

Інструкція для виконання лабораторної роботи №1.

- **Node.js** – завантажуюмо Node.js з інтернету
(для тих у кого Windows 7, його просто не інстальюєш, потрібно буде перейти за посиланням <https://nodejs.org/download/release/v13.14.0/> і завантажити версію 13.14 (x64.msi або x32.msi)
- **myProject** – створюємо папку myProject, в якій буде міститися вся лабораторна робота №1
- **Консоль** – win + R (cmd)
 - Windows (7 – 8) – Shift + ПКМ (Відкрити в консолі)
 - Windows 10 - win + R (cmd)
 - cd folder_name — після команди cd вводимо відносний шлях до папки проекту
- **npm i gulp -g** (встановлюємо gulp глобальний)
 - Linux + Mac (sudo npm i gulp -g)
- node -v, gulp -v, npm -v (перевірка версій встановлених програм, якщо не показує, то сталася помилка).
- Обов'язково включити відображення розширення (необхідно тільки для windows - панель управління -> параметри папок -> відображати розширення для зареєстрованих типів файлів)
- Якщо треба почистити консоль (у разі неправильного набору), то набираємо cls — для windows, clear — для linux + mac

Етапи створення проекту

1. Переходимо у папку **myProject**
2. **npm init** – відповідаємо на запитання
 - package-name: назва проекту (необхідно назвати відповідно до рекомендацій у завданні);
 - version: версія (можна залишити за замовчуванням 1.0.0);
 - description: короткий опис проекту (можна пропустити)
 - entry point: (index.js), test command:, keywords: ці поля залишаємо за замовчуванням;
 - git repository: (поки що нічого не пишемо)
 - author: ім'я автора проекту;
 - license: залишити за замовчуванням;
 - Is this ok? — потрібно ввести «yes» і натиснути Enter. (або enter)

3. **npm i gulp -D** – встановлення gulp локально

Далі можна перевірити версію gulp –v, або відкрити у папці myProject файл package.json і у розділі devDependencies буде вказано gulp і його версія

4. **встановлення необхідних модулів**

```
npm i -D gulp- concat
npm i -D gulp- sass
npm i -D gulp- autoprefixer
npm i -D gulp- cssnano
npm i -D gulp- rename
npm i -D gulp- uglify
npm i -D gulp- imagemin
npm i -D browsersync
```

5. **задаємо структуру папок**

- app/ - список файлів, ресурсів, картинок
 - sass – файли препроцесора sass
 - js – файли скриптів javascript
 - img – файли зображень
 - css – файли css стилів
 - fonts – файли шрифтів
 - index.html – головний html файл
- dist/ - готовий результат
- gulpfile.js – список задач для gulp 4 (файл)

6. написання задач для gulp 4:

писати у файлі gulpfile.js (відкривши у будь-якому блокноті).

```
function simpleTask() {
  console.log ('This is a test task!');
}
exports.default = simpleTask
```

(У консолі прописуємо: gulp simpleTask)

////////////////////////////////////

```
const { src, dest, watch, series } = require ("gulp");
const concat = require ("gulp-concat");
const sass = require ("gulp-sass");
const autoprefixer = require ("gulp-autoprefixer");
const cssnano = require ("gulp-cssnano");
const rename = require ("gulp-rename");
const uglify = require ("gulp-uglify");
const imagemin = require ("gulp-imagemin");
```

```
//копіювання HTML файлів в папку dist
function task_html () {
  return src ( "app/*.html")
    .pipe (dest ( "dist"));
}
exports.html = task_html
```

(У консолі прописуємо: gulp html)

//об'єднання, компіляція Sass в CSS, додавання префіксів і подальша мінімізація коду

```
function task_sass () {
  return src ( "app/sass/*.sass")
    .pipe (concat ( 'styles.sass'))
    .pipe (sass ())
    .pipe (autoprefixer ({
      browsers: [ 'last 2 versions'],
      cascade: false
    })))
    .pipe (cssnano ())
    .pipe (rename ({suffix: '.min'}))
    .pipe (dest ( "dist/css"));
}
exports.sass = task_sass
```

(У консолі прописуємо: gulp sass)

//об'єднання і стиснення JS-файлів

```
function task_scripts () {  
  return src ( "app/js/*.js" ) //вихідна директорія файлів  
    .pipe (concat ( 'scripts.js' )) // конкатенація js-файлів в один  
    .pipe (uglify ()) //стиснення коду  
    .pipe (rename ( {suffix: '.min'} )) //перейменування файлу з приставкою  
    .min  
    .pipe (dest ("dist/js")); // директорія продакшена  
}  
exports.scripts = task_scripts
```

(У консолі прописуємо: gulp scripts)

//стискання зображень

```
function task_imgs() {  
  return src ( "app/img/*.(jpg|jpeg|png|gif)" )  
    .pipe (imagemin ( {  
      progressive: true,  
      svgoPlugins: [{removeViewBox: false}],  
      interlaced: true  
    } ))  
    .pipe (dest ("dist/images"))  
}  
exports.imgs = task_imgs
```

(У консолі прописуємо: gulp imgs)

//Відстежування за змінами у файлах

```
function task_watch() {  
  watch ("app/*.html", task_html);  
  watch ("app/js/*.js", task_scripts);  
  watch ("app/sass/*.sass", task_sass);  
  watch ("app/images/*.(jpg|jpeg|png|gif)", task_imgs);  
}  
exports.watch = task_watch
```

(У консолі прописуємо: gulp watch)

//Запуск тасків за замовчуванням

```
exports.build = series(task_html, task_sass, task_scripts, task_imgs,  
task_watch).
```

Після кожної задачі зберігаємо файл!!!

Для перевірки команд вводимо *gulp –tasks*

Корисні посилання:

<https://www.youtube.com/watch?v=n-N1BnloIVE&t=2381s>

(Детальні відео-уроки по gulp 4)