

Лекция 12

История и состояние

Внедрение фреймворков для

фронта

Интеграции

План

История

Bootstrap + Webpacker

Боты для уведомлений

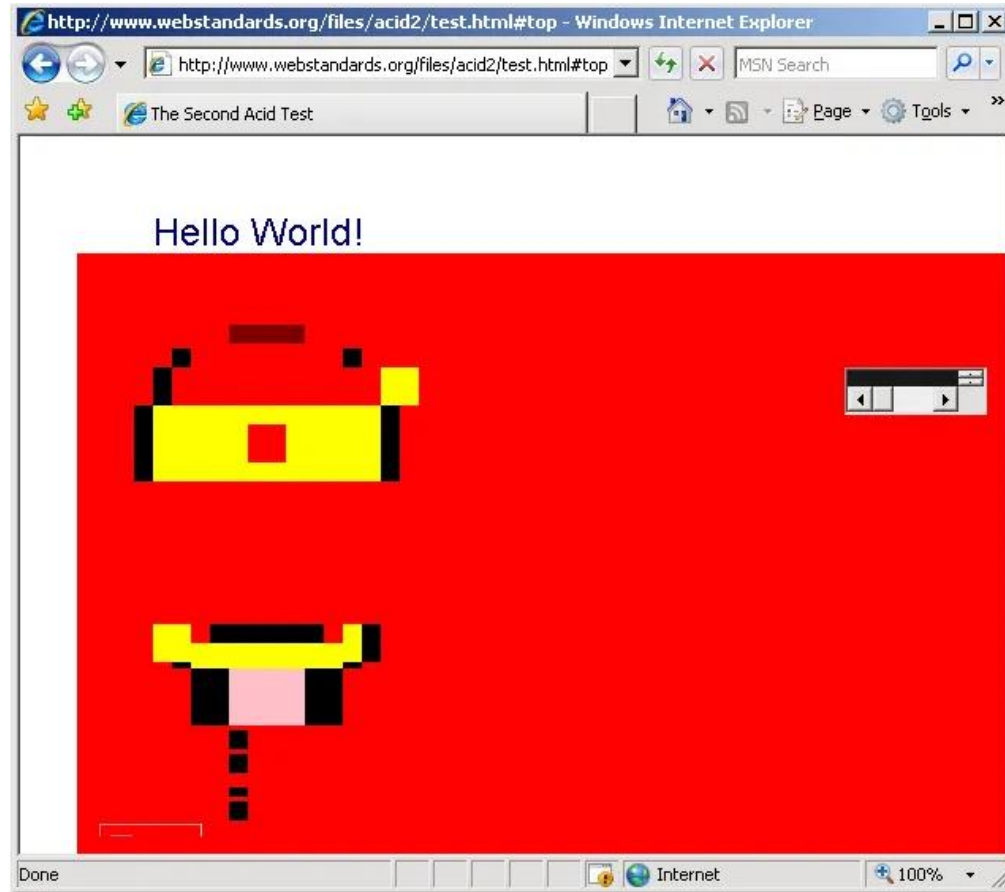
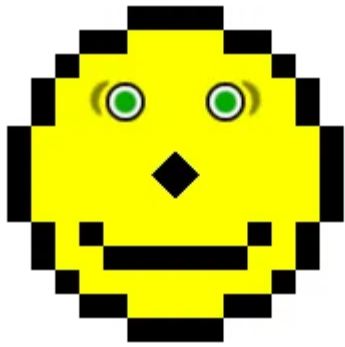
Кто стандартизирует технологии?

