

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ЦИВІЛЬНОГО ЗАХИСТУ УКРАЇНИ

факультет пожежної безпеки

кафедра автоматичних систем безпеки та інформаційних технологій

МЕТОДИЧНІ ВКАЗІВКИ
до виконання індивідуального завдання
з навчальної дисципліни

"Інформаційні технології в практиці наукових досліджень"
(назва навчальної дисципліни)

циклу загальної (вибіркової) підготовки
(загальної/професійної, обов'язкової/вибіркової)

за третім освітньо-науковим (доктор філософії) рівнем вищої освіти
(назва рівня вищої освіти)

галузь знань

26 "Цивільна безпека"
(шифр і назва)

спеціальність

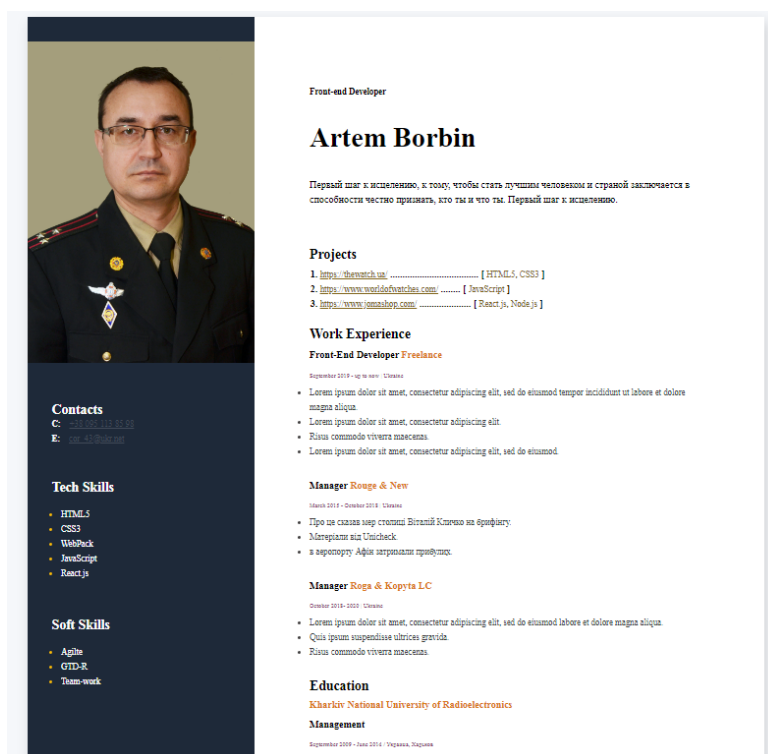
261 "Пожежна безпека"
(шифр і назва)

за освітньо-науковою програмою

Пожежна безпека
(назва програми)

2020

ЗАВДАННЯ: за індивідуальним проектом розробити персональну сторінку зі своїми досягненнями за визначеним макетом:



Потрібно самостійно встановити на ПК та налаштувати редактор коду *VSCode*, створити структуру проекту з особистим дизайном макету свого персонального сайту-резюме.

Для розгляду та аналізу макету можна використовувати онлайн-редактор *Avocode*.

У разі потреби створення скріншотів можна встановити та використовувати безкоштовну програму <http://joxi.ru>.

Виконання завдання треба починати верстування на HTML, але без стилів (без CSS).

1. СТВОРЕННЯ СТРУКТУРИ ПРОЕКТУ

1.1. Теги

Теги – це найпростіші елементи розмітки вебсторінки, щоб браузер при її завантаженні розумів, як саме відображати той чи інший елемент (текст, зображення, відео).

Усі теги мають однаковий формат: вони починаються знаком "<" і закінчуються знаком ">". Зазвичай є два тега – відкриває: `<h1>` і закриває: `</h1>`. Різниця в тому, що в закриває є слеш "/". Весь вміст, поміщене між відкриває і закриває тегами, є вмістом тега.

Тег `<h1> </h1>` визначає заголовок, наприклад:

```
<h1>Hello world!</h1>
```

1.2. Базова структура сторінки

Спочатку потрібно створити файл *index.html*. Другий крок – поставте **!** і натисніть *tab*.

```
index.html •
index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8" />
5  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
7  <title>Document</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
13
```

1.3. Теги верхнього рівня

Ці теги призначені для формування структури веб-сторінки і визначають розділ заголовка і тіла документа.

html – кореневий елемент документа, як контейнер який містить в собі весь вміст веб-сторінки, включаючи теги *head* і *body*.

head – призначений для зберігання технічної інформації про сторінку: заголовок, опис, ключові слова для пошукових машин, кодування і т.д. Вся ця інформація не відображається у вікні браузера, однак містить дані, які вказують браузеру, як слід обробляти сторінку.

