



html academy

интерактивные
онлайн-курсы

0x02 раздел: Модули

Разбор домашки

- **спес.md** из Угадай Мелодию



Разбор домашки

- **spec.md** из Угадай Мелодию
- **npm start** странно себя ведет (цикл)



Разбор домашки

- **spec.md** из Угадай Мелодию
- **npm start** странно себя ведет (цикл)
- страшное исключение на **npm test**



Сегодня в программе

- Деструктуризация



Сегодня в программе

- Деструктуризация
- Шаблонные строки, **rest** и **spread** операторы



Сегодня в программе

- Деструктуризация
- Шаблонные строки, **rest** и **spread** операторы
- Модули



Сегодня в программе

- Деструктуризация
- Шаблонные строки, **rest** и **spread** операторы
- Модули
- Модули ECMAScript2015



Сегодня в программе

- Деструктуризация
- Шаблонные строки, **rest** и **spread** операторы
- Модули
- Модули ECMAScript2015
- Инфраструктура



Сегодня в программе

- Деструктуризация
- Шаблонные строки, **rest** и **spread** операторы
- Модули
- Модули ECMAScript2015
- Инфраструктура
- Сборка проекта



Деструктуризация

интерактивное демо



Деструктуризация

// ES2015

```
const list = [1, 2, 3];  
let [ first, ,last ] = list;  
[ last, first ] = [ first, last ];  
  
console.info(first, last); // 3, 1
```

// ES5

```
var list = [ 1, 2, 3 ];  
var first = list[0], last = list[2];  
var tmp = first; first = last; last = tmp;
```



Деструктуризация

```
// ES2015
const cat = {
  name: 'Keks',
  male: true,
  age: 7,
  weight: 10
};
const { name, age } = cat;
console.info(name, age); // "Keks", 7
```

```
// ES5
var cat = {
  name: 'Keks',
  male: true,
  age: 7,
  weight: 10
};

var name = cat.name;
var age = cat.age;
console.info(name, age); // "Keks", 7
```



rest и *spread* операторы

Интерактивное демо



Прочие параметры (rest)

```
const getFullName = (name, surname, ...titles) => {  
  const space = titles.length > 0 ? ' ' : '';  
  return `${titles.join(' и ')}${space}${name} ${surname}`;  
};  
  
console.info(`Уважаемый ${getFullName('Кот', 'Матроскин')}!`); // Уважаемый Кот Матроскин!  
console.info(`Уважаемый ${getFullName('Кот', 'Матроскин', 'Единственный')}!`);  
// Уважаемый Единственный Кот Матроскин!  
console.info(`Уважаемый ${getFullName('Кот', 'Матроскин',  
  'Единственный', 'Неповторимый')}!`); // Уважаемый Единственный и Повторимый Кот  
Матроскин!
```



Разыменование в список параметров (*spread*)

```
const getFullName = (name, surname, ...titles) => {  
  const space = titles.length > 0 ? ' ' : '';  
  return `${titles.join(' и ')}${space}${name} ${surname}`;  
};
```

```
const president = ['Владимир', 'Путин', 'Президент'];  
console.info(`Уважаемый ${getFullName(...president)}!`);  
// Уважаемый Президент Владимир Путин!  
console.info(`Уважаемый ${getFullName(...president.slice(0, 2))}!`);  
// Уважаемый Владимир Путин!
```



Шаблонные строки и функции

Интерактивное демо



Функции шаблонизации

```
const anonymize = function (strings, ...values) {  
  console.log(strings[0]); // "Пока, "  
  console.log(strings[1]); // "!"  
  
  console.log(values[0]); // "Михаил"  
  
  return `${strings[0]}Неопознанный Енот${strings[1]}`;  
};  
  
const name = 'Михаил';  
  
console.info(anonymize`Привет, ${name}!`); // "Привет, Неопознанный Енот!"  
console.info(anonymize`Пока, ${name}!`); // "Пока, Неопознанный Енот!"  
  
console.info(String.raw`foo\n${ 42 }ba\tr`) // "foo\n42ba\tr"
```



To be continued...

