

# Webpack



# Webpack

Webpack — это инструмент, позволяющий скомпилировать, например, JavaScript модули в единый JS-файл. Webpack также известен как сборщик модулей.

При большом количестве файлов он создает один объемный файл (или несколько файлов) для запуска вашего приложения.

# Webpack

Он также способен выполнять множество иных операций:

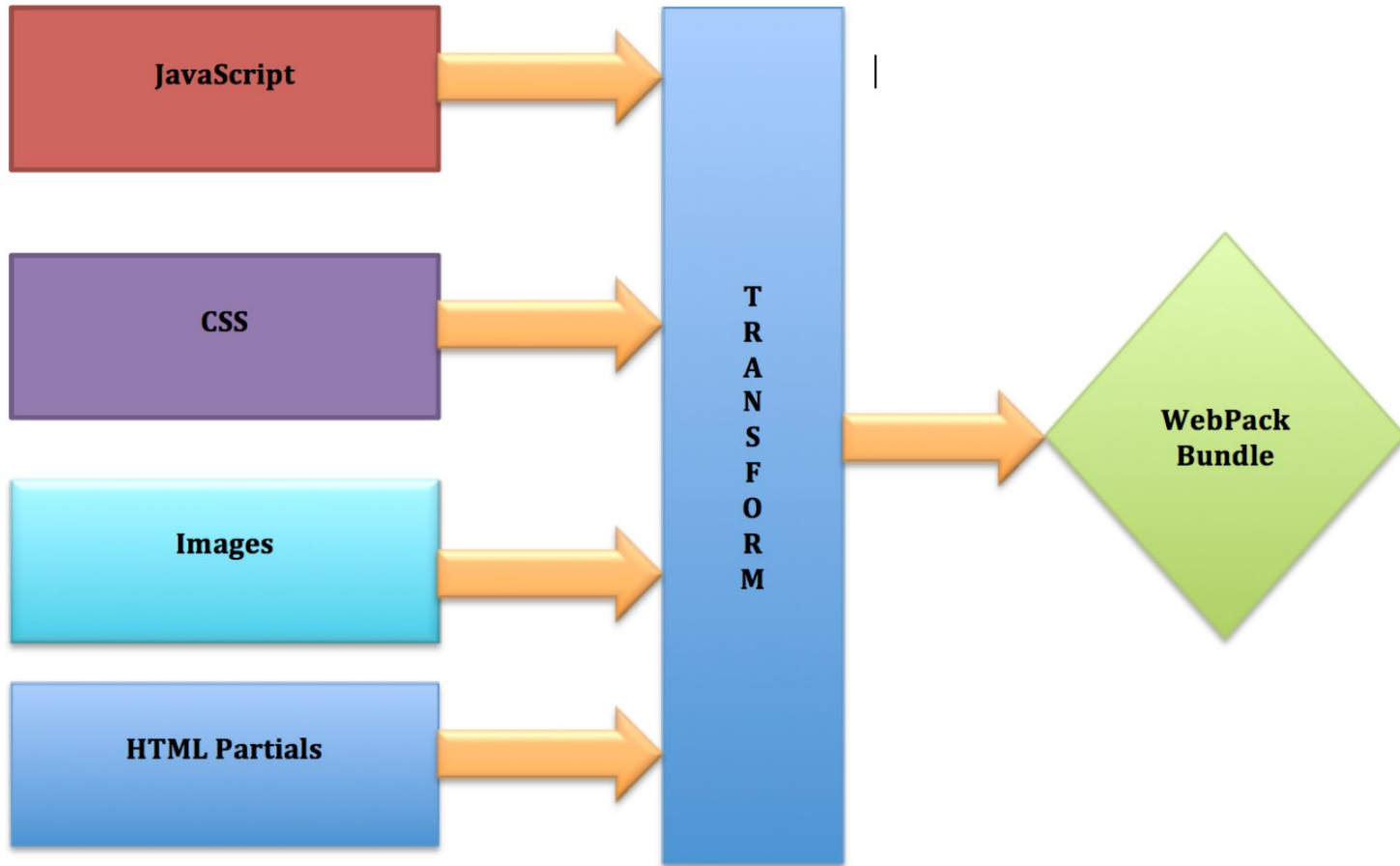
- помогает собрать воедино ваши ресурсы
- следит за изменениями и повторно выполняет задачи
- может выполнить транспиляцию JavaScript следующего поколения до более старого стандарта JavaScript (ES5) с помощью Babel, что позволит использовать новейшие функции JavaScript, не беспокоясь о том, поддерживает их браузер или нет

