
GRUNT


Gulp


Brunch

Зачем нужны подобные инструменты?

- Они выполняют важные задачи по автоматизации
- Позволяют автоматизировать процесс разработки
- Улучшают качество конечного продукта

Grunt

- Автоматизирует повседневные задачи, для настройки использует конфигурационные файлы, в виде словарей
- Очень гибок и достаточно сложен в настройке
- Имеет самое большое количество плагинов

Gulp

- Использует идеологию "потока" при работе, вся конфигурация представлена в виде js-кода
- Считается одним из самых быстрых в работе
- Имеет большое количество плагинов

Brunch

- Менее популярный проект, главная идея которого - простота в настройке и скорость "начала" работы
- Встроенный сборщик в Phoenix Framework (Elixir)
- Имеет меньше всех плагинов (но достаточно для работы)

Webpack

- Самый модный на данный момент сборщик
- Позволяет включать в js сразу все: стили, шаблоны, json-файлы
- https://habrahabr.ru/post/245991/

Какие задачи выполняют чаще всего?

- Оптимизация изображений
- Объединение и сжатие стилей
- Объединение и сжатие скриптов
- Пост-проверка кода, запуск тестов

Соединение стилей

- Просто объединяет несколько файлов в один
- Убирает пробелы и комментарии

Соединение JS

- Существует несколько конкурирующих подходов: AMD, CommonJS, другие
- Существует много систем сборки, реализующие данные методы: RequireJS, browserify, другие
- Их нужно подключить к задачам

Практика

Контроль версий

- Версии позволяют управлять кешем
- Позволяет понимать, с чем мы работаем

Плагины

autoprefixer

- Анализирует код, добавляет необходимые префиксы к css, убирает лишние

- Является часть технологии PostCSS

css-lint, js-lint

- Анализируют код, выводят информацию о стилистических и других ошибках

Modernizr

- Анализирует код, оставляет в продуктивной версии только необходимые тесты

Livereload

- Заменяет код без перезагрузки страницы

Практика