- Новые объявления полей, методов объектов
- Symbol
- Итератор
- Генераторы
- Новые методы в старых объектах
- Метапрограммирование
- Интернационализация и локализация
- Типизированные массивы
- Словари и множества



Модули



Модули и сборщики модулей



Модули и сборщики модулей

- Модуль



Модули и сборщики модулей

- Модуль
 - изолирует пространство имен



Модули и сборщики модулей

- Модуль
 - изолирует пространство имен
 - описывает зависимости



Модули и сборщики модулей

- Модуль
 - изолирует пространство имен
 - описывает зависимости
 - описывает то, что он предоставляет наружу



Модули и сборщики модулей

- Модуль
 - изолирует пространство имен
 - описывает зависимости
 - описывает то, что он предоставляет наружу
- Сборщик модулей



Модули и сборщики модулей

- Модуль
 - изолирует пространство имен
 - описывает зависимости
 - описывает то, что он предоставляет наружу
- Сборщик модулей
 - выстраивает порядок подключения



Модули и сборщики модулей

- Модуль
 - изолирует пространство имен
 - описывает зависимости
 - описывает то, что он предоставляет наружу
- Сборщик модулей
 - выстраивает порядок подключения
 - загружает, выполняет, хранит файлы-модулей



Модули и сборщики модулей

- Модуль
 - изолирует пространство имен
 - описывает зависимости
 - описывает то, что он предоставляет наружу
- Сборщик модулей
 - выстраивает порядок подключения
 - загружает, выполняет, хранит файлы-модулей
 - собирает, минифицирует и запаковывает



IIFE (Immediately Invoked Function Expression)

```
// Файл mother.js
'use strict';
(function () {
  window.mother = {
    name: `Мама`,
    age: 12,
    male: false
  }
})();
```

```
// Файл father.js
'use strict';
(function () {
  window.father = {
    name: `Папа`,
    age: 12,
    male: true
  }
})();
```

```
// Файл family.js
(function () {
  const family = {
    father: window.father,
    mother: window.mother
  };

```

```
  const makeChild = () => `Отец: ${family.father.name}, мать: $
{family.mother.name}`;
```

```
  window.child = makeChild();
})();
```

benalman.com/news/2010/11/immediately-invoked-function-expression



IIFE сборка

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Имя сайта</title>  
</head>  
<body>  
  
  <script src="mother.js"></script>  
  <script src="father.js"></script>  
  <script src="family.js"></script>  
  
</body>  
</html>
```



CommonJS

```
// Файл mother.js
module.exports = {
  name: `Мама`,
  age: 12,
  male: false
};
```

```
// Файл father.js
module.exports = {
  name: `Папа`,
  age: 12,
  male: true
};
```

```
// Файл family.js
const fatherName = require(`./father`).name;
const motherName = require(`./mother`).name;

const makeChild = () => `Отец: ${fatherName}, мать: ${motherName}`;

module.exports = makeChild();
```



CommonJS ES2015

```
// Файл mother.js
module.exports = {
  name: `Мама`,
  age: 12,
  male: false
};
```

```
// Файл father.js
module.exports = {
  name: `Папа`,
  age: 12,
  male: true
};
```

```
// Файл family.js
const {name: fatherName} = require(`./father`);
const {name: motherName} = require(`./mother`);

const makeChild = () => `Отец: ${fatherName}, мать: ${motherName}`;

module.exports = makeChild();
```



AMD

```
// Файл mother.js
define([], function () {
  return {
    name: `Мама`,
    age: 12,
    male: false
  };
});
```

```
// Файл father.js
define([], function () {
  return {
    name: `Папа`,
    age: 12,
    male: true
  };
});
```