- HTML & CSS - World Wide Web Consortium (W3C)
- Mocha -> LifeScript -> JavaScript
Ecma International

История

- **1990–1995** : HTML, CSS, JavaScript
- **1996–1999** : Война браузеров
- **2000–2004** : CSS Frameworks, jQuery, управление пакетами
- **2005–2009** : W3C стандартизирует web, адаптивный дизайн
- **2010–2015**: Backbone, Ember, AngularJS, React, Angular, Vue.
- **2016–2018**: Платформы для фреймворков

Стандартизация верстки (ACID)



Появление маленьких экранов

- 2006 год – первый мобильный браузер проходит ACID
- 2007 – iPhone, затем волна Android
- 2011 – Twitter Bootstrap, 2017 – Bootstrap 4
- 2011 – Foundation

Document Object Model

```
<!DOCTYPE html>
<html lang="ru"> прокручиваемый
  ▶ <head> ... </head>
  ▼ <body>
    ▶ <header class="FM_header"> ... </header>
    ▶ <article class="FM_content" itemscope="" itemtype="http://schema.org/Article"> ... </article>
    ▶ <footer class="FM_footer"> ... </footer>
  </body>
</html>
```

PreProcessor - CSS - PostProcessor

```
<style>
```

```
p.one {
```

```
    border-style: solid;
```

```
    border-color: #0000ff;
```

```
}
```

```
p.two {
```

```
    border-style: solid;
```

```
    border-color: #ff0000 #0000ff;
```

```
}
```

```
</style>
```


PreProcessing

- SASS – включен в HAML, написан на Ruby (больше нет)

#header

background: #FFFFFF

.error

color: #FF0000

a

text-decoration: none

&:hover

text-decoration: underline

```
#header {
```

```
  background: #FFFFFF;
```

```
}
```

```
#header .error {
```

```
  color: #FF0000;
```

```
}
```

```
#header a {
```

```
  text-decoration: none;
```

```
}
```

```
#header a:hover {
```

```
  text-decoration: underline;
```

```
}
```

SASS

```
!primary-color= hotpink

=border-radius(!radius)
  -webkit-border-radius= !radius
  -moz-border-radius= !radius
  border-radius= !radius

.my-element
  color= !primary-color
  width= 100%
  overflow= hidden

.my-other-element
  +border-radius(5px)
```

Окончательное превращение в SCSS

```
$primary-color: hotpink;
```

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  border-radius: $radius;  
}
```

```
.my-element {  
  color: $primary-color;  
  width: 100%;  
  overflow: hidden;  
}
```

```
.my-other-element {  
  @include border-radius(5px);  
}
```

PostCSS

```
const postcss = require('postcss')
const autoprefixer = require('autoprefixer')
const precss = require('precss')

const processor = postcss([autoprefixer, precss])

processor.process('a {}', { from: './app.css', to: './app.build.css' })

  .then(result => {
    console.log(result.css)
    for ( let message of result.warnings() ) {
      console.warn(message.toString())
    }
  })
```

Vanilla JS

```
const elem = document.querySelector('#some-elem');
```

```
let html = elem.innerHTML;
```

```
elem.innerHTML = 'We can dynamically change the HTML.  
We can even include HTML elements  
like <a href="#">this link</a>.';
```

jQuery

```
$('#some-elem').html(linkText);
```

jQuery Selectors














```
$("tr:odd")
```

```
$(" :contains('Hello')")
```

```
$("p:nth-last-of-type(2)")
```

jQuery умирает?

Top In JavaScript Library Usage Distribution in the Top 1 Million Sites

Technology	Websites	%
 jQuery	800,912	80.09
 Bootstrap.js	167,993	16.8
 Moment JS	53,066	5.31
 Yahoo User Interface	37,376	3.74
 lodash	35,468	3.55
 Prototype	18,264	1.83
 MooTools	16,855	1.69
 script.aculo.us	16,014	1.6
 Enquire JS	15,319	1.53
 React	14,210	1.42
 Backbone.js	10,831	1.08
 jsTimezoneDetect	9,987	1
 KnockoutJS	8,532	0.85

Фреймворки

- Большой объем работы с DOM
- SPA
- Сложная логика на клиенте
- Angular, React, Vue, Svelte...

Диалекты

- CoffeeScript
- TypeScript
- Сам EcmaScript 6

Bootstrap

Pricing

Quickly build an effective pricing table for your potential customers with this Bootstrap example. It's built with default Bootstrap components and utilities with little customization.

Free	Pro	Enterprise
\$0 / mo 10 users included 2 GB of storage Email support Help center access Sign up for free	\$15 / mo 20 users included 10 GB of storage Priority email support Help center access Get started	\$29 / mo 30 users included 15 GB of storage Phone and email support Help center access Contact us

Company name

[Features](#) [Enterprise](#) [Support](#) [Pricing](#)

[Sign up](#)

Pricing

Quickly build an effective pricing table for your potential customers with this Bootstrap example. It's built with default Bootstrap components and utilities with little customization.

Free
\$0 / mo 10 users included 2 GB of storage Email support Help center access Sign up for free

Будущее

- Конструкторы
- Статика (Gatsby, Jekyll)

