**Gulp**

**Встановлення Gulp в проект**

Спершу скачати і встановити **Node JS**

1. **Глобально задати gulp** (npm install –global gulp-cli)

**Заходимо в папку проекту**

1. **Створення json файла** (npm init) після заповняємо анкетку і продовжуємо
2. **Загружаємо загальнообов’язкові плагіни:**
   1. npm install gulp --save – dev – **встановлення gulp з останньою версією на проект**
3. **Створюємо папку з налаштуваннями:** gulpfile.js

**!Необхідно, щоб плагіни встановлювались в devDependencies, а не в dependencies**

**Для перенесення готової збірки в проект:**

1. Скопіювати файли: gulpfile.js, package.json, scr в новий проект
2. В терміналі вбиваємо команду: npm i

**Встановлення додаткових плагінів:**

**HTML/CSS**

**Використані:**

1. Плагін на **оновлення браузера при зміні файлів** (npm I browser-sync --save-dev)
2. Плагін для **роботи із sass I scss:** (npm I gulp-sass --save - dev)
3. Плагін для **добавлення автопрефіксів**: (npm I gulp-autoprefixer --save - dev) – для підтримки даного коду найновішими версіями браузера, добавлення вендерних префіксів
4. Плагін для **збору усіх медіа запитів по їх спільних характеристиках в один файл і запису їх вкінці серверного стильового файлу**, згрупування їх і запису в кінець єдиного файлу при умові використання медіа в файлі розробки, а не в окремому файлі медіа – аикоистовується лише для оптимізації (npm install gulp-group-css-media-queries)
5. Пдагін для **чиски і зжимання Css** стилів: (npm I --save-dev –gulp-clean-css)
6. Плагін **для зжимання css:** (npm I gulp-cssnano --save-dev)
7. Плагін для **зжимання html файлів**: (npm I gulp-htmlmin --save-dev)

**Не використані:**

**JS**

**Використані:**

1. Плагін для **зжимання js файлів**: (npm I gulp-uglify --save-dev)
2. Плагін для **зжимання файлів в один** : (npm I gulp-concat --save-dev)

**Не використані:**

1. Плагін для створення sourcemap – завдяки якому ми зможемо через консоль розробника звернутист до не зжатого вихідного файлу: (npm I gulp-sourcemap --save-dev)

**Img**

**Використані:**

1. Плагін для **зжимання (оптимізації ) картинок**: (npm I gulp-imagemin --save-dev)
2. Плагін для **використання формату картинок webp**: (npm I gulp-webp --save-dev)
3. Плагін **полегшує підключку webp картинок в html код**, як звичайну картинку jpg, а решта конструкції записується автоматично в фінальну версію файла
4. Плагін для **використання картинок в форматі webp в css**: (npm I gulp-webpcss --save-dev)
5. Плагін **для кешування** **даних**: (npm I gulp-cache --save-dev)
6. Плагін для **оптимізації png картинок**: (npm I gulp-pngquant --save-dev)

**Не використані:**

1. Плагін для адаптивності картинок: один из самых быстрых модулей Node.js для **изменения размера изображений JPEG, PNG, WebP и TIFF**.(npm I gulp-responsive --save-dev)
2. Плагін для **работы с JPEG, PNG, WebP и TIFF изображениями. Плагин умеет изменять размер, ориентацию, фон, альфа-канал и многое другое** (npm I gulp-sharp --save-dev)
3. Плагін, которая **автоматически преобразовывает подключенные в CSS изображения PNG, JPG, SVG в Data URI**. (npm I gulp-imacss --save-dev)

**Icons**

**Використані:**

1. Плагін для **використання спрайтів**: (npm I gulp-svg-sprite --save-dev)

**Не використані:**

1. Плагін для створення всіх типів шрифтів з svg: (npm i gulp-iconfont --save-dev)
2. Плагін для створення спрайтів: (npm I gulp-spritesmith --save-dev)