```
// Файл family.js
define([`./father`, `./mother`] , function (father, mother) {
  const {name: fatherName} = father;
  const {name: motherName} = mother;

  const makeChild = () => `Отец: ${fatherName}, мать: ${motherName}`;

  return makeChild();
});
```



CommonJS/AMD сборка



CommonJS/AMD сборка

- Загрузчик модулей (module loader):



CommonJS/AMD сборка

- Загрузчик модулей (module loader):
 - загружает файлы



CommonJS/AMD сборка

- Загрузчик модулей (module loader):
 - загружает файлы
 - добавляет переменные *module*, *require*



CommonJS/AMD сборка

- Загрузчик модулей (module loader):
 - загружает файлы
 - добавляет переменные *module*, *require*
- Сборщик модулей (module bundler)



CommonJS/AMD сборка

- Загрузчик модулей (module loader):
 - загружает файлы
 - добавляет переменные *module*, *require*
- Сборщик модулей (module bundler)
 - собирает все модули в один файл



CommonJS/AMD сборка

- Загрузчик модулей (module loader):
 - загружает файлы
 - добавляет переменные *module*, *require*
- Сборщик модулей (module bundler)
 - собирает все модули в один файл
 - минифицирует



CommonJS/AMD сборка

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Имя сайта</title>

  <script data-main="scripts/main" src="lib/require.js"></script>
</head>
<body>

</body>
```

```
// Файл main.js
requirejs.config({
  baseUrl: 'lib',
  paths: {
    app: './'
  }
});

requirejs(['app/family']);
```



UMD (Universal Module Definition)

```
(function (root, factory) {  
  
    if (typeof define === 'function' && define.amd) {  
  
        // AMD. Register as an anonymous module.  
        define(['exports', './father', './mother'], factory);  
    } else if (typeof exports === 'object' && typeof exports.nodeName !==  
'string') {  
  
        // CommonJS  
        factory(exports, require('./father'), require('./mother'));  
    } else {  
  
        // Browser globals  
        factory((root.commonJsStrict = {}), root.father, root.mother);  
    }  
  
})(this, function (exports, father, mother) {  
    const {name: fatherName} = father;  
    const {name: motherName} = mother;  
  
    const makeChild = () => `Отец: ${fatherName}, мать: ${motherName}`;  
  
    exports.child = makeChild(); github.com/umdjs/umd  
}));
```



ECMAScript 2015 Modules

```
// Файл mother.js
export const name = `Мама`;
export const age = 12;
export const male = false;
```

```
// Файл father.js
export default {
  name: `Папа`,
  age: 12,
  male: true
};
```

```
// Файл family.js
import {name as motherName} from './mother';
import father from './father';

const makeChild = () => `Отец: ${father.name}, мать: ${motherName}`;

export default makeChild();
```



Import/Export

```
// lib/mathplusplus.js
export * from 'lib/math';

export const EPSILON = 2.220446049250313e-16;

export const sign = (x) => {
  return isNaN(x) ? NaN : x < 0 ? -1 : x > 0 ? 1 : 0;
};

// count.js
import {EPSILON, sign, abs} from 'lib/mathplusplus';

console.info(sign(EPSILON)); // 1
console.info(abs(-100));    // 100
```



Import по-умолчанию

```
// lib/mathplusplus.js
import * as math from 'lib/math';

export const PI2 = 2 * math.PI;

export * from 'lib/math';

export const EPSILON = 2.220446049250313e-16;

export default sign = (x) => {
  return isNaN(x) ? NaN : x < 0 ? -1 : x > 0 ? 1 : 0;
};

// count.js
import sign, {EPSILON, abs} from 'lib/mathplusplus';

console.info(sign(EPSILON)); // 1
console.info(abs(-100));    // 100
```



import запрещен во вложенных конструкциях

```
// lib/mathplusplus.js
const MyMath = require('lib/math');
let library;
if(MyMath.random() > 0.5) {
  library = require('library1');
} else {
  library = require('library2');
}

library = require('library3');
```



Сборщик модулей ES2015



Сборщик модулей ES2015

- Встроенный в браузер



Сборщик модулей ES2015

- Встроенный в браузер
 - HTML спецификация
jakearchibald.com/2017/es-modules-in-browsers



Сборщик модулей ES2015

- Встроенный в браузер
 - HTML спецификация
jakearchibald.com/2017/es-modules-in-browsers
- Статические утилиты



Сборщик модулей ES2015

- Встроенный в браузер
 - HTML спецификация
jakearchibald.com/2017/es-modules-in-browsers
- Статические утилиты
 - Webpack
webpack.js.org