```
gem install jekyll bundler
```

```
jekyll new myblog
```

```
bundle exec jekyll serve
```

Welcome to Jekyll!

Dec 6, 2019

You'll find this post in your `_posts` directory. Go ahead and edit it and re-build the site to see your changes. You can rebuild the site in many different ways, but the most common way is to run `jekyll serve`, which launches a web server and auto-regenerates your site when a file is updated.

Jekyll requires blog post files to be named according to the following format:

```
YEAR-MONTH-DAY-title.MARKUP
```

Where `YEAR` is a four-digit number, `MONTH` and `DAY` are both two-digit numbers, and `MARKUP` is the file extension representing the format used in the file. After that, include the necessary front matter. Take a look at the source for this post to get an idea about how it works.

Jekyll also offers powerful support for code snippets:

```
def print_hi(name)
  puts "Hi, #{name}"
end
print_hi('Tom')
#=> prints 'Hi, Tom' to STDOUT.
```

Check out the [Jekyll docs](#) for more info on how to get the most out of Jekyll. File all bugs/feature requests at [Jekyll's GitHub repo](#). If you have questions, you can ask them on [Jekyll Talk](#).

Проблемы!

- Webpacker по умолчанию собирает JS, но не CSS!
- Библиотеки содержат и то, и другое
- Есть ненулевая задержка в сборке ассетов

БЭМ Блок, Элемент, Модификатор

- Разделение интерфейса на блоки
- Разделение логики на отдельные части
- Понятно, где искать причину и куда писать код

Термины

- Блок – Функционально независимый компонент страницы, который может быть повторно использован.
- Элемент – Составная часть блока, которая не может использоваться в отрыве от него.
- Модификатор – Сущность, определяющая внешний вид, состояние или поведение блока либо элемента.