# Webpack

- может конвертировать встроенные изображения в data:URI
- позволяет использовать require() для CSS файлов
- может запустить webpack-dev-server (в нём встроен локальный сервер и livereload (“живая перезагрузка браузера”))
- может работать с Hot Module Replacement (замена горячего модуля)
- может разделить выходной файл (output file) на несколько файлов, чтобы избежать медленной загрузки страницы из-за большого размера JS-файла

# Webpack

Вам достаточно указать точку входа в ваше приложение (это может быть даже HTML-файл с тегам `<script>`), а webpack проанализирует файлы и объединит их в один выходной JavaScript-файл, содержащий все необходимое для запуска приложения.



# Webpack

Вам достаточно указать точку входа в ваше приложение (это может быть даже HTML-файл с тегами `<script>`), а webpack проанализирует файлы и объединит их в один выходной JavaScript-файл, содержащий все необходимое для запуска приложения.

# Установка Webpack

```
npm install -g webpack
```

Сборка проекта:

```
npm init
```



# Webpack

По умолчанию, точкой входа является ./src/index.js. Нижеприведенный пример использует файл ./index.js в качестве входной точки.

```
module.exports = {  
  /*...*/  
  entry: './index.js'  
  /*...*/  
}
```

# Webpack

По умолчанию, вывод размещается в `./dist/main.js`. В нижеприведенном примере, результат работы в Webpack генерируется в файле `app.js`:

```
module.exports = {  
  /*...*/  
  output: {  
    path: path.resolve(__dirname, 'dist'),  
    filename: 'app.js'  
  }  
  /*...*/  
}
```

# Webpack

Чтобы импортировать файлы стилей:

```
import 'style.css'
```

```
module.exports = {  
  /*...*/  
  module: {  
    rules: [  
      { test: /\.css$/, use: 'css-loader' },  
    ]  
  }  
  /*...*/  
}
```

# Webpack

Babel — он  
используется для  
транспилиации  
современного  
JavaScript в ES5

```
module.exports = {
  /* ... */
  module: {
    rules: [
      {
        test: /\.js$/,
        exclude: /(node_modules|bower_components)/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env']
          }
        }
      }
    ]
  }
  /* ... */
}
```

# Webpack

Горячая перезагрузка

Для горячей перезагрузки (hot reloading) нужен webpack-dev-server.

```
$ npm install -g webpack-dev-server
```

# Webpack

Плагины — это почти то же самое, что и загрузчики, они могут сделать то, что не могут загрузчики. Ко всему прочему, Webpack построен на системе плагинов, которые вы используете в своем файле конфигурации.

# Webpack

Плагин HTMLWebpackPlugin автоматически создает HTML-файл с уже подключенным скриптом.

```
module.exports = {  
  /*...*/  
  plugins: [  
    new HTMLWebpackPlugin()  
  ]  
  /*...*/  
}
```

# Webpack

CleanWebpackPlugin - перед регенерацией файлов, чтобы очистить нашу папку dist/ и получить файл с конфигурацией.

```
module.exports = {  
  /*...*/  
  plugins: [  
    new  
    CleanWebpackPlugin(['dist']),  
  ]  
  /*...*/  
}
```



# Webpack плагины

<https://webpack.js.org/plugins/>

# Webpack

Режимы вебпака: режим может быть настроен на development или production (по умолчанию стоит production).

```
module.exports = {  
  entry: './index.js',  
  mode: 'development',  
  output: {  
    path: path.resolve(__dirname,  
      'dist'),  
    filename: 'app.js'  
  }  
}
```

# Webpack

Режим development:

- менее оптимизирован, чем production
- работает быстрее
- не удаляет комментарии
- предоставляет более подробные сообщения об ошибках и способы их решения
- сильно облегчает отладку

# Обработка изображений

```
module.exports = {  
  /*...*/  
  module: {  
    rules: [  
      {  
        test: /\.png|svg|jpg|gif$/,  
        use: [  
          'file-loader'  
        ]  
      }  
    ]  
  }  
  /*...*/  
}
```

Позволяет импортировать  
изображения в ваш  
JavaScript