Сборщик модулей ES2015

- Встроенный в браузер
 - HTML спецификация
jakearchibald.com/2017/es-modules-in-browsers
- Статические утилиты
 - Webpack
webpack.js.org
 - rollup.js
rollupjs.org



Сборщик модулей ES2015

- Встроенный в браузер
 - HTML спецификация
jakearchibald.com/2017/es-modules-in-browsers
- Статические утилиты
 - Webpack
webpack.js.org
 - rollup.js
rollupjs.org
 - прочие загрузчики модулей



Виды модулей JavaScript

- IIFE
- CommonJS
www.commonjs.org
- AMD
github.com/amdjs/amdjs-api/wiki/AMD
- UMD
github.com/umdjs/umd
- ECMAScript 2015 Modules
developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import



rollup.js

Сборщик модулей — утилита, которая на вход получает модули в любом стиле (CommonJS, AMD, ES2015...) и на выход отдает получившийся файл в виде модуля или самостоятельного скрипта



source map («карта исходников»)

Способ задать соответствие между исходными файлами и преобразованными для отладки или просмотра кода



Демо



Соберем приложение (demo)



Node.js

Программа позволяющая запускать JS-файлы и программы в окружении Операционной Системы



NPM (Node Package Manager)

Программа для управления пакетами и зависимостями
для программ разрабатываемых на базе **Node.js**



package.json

```
{
  "name": "420-superquest",
  "version": "1.0.0",
  "description": "Super Quest",
  "scripts": {
    "build": "gulp build",
    "eslint": "eslint js/",
    "start": "gulp serve",
    "test": "npm run eslint && gulp test"
  },
  "devDependencies": {
    "autoprefixer": "6.7.7",
    "browser-sync": "2.18.8",
    "eslint": "3.19.0",
    "eslint-config-htmlacademy": "0.2.x",
    "gulp": "3.9.1"
  }
}
```

docs.npmjs.com/files/package.json



Работа с NPM

- *npm install* — устанавливает все зависимости указанные в *package.json*



Работа с NPM

- *npm install* — устанавливает все зависимости указанные в *package.json*
- *npm run <script-name>* — запускает скрип с именем *script-name*



Работа с NPM

- *npm install* — устанавливает все зависимости указанные в *package.json*
- *npm run <script-name>* — запускает скрип с именем *script-name*
- *npm test* — тоже самое что *npm run test*



Работа с NPM

- *npm install* — устанавливает все зависимости указанные в *package.json*
- *npm run <script-name>* — запускает скрип с именем *script-name*
- *npm test* — тоже самое что *npm run test*
- *npm install <dependency-name> --save-dev --exact* — устанавливает зависимость *dependency-name* и сохраняет её в *devDependencies* с фиксированной версией



Работа с NPM

- *npm install* — устанавливает все зависимости указанные в *package.json*
- *npm run <script-name>* — запускает скрип с именем *script-name*
- *npm test* — тоже самое что *npm run test*
- *npm install <dependency-name> --save-dev --exact* — устанавливает зависимость *dependency-name* и сохраняет её в *devDependencies* с фиксированной версией
- *npm i <dependency-name> -DE === npm install <dependency-name> --save-dev --exact*



Работа с NPM

- *npm install* — устанавливает все зависимости указанные в *package.json*
- *npm run <script-name>* — запускает скрип с именем *script-name*
- *npm test* — тоже самое что *npm run test*
- *npm install <dependency-name> --save-dev --exact* — устанавливает зависимость *dependency-name* и сохраняет её в *devDependencies* с фиксированной версией
- *npm i <dependency-name> -DE === npm install <dependency-name> --save-dev --exact*
- *npm i <dependency-name> --global* — устанавливает пакет глобально в систему



js/main.js ECMAScript 5

```
var mainElement = document.getElementById('main');

mainElement.querySelector('.start__button').onclick = function (e) {
    e.preventDefault();

    var greeting = document.createElement('p');
    greeting.innerHTML = 'Время приключений!\n' +
        'Поехали!';
    mainElement.appendChild(greeting)
};
```