Соглашение об именовании

menuItemvisible

menu-item-visible

menuItemVisible

menu_item_visible

БЭМ-соглашение

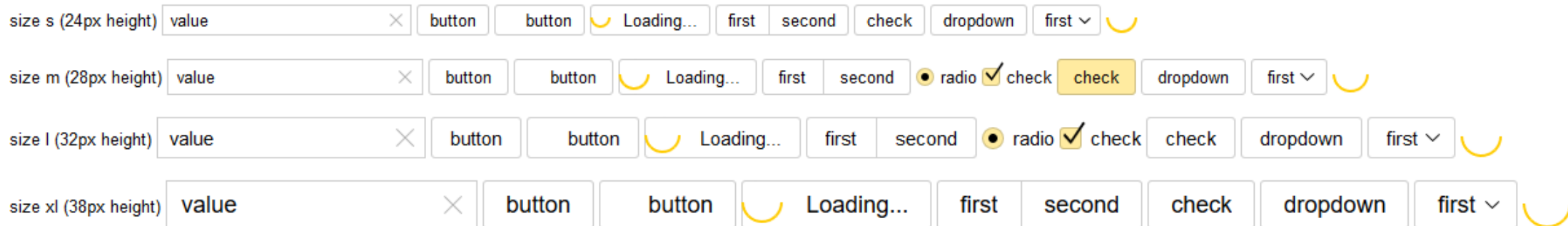
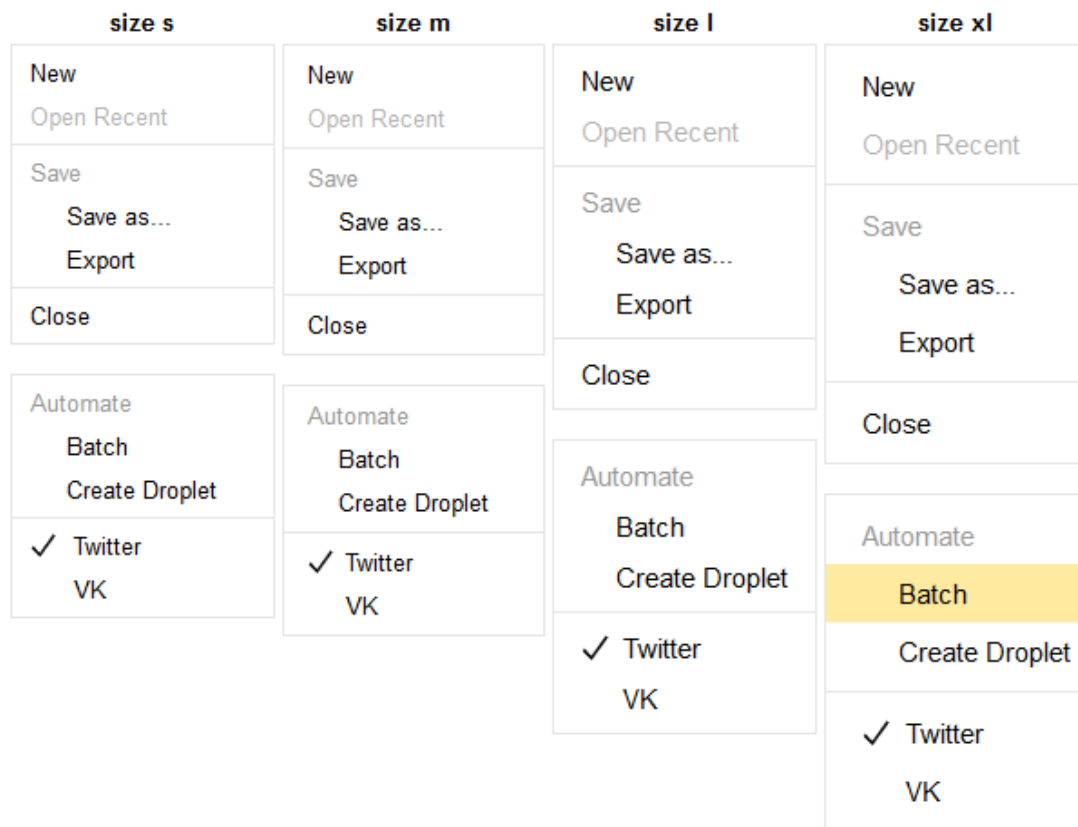
`block-name__elem-name_mod-name_mod-val`

added 786 packages from 604 contributors
and audited 9633 packages in 192.752s
found 60 vulnerabilities (36 low, 12
moderate, 12 high)
run `npm audit fix` to fix them, or `npm
audit` for details

header content goes here

This is a demo page to show blocks from bem-components library.
Feel free to replace it with your own content in `desktop.bundles/index/index.bemjson.js`.
For more info about BEM check out [bem.info](#).

islands theme



Переносим CSS на Webpacker

1. Перемещаем все `.css` в `app/javascript/packs/stylesheets/`
2. Исправляем `applications.scss`

```
@import "card";  
@import "table";  
@import "actiontext";
```

Добавляем Bootstrap

- `yarn add bootstrap@4.3.1 jquery popper.js`
- В `config/webpack/environment.js`

```
const { environment } = require('@rails/webpacker');
```

```
const webpack = require('webpack');  
environment.plugins.append('Provide', new webpack.ProvidePlugin({  
  $: 'jquery',  
  jQuery: 'jquery',  
  Popper: ['popper.js', 'default']  
}));
```

```
module.exports = environment;
```

Добавляем font-awesome

- `yarn add @fontawesome/fontawesome-free`

```
@import '~@fontawesome/fontawesome-free';
```

```
import "@fontawesome/fontawesome-free/js/all";
```

Стало моднее, но медленнее

```
Processing by HomeController#index as HTML
```

```
  Rendering home/index.html within layouts/application
```

```
  Rendered home/index.html within layouts/application (Duration: 9.5ms | Allocations: 280)
```

```
[Webpacker] Compiling...
```

Foreman!

