Вт, 12 августа 2025

ДЗ к уроку №4

**Дополнительное задание**

Задание 1

Изучить возможности плагинов babel и расширить возможности таска с урока путем установки новых плагинов.

Код:

npm install --save-dev @babel/plugin-proposal-class-properties @babel/plugin-proposal-optional-chaining

{

  "presets": ["@babel/preset-env"],

  "plugins": [

    "@babel/plugin-proposal-class-properties",

    "@babel/plugin-proposal-optional-chaining"

  ]

}

class User {

  name = 'Gulp User';

  printName() {

    console.log(this.name?.toUpperCase());

  }

}

const user = new User();

user.printName();

Результат:

D:\ITVDN\Спеціальності\Frontend Developer\Курси\Верстка сторінок з використанням Gulp\003>npx gulp js

[22:04:59] Using gulpfile D:\ITVDN\Спеціальності\Frontend Developer\Курси\Верстка сторінок з використанням Gulp\003\gulpfile.js

[22:04:59] Starting 'js'...

[22:05:00] Finished 'js' after 870 ms

"use strict";

function \_typeof(o) { "@babel/helpers - typeof"; return \_typeof = "function" == typeof Symbol && "symbol" == typeof Symbol.iterator ? function (o) { return typeof o; } : function (o) { return o && "function" == typeof Symbol && o.constructor === Symbol && o !== Symbol.prototype ? "symbol" : typeof o; }, \_typeof(o); }

function \_classCallCheck(a, n) { if (!(a instanceof n)) throw new TypeError("Cannot call a class as a function"); }

function \_defineProperties(e, r) { for (var t = 0; t < r.length; t++) { var o = r[t]; o.enumerable = o.enumerable || !1, o.configurable = !0, "value" in o && (o.writable = !0), Object.defineProperty(e, \_toPropertyKey(o.key), o); } }

function \_createClass(e, r, t) { return r && \_defineProperties(e.prototype, r), t && \_defineProperties(e, t), Object.defineProperty(e, "prototype", { writable: !1 }), e; }

function \_defineProperty(e, r, t) { return (r = \_toPropertyKey(r)) in e ? Object.defineProperty(e, r, { value: t, enumerable: !0, configurable: !0, writable: !0 }) : e[r] = t, e; }

function \_toPropertyKey(t) { var i = \_toPrimitive(t, "string"); return "symbol" == \_typeof(i) ? i : i + ""; }

function \_toPrimitive(t, r) { if ("object" != \_typeof(t) || !t) return t; var e = t[Symbol.toPrimitive]; if (void 0 !== e) { var i = e.call(t, r || "default"); if ("object" != \_typeof(i)) return i; throw new TypeError("@@toPrimitive must return a primitive value."); } return ("string" === r ? String : Number)(t); }

var User = /\*#\_\_PURE\_\_\*/function () {

  function User() {

    \_classCallCheck(this, User);

    \_defineProperty(this, "name", 'Gulp User');

  }

  return \_createClass(User, [{

    key: "printName",

    value: function printName() {

      var \_this$name;

      console.log((\_this$name = this.name) === null || \_this$name === void 0 ? void 0 : \_this$name.toUpperCase());

    }

  }]);

}();

var user = new User();

user.printName();

**Самостоятельная деятельность учащегося**

Задание 1

Создайте репозиторий на github.

Результат:

Задание 2

Создайте файл js c современным синтаксисом и создайте таск для их обработки с помощью babel.

Код:

const greet = (name) => {

  console.log(`Привет, ${name}!`);

};

greet('Gulp + Babel');

npm install --save-dev @babel/core @babel/preset-env gulp-babel

{

  "presets": ["@babel/preset-env"]

}

const gulp = require('gulp');

const sass = require('gulp-sass')(require('sass'));

const babel = require('gulp-babel');

gulp.task('scss', function() {

  return gulp.src('src/styles/\*\*/\*.scss')

    .pipe(sass().on('error', sass.logError))

    .pipe(gulp.dest('build/styles'));

});

gulp.task('js', function() {

  return gulp.src('src/scripts/\*\*/\*.js')

    .pipe(babel())

    .pipe(gulp.dest('build/scripts'));

});

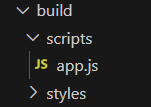
Результат:

D:\ITVDN\Спеціальності\Frontend Developer\Курси\Верстка сторінок з використанням Gulp\003>npx gulp js

[21:54:33] Using gulpfile D:\ITVDN\Спеціальності\Frontend Developer\Курси\Верстка сторінок з використанням Gulp\003\gulpfile.js

[21:54:33] Starting 'js'...

[21:54:34] Finished 'js' after 858 ms



Задание 3

Создайте таск watcher, который будет следить за изменениями в файлах вашего проекта и будет запускать таски с задания 1 и 2.

Код:

const gulp = require('gulp');

const sass = require('gulp-sass')(require('sass'));

const babel = require('gulp-babel');

gulp.task('scss', function() {

  return gulp.src('src/styles/\*\*/\*.scss')

    .pipe(sass().on('error', sass.logError))

    .pipe(gulp.dest('build/styles'));

});

gulp.task('js', function() {

  return gulp.src('src/scripts/\*\*/\*.js')

    .pipe(babel())

    .pipe(gulp.dest('build/scripts'));

});

gulp.task('watcher', function() {

  gulp.watch('src/styles/\*\*/\*.scss', gulp.series('scss'));

  gulp.watch('src/scripts/\*\*/\*.js', gulp.series('js'));

});

Внесем изменения в файлы main.scss, app.js в src/

app.js

const greet = (name) => {

  console.log(`Привет, ${name}!`);

};

greet('Gulp + Babel + SCSS');

main.scss

$primary-color: #849dff;

body {

  font-family: Arial, sans-serif;

  background-color: $primary-color;

  color: white;

}

Результат:

D:\ITVDN\Спеціальності\Frontend Developer\Курси\Верстка сторінок з використанням Gulp\003>npx gulp watcher

[21:57:42] Using gulpfile D:\ITVDN\Спеціальності\Frontend Developer\Курси\Верстка сторінок з використанням Gulp\003\gulpfile.js

[21:57:42] Starting 'watcher'...

[21:58:27] Starting 'js'...

[21:58:28] Finished 'js' after 997 ms

[21:58:43] Starting 'scss'...

[21:58:43] Finished 'scss' after 138 ms

Изменения в аналогичных файлах в build/

app.js

"use strict";

var greet = function greet(name) {

  console.log("\u041F\u0440\u0438\u0432\u0435\u0442, ".concat(name, "!"));

};

greet('Gulp + Babel + SCSS');

main.css

body {

  font-family: Arial, sans-serif;

  background-color: #849dff;

  color: white;

}