Проверим работу

Проверим правильность кода при помощи ESLint

npm test

```
zsh
htmlacademy ~/superquest module1-task2 npm test
~/superquest/js/main.js
 1:1  error  Unexpected var, use let or const instead  no-var
 1:43 error  Strings must use backtick                quotes
 2:27 error  Strings must use backtick                quotes
 4:3   error  Unexpected var, use let or const instead  no-var
 4:41 error  Strings must use backtick                quotes
 5:24 error  Strings must use backtick                quotes
 5:49 error  Strings must use backtick                quotes
 6:36 error  Missing semicolon                       semi
 7:3   error  Irregular whitespace not allowed         no-irregular-whitespace
```

✖ 9 problems (9 errors, 0 warnings)

```
htmlacademy ~/superquest module1-task2
```



ESLint

parserOptions:

sourceType: 'module'

env:

es6: true

browser: true

extends: 'htmlacademy/es6'



js/main.js ECMAScript 2015

```
const mainElement = document.getElementById(`main`);

mainElement.querySelector(`.start__button`).onclick = (evt) => {
  evt.preventDefault();

  const greeting = document.createElement(`p`);
  greeting.innerHTML = `Время приключений!\n` + `Поехали!`;
  mainElement.appendChild(greeting);
};
```



Соберем наше приложение

Сборка проекта производится при помощи Gulp.js

npm start

```
htmlacademy ~/pixel-hunter module1-task2 npm start
> 420-superquest@1.0.0 start ~/superquest
> gulp serve
```

```
[17:23:32] Using gulpfile ~/superquest/gulpfile.js
[17:23:32] Starting 'clean'...
[17:23:32] Finished 'clean' after 17 ms
[17:23:32] Starting 'assemble'...
[17:23:32] Starting 'copy-html'...
[17:23:32] Starting 'scripts'...
[17:23:32] Starting 'style'...
[17:23:32] Finished 'style' after 7.96 ms
[17:23:32] Finished 'assemble' after 35 ms
[17:23:32] Starting 'serve'...
[17:23:32] Finished 'serve' after 37 ms
[BS] Access URLs:
```

```
-----
Local: http://localhost:3502
External: http://192.168.20.128:3502
```



Gulpfile.js

```
gulp.task('style', function () { ... });

gulp.task('imagemin', ['copy'], function () { ... });

gulp.task('copy-html', function () { ... });

gulp.task('copy', ['copy-html', 'scripts', 'style'], function () { ... });

gulp.task('scripts', function () { ... });

gulp.task('clean', function () { ... });

gulp.task('serve', ['assemble'], function () {
  server.init({
    server: './build',
    open: true,
    port: 3501
  });
  ...
});

gulp.task('assemble', ['clean'], function () {
  gulp.start('copy', 'style');
});
```