```
gem install foreman
```

Интеграция с Telegram

1. Ищем наиболее подходящий гем
2. Пишем в Gemfile

```
gem 'telegram-bot-ruby'
```

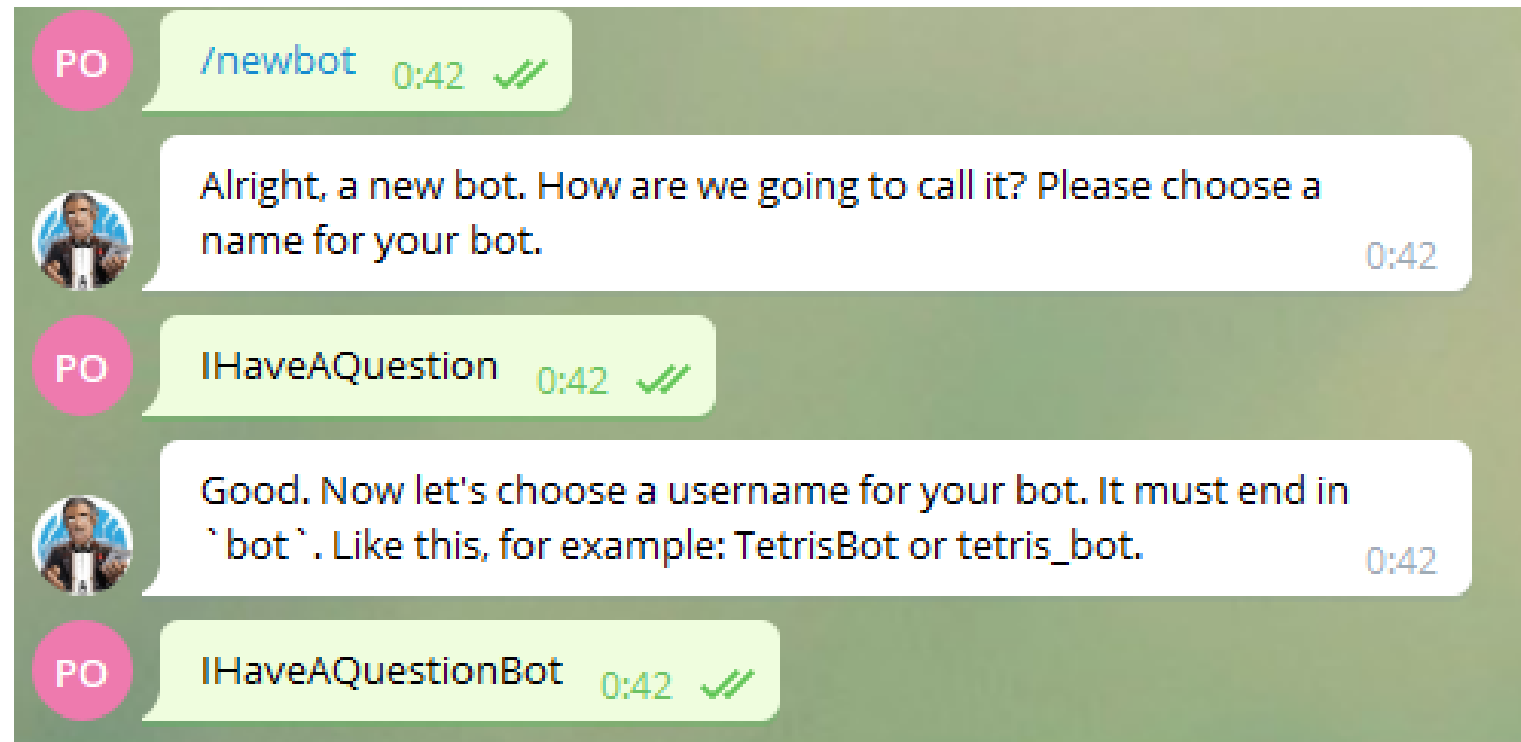
3. Выполняем **bundle install**

Приложению нужен один бот!

- Паттерн проектирования – singleton

