT A PRESENTER BE THE TOOL FOR THE JOB?

a) computational logic in a controller (-like object)

b) repetitive markup

c) complex logic in the view

d) multiple objects used in the view

e) all the above ✅

# WHAT DO YOU LOVE ABOUT THE RUBY LANGUAGE?

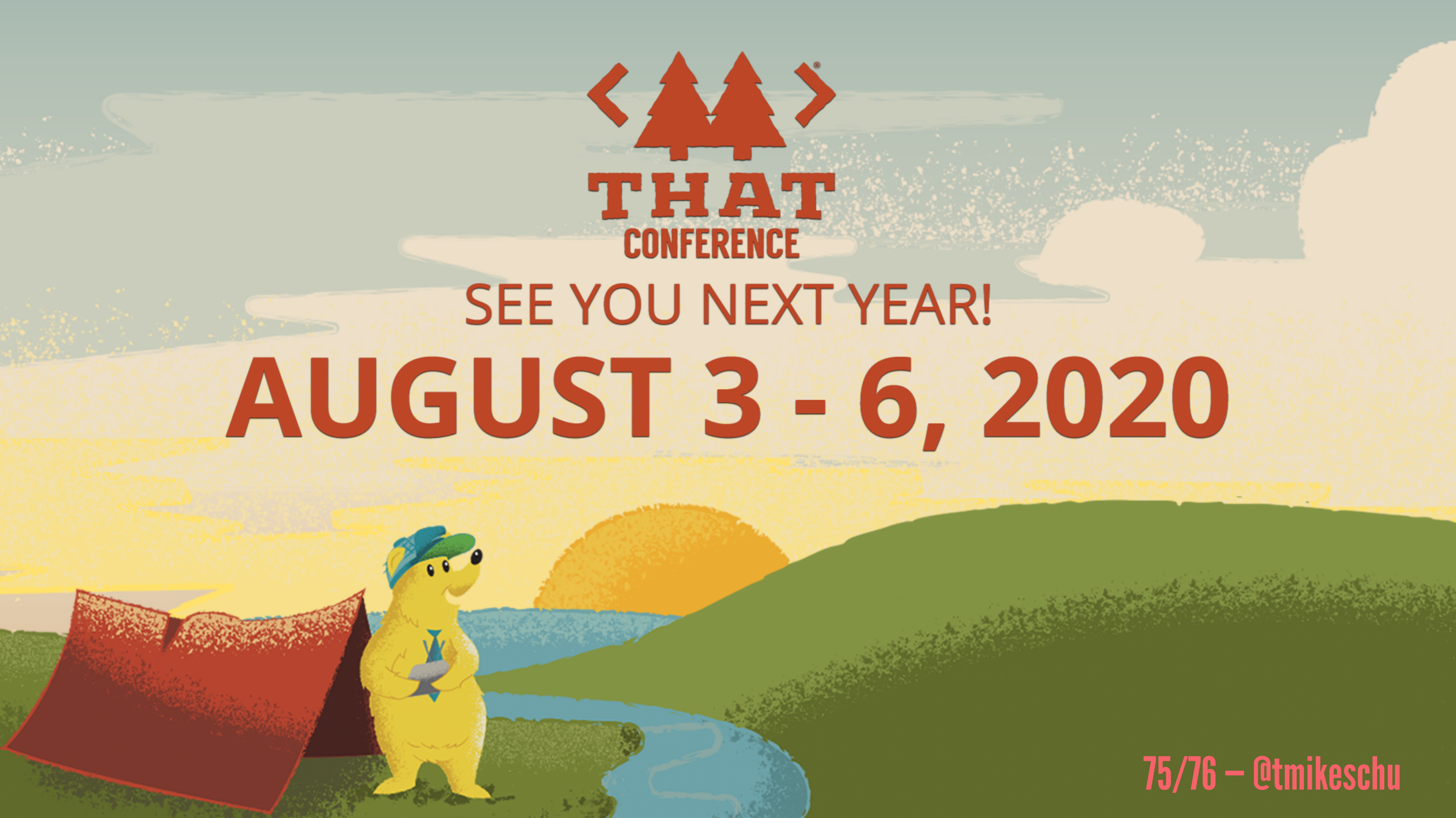
a) yes

b) I choose to fail the quiz

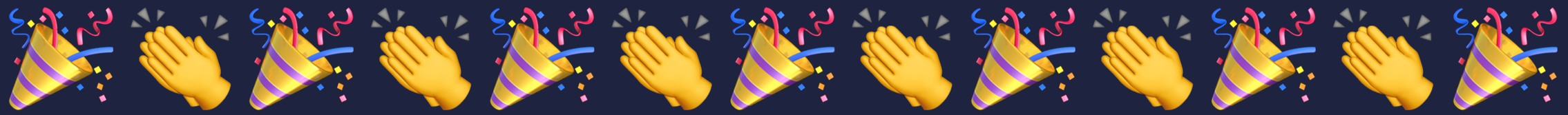
# WHAT DO YOU LOVE ABOUT THE RUBY LANGUAGE?

a) yes ✅

b) I choose to fail the quiz


THAT
CONFERENCE
SEE YOU NEXT YEAR!
AUGUST 3 - 6, 2020
75/76 – @tmikeschu

# THANK YOU!



Questions/comments: @tmikeschu #getOffMyGrass