gulpjs.com



Добавим сборку модулей в проект

Установим плагин для Gulp – gulp-better-rollup

npm install --save-dev --save-exact gulp-better-rollup



zsh

```
htmlacademy ~/superquest module1-task2 npm i gulp-better-rollup -DE
420-superquest@1.0.0 ~/superquest
└─ gulp-better-rollup@1.1.1

npm WARN 420-superquest@1.0.0 No repository field.
npm WARN 420-superquest@1.0.0 No license field.

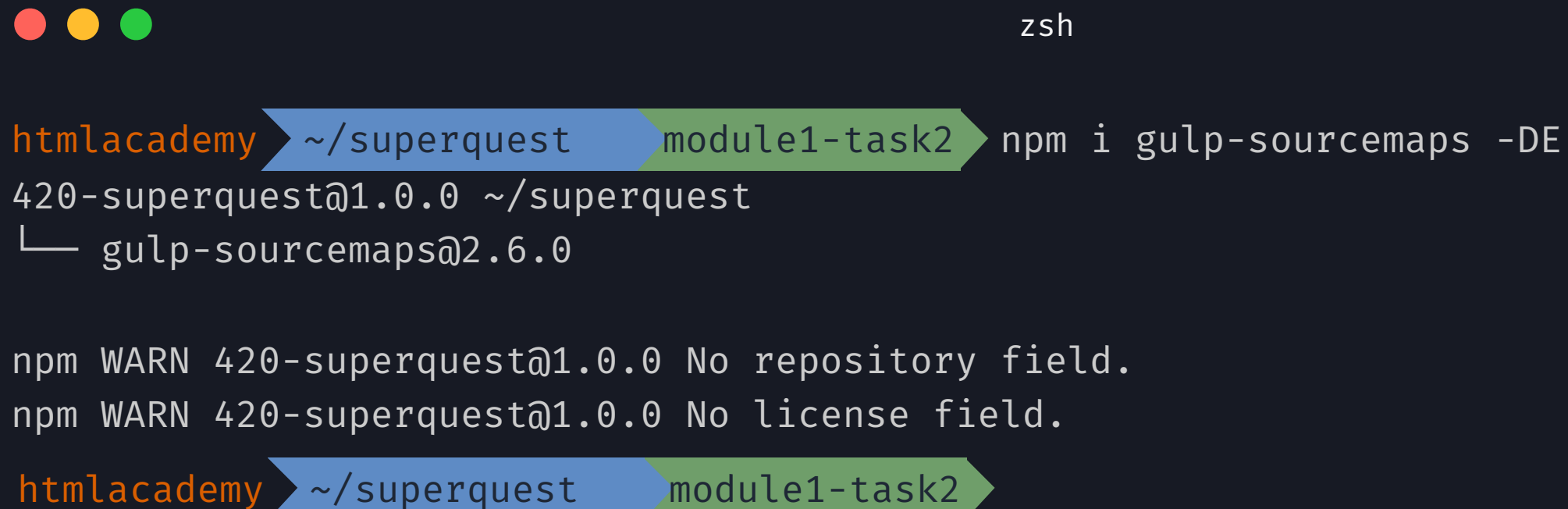
htmlacademy ~/superquest module1-task2
```



Добавим плагин для сборки source map

Установим плагин для Gulp – gulp-sourcemaps

npm install --save-dev --save-exact gulp-sourcemaps



```
htmlacademy ~/superquest module1-task2 npm i gulp-sourcemaps -DE
420-superquest@1.0.0 ~/superquest
└─ gulp-sourcemaps@2.6.0

npm WARN 420-superquest@1.0.0 No repository field.
npm WARN 420-superquest@1.0.0 No license field.

htmlacademy ~/superquest module1-task2
```



Добавим rollup.js в сборку

```
gulp.task('scripts', function () {  
  return gulp.src('js/**/*.js')  
    .pipe(plumber())  
    .pipe(gulp.dest('build/js/'));  
});
```



Добавим rollup.js в сборку

```
const rollup = require('gulp-better-rollup');

gulp.task('scripts', function () {
  return gulp.src('js/main.js')
    .pipe(plumber())
    // use IIFE format
    .pipe(rollup({}, 'iife'))
    .pipe(gulp.dest('build/js'));
});
```



Добавим sourcemaps в сборку

```
const rollup = require('gulp-better-rollup');
const sourcemaps = require('gulp-sourcemaps');

gulp.task('scripts', function () {
  return gulp.src('js/main.js')
    .pipe(plumber())
    .pipe(sourcemaps.init())
    // use IIFE format
    .pipe(rollup({}, 'iife'))
    // save sourcemap as separate file (in the same folder)
    .pipe(sourcemaps.write(''))
    .pipe(gulp.dest('build/js'));
});
```