```
require 'singleton'
require 'telegram/bot'
class TelegramService
  include Singleton
end
def initialize(token)
end
def send_message(message)
end
end
```

Регистрация бота



Использование проху

```
Telegram::Bot::Client.run(token,  
    url: 'https://proxy.example.com') do |bot|  
  # ...  
end
```

Запускаем бота

```
def start
  Thread.new do
    Telegram::Bot::Client.run(@token) do |bot|
      bot.listen do |message|
        case message.text
        when '/start'
          bot.api.send_message
            (chat_id: message.chat.id, text: "Hello, #{message.from.first_name}")
        when '/stop'
          bot.api.send_message
            (chat_id: message.chat.id, text: "Bye, #{message.from.first_name}")
        end
      end
    end
  end
end
```

Обрабатываем ошибку

```
def start
  Telegram::Bot::Client.run(@token) do |bot|
    #some code
  end
rescue StandardError => e
  Rails.logger.error e.message
end
```

Rails.logger

- **Debug**
- **Info**
- **Warn**
- **Error**
- **Fatal**
- **Unknown**

```
if ENV["RAILS_LOG_TO_STDOUT"].present?  
  Logger                = ActiveSupport::Logger.new(STDOUT)  
  Logger.formatter      = config.log_formatter  
  config.logger         = ActiveSupport::TaggedLogging.new(Logger)  
end
```

СПИСОК ПОДПИСЧИКОВ

```
class CreateSubscribes < ActiveRecord::Migration[6.0]
  def change
    create_table :subscribes do |t|
      t.string :chat_name
    end
  end
end
```

```
heroku run rake db:migrate
-a ihaveaquestion
```

Регистрируем подписчиков

```
bot.listen do |message|
  case message.text
  when '/subscribe'
    new_subscribe(message.chat.id)
  when '/unsubscribe'
    unsubscribe(message.chat.id)
  end
end
```

```
def new_subscribe(id)
  text = if Subscribe.find_by_chat_name(id).present?
    'Already Subscribed'
  else
    Subscribe.create(chat_name: id).valid? ? 'Subscribed!' : 'Can\'t Subscribed!'
  end
  @bot.api.send_message(chat_id: id, text: text)
end
```


Делаем рассылку

```
def create
  question_params = params[:question]
  question = Question.new(title: question_params[:title],
                          body: question_params[:body])

  question.user = @current_user
  question.save
  if question.valid?
    #в коде тут
    redirect_to question
  else
    flash[:alert] = 'Не удалось создать вопрос.'
    redirect_to new_question_path
  end
end
```

Active Job

- Нужно выполнять задачи асинхронно!

```
$ rails generate job notify_subscribers
```

```
class NotifySubscribersJob < ApplicationJob  
  queue_as :default  
  
  def perform(*args)  
  
  end  
end
```

Пишем task

```
def perform(*args)
  text = args[0]
  Subscribe.all do |subscribe|
    TelegramService.instance
      .send_message(text, subscribe.chat_name)
  end
end
```

ЧТО-ТО ПОШЛО НЕ ТАК

```
class RenameSubscribeToSubscription  
    < ActiveRecord::Migration[6.0]  
  def change  
    rename_table :subscribes, :subscriptions  
  end  
end
```

Как можно выполнить task?

```
NotifySubscribersJob.perform_later message
```

```
NotifySubscribersJob
```

```
  .set(wait_until: Date.tomorrow.noon)  
  .perform_later(message)
```

```
NotifySubscribersJob
```

```
  .set(wait: 1.week)  
  .perform_later(message)
```

УВЕДОМЛЯЕМ!

```
if question.valid?  
  NotifySubscribersJob.perform_later  
    "New question: #{question.title}"  
  redirect_to question  
else  
  ...
```

ССЫЛКИ

- <https://blog.logrocket.com/history-of-frontend-frameworks/>
- <https://api.jquery.com/category/selectors/>
- <https://medium.com/@guilhermepejon/how-to-install-bootstrap-4-3-in-a-rails-6-app-using-webpack-9eae7a6e2832>
- <https://stackify.com/rails-logger-and-rails-logging-best-practices/>
- <https://ru.bem.info/>