**Fonts**

1. Плагін для **роботи зі** **шрифтами конвертація з ttf2 в woff i woff2**: (npm I gulp-ttf2woff gulp-ttf2woff2 --save-dev)

**На всі випадки життя:**

Плагін для **видалення файлів**: (npm I del --save - dev)

1. Плагін для **перейменування файлу** над якими проводиться операція в потоці: (npm I --save-dev gulp-rename)
2. Плагін для **компіляції даних із багатьох файлів одного типу в один:** (npm I gulp-file-include --save-dev):

Використовується при розмітці основної html сторінки, як конструктора з різних блоків, для цього ми в головний файл пыдключаэмо на необхідне ммісце на сторінці конструкцію

**@@include(‘nameHTMLfile’);**

Працює, як шаблонізатор

**NPM пакети для роботи проекту:**

**Трійка пакетів для комфортнішої роботи зі js елементами**

* **Підключаються і на рівні scss|sass I js**

1. Nmp I Normalize.css
2. Npm I slick-carousel
3. Npm I magnific-popup

**Елементи для налаштування функціоналу:**

1. Для **піключення плагінів в код** використовуємо даний синтаксис:

Let valueName = require (‘plaginName’)

1. **dest()** – створює файл при першій активації і у всіх наступних активаціях оновлює цей файл, в створену папку він вносить файл взятий із scr();

dest() бере файли з src() і паралельно з цим створює ієрархію папок, вказаних в парамерт і вкладає файл з ісходника в низ цієї ієрархії, при повторному виконанні даного метода він робить оновлення змінених файлів

1. **src()** – фіксують в собі шлях до певного файлу для виконання комад через pipe() з цим файлом
2. **pipe()** – методи, які призначені для запису в них команд
3. **parallel()** – метод для паралельного(одночасного виконання) методів
4. **series()** – метод для послідовного виконання вказаних методів
5. **watch(а, в)** – дозволяє відстежувати змінив файлі в процесі роботи в них і другим параметром ми передаємо метод, який опрацьовує ці зміни
6. **exports.valueName** = function \ valueName which include function – exports. Записує в себе змінні які містять в собі методи або назви самих методів, які ми описуємо для роботи
7. **task(*‘taskName’, function() {}*)** – задача, яка виконується окремо від усіх інших командою в командній строці: gulp taskName

**Другим параметром можна вказати масив тасків, які виконаються перед даним**

1. **let project\_folder = *require(‘path’).basename(\_\_dirname*)-** задає серверній папці назву проекту

**Виникнення і вирішення помилок:**

1. По завершенню встановлення gulp – в gulpfile.js встановлюємо таку строку:

function defaultTask(cb) {

// place code for your default task here

cb();

}

exports.default = defaultTask

Якщо результат не співпадає з даними з сайту, то перевстановлємо весь gulp

Для **очистки всього пакету gulp**: npm cache clean –force

Встановлення **заново пакетний менеджер**: npm I npm -g

Після видаляємо тестовий код вище!

**Алгоритм дій для підключки особливого функціоналу:**

1. **Підключка і зжимання jquery плагінів** при передачі їх на серверну папку
   1. Створюємо папку libs в src для зберігання усіх сторонніх js файлів
   2. Встановити git – якщо не встановлено
   3. Встановити bower: npm I –g bower
   4. В папці проекту через FileZilla створити файл .bowerrc
   5. В файлі в круглих скоюках прописати ‘directory’ : ‘src/libs/’
   6. В термінала прописуємо bower I (і плагіни js , які вам необхідні), в свою чергу вони встановляться в папку вказану в bowerrc файлі (normalize.css | magnific-popup | slick-carousel | slick\_slider)
   7. Імпортим їх в style.(scss|sass|css) без вказування типу файлу(але css версію, якщо є версія файлу із іншим розширенням його можна видалити)

**Додаткові команди:**

1. **npm update --save** - оновлює усі статично активовані пакети