Запустим приложение

Сборка проекта производится при помощи Gulp.js

npm start

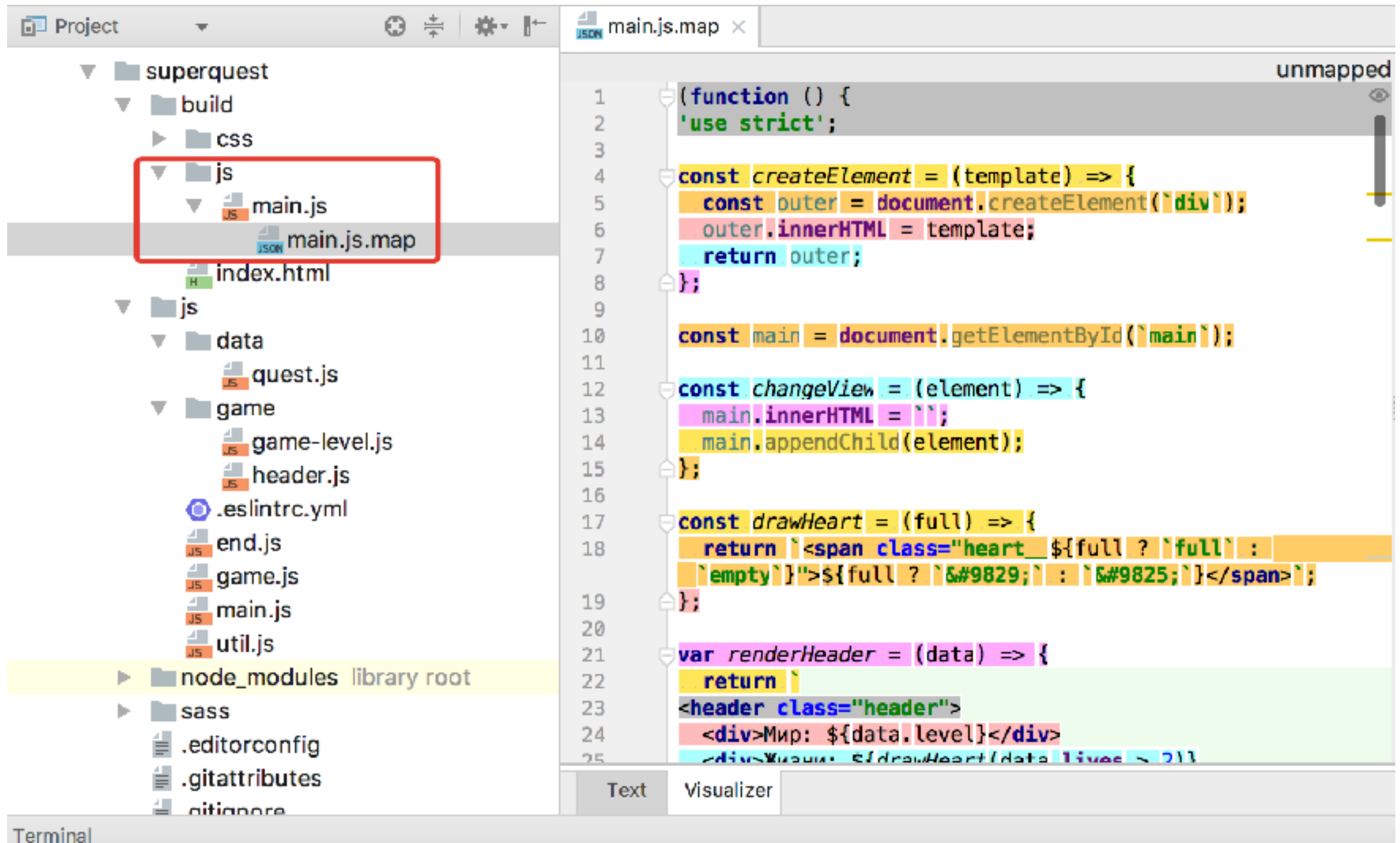
```
htmlacademy ~/pixel-hunter module1-task2 npm start
> 420-superquest@1.0.0 start ~/superquest
> gulp serve
```

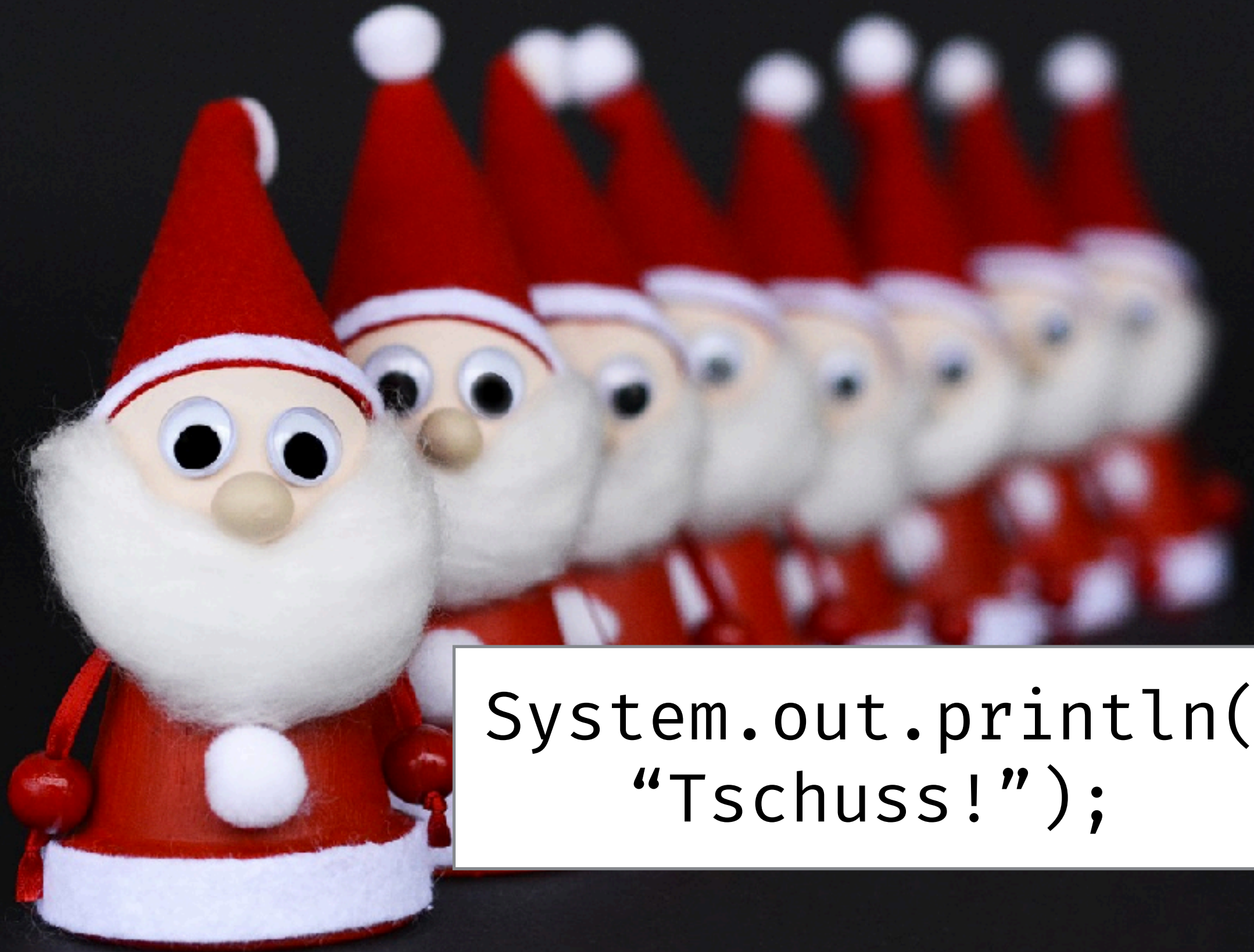
```
[17:23:32] Using gulpfile ~/superquest/gulpfile.js
[17:23:32] Starting 'clean'...
[17:23:32] Finished 'clean' after 17 ms
[17:23:32] Starting 'assemble'...
[17:23:32] Starting 'copy-html'...
[17:23:32] Starting 'scripts'...
[17:23:32] Starting 'style'...
[17:23:32] Finished 'style' after 7.96 ms
[17:23:32] Finished 'assemble' after 35 ms
[17:23:32] Starting 'serve'...
[17:23:32] Finished 'serve' after 37 ms
[BS] Access URLs:
```

```
-----
Local: http://localhost:3502
External: http://192.168.20.128:3502
```



Sourcemap (карта «исходников»)





```
System.out.println(  
    "Tschuss!");
```