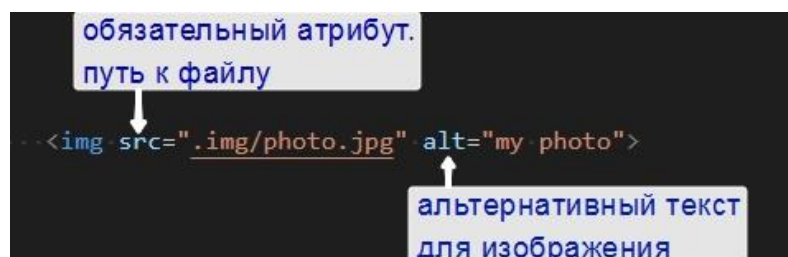
body – призначений для зберігання змісту веб-сторінки, що відображається у вікні браузера. Контент, який повинен відображатися на сторінці, слід розташовувати саме всередині контейнера *body*. До такого контенту відносяться текст, зображення, таблиці, списки та ін.

1.4. Верстування

1.4.1. Атрибут тега

Атрибути дозволяють змінювати властивості і поведінку елемента, для якого вони задані. Загальна кількість атрибутів досить велике, але їх значення, як правило, можна згрупувати за різними типами, наприклад, задають розмір, адресу та ін.

Атрибути записуються всередині відкриваючого тега, а їх значення розташовуються усередині лапок. Атрибутів може бути декілька. Деякі теги не можуть використовуватися без атрибутів.



The diagram shows an HTML tag: ``. Two callouts with arrows point to parts of the tag: one points to `src` with the text "обязательный атрибут. путь к файлу" (mandatory attribute. path to the file), and the other points to `alt` with the text "альтернативный текст для изображения" (alternative text for the image).

1.4.2. Тег `<div>`

Універсальний блоковий парний тег призначений для групування елементів документа. У більшості ситуацій використовується як блок-обгортка для логічного зв'язку контенту

1.4.3. Тег <aside>

Визначає блок збоку від контенту для розміщення рубрик, посилань на архів, міток та іншої інформації. Такий блок, як правило, називається «сайдбар» або «бічна панель»

1.4.4. Тег

Тег `img` непарний, призначений для відображення на веб-сторінці зображень в різних графічних форматах.

Атрибути:

src – обов'язковий і вказує браузеру, де можна знайти зображення. Як і тег `a`, шлях до картинки може бути абсолютним, або відносним;

alt – є альтернативним описом. Це необхідно для надання значимої інформації для користувачів, які не можуть бачити зображення (наприклад, якщо вони погано бачать), або якщо картинка просто не завантажилася;

width – задає ширину зображення;

height – задає висоту зображення.

1.4.5. Теги і

Маркований список:

- *ul* – створює маркований список, кожен елемент якого починається з невеликого символу - маркера;

- *ol* – створює нумерований список, тобто кожен елемент списку починається з числа. Браузер нумерує елементи по порядку автоматично і якщо видалити один або кілька елементів такого списку, то інші номери будуть автоматично перераховані;

- *li* – визначає окремий елемент списку. Може перебувати тільки між `<ul>` і `</ul>` або `<ol>` і `</ol>`.

1.4.6. Тег <a>

Призначений для створення посилань. Всередину тега `<a>` поміщається текст, який буде відображатися на веб-сторінці. Текст посилання відображається в браузері з підкресленням, колір шрифту - синій, при наведенні на посилання курсор миші змінює вид.

Атрибути:

- *href* – обов'язковим атрибутом, задає адресу веб-сторінки або якогось ресурсу (може містити як абсолютний так і відносний шлях);

- *download* ("шлях до файлу") – повідомляє браузеру, що вказаний в *href* ресурс повинен бути завантажений в момент, коли користувач клацає по посиланню. Ставлячи ім'я для атрибута, ми таким чином задаємо ім'я завантажуваного файлу. Можна використовувати атрибут без вказівки його значення;

- *target* – вказує на те, в якому вікні повинен відкриватися документ, до якого веде посилання. Цікавим значенням можна виділити *_blank*, при якому сторінка відкривається в новому вікні браузера;

- *rel* – доповнює атрибут *href* інформацією у цій справі між поточним і пов'язаним документом.

У посилань є можливість не тільки переходити на інші сторінки і завантажувати файли, але і здійснювати дзвінки на телефони, відправляти повідомлення або дзвонити по скайпу тощо.

1.4.7. Тег

Рядковий, підпорядковується правилу потоку документа "зліва направо" і займає ширину по своєму контенту. Призначений для визначення сатиричного елемента всередині

документа, не замінює тег *p*. Може використовуватися для угруповання і виділення сатиричного контенту всередині абзацу для подальшої стилізації.

1.4.8. Тег <p>

Універсальний контейнер для угруповання дрібних фразових елементів, відділення їх один від одного, і подальшої стилізації. Тег парний. Використовується не тільки для вказівки параграфів, це скоріше структурний тег ніж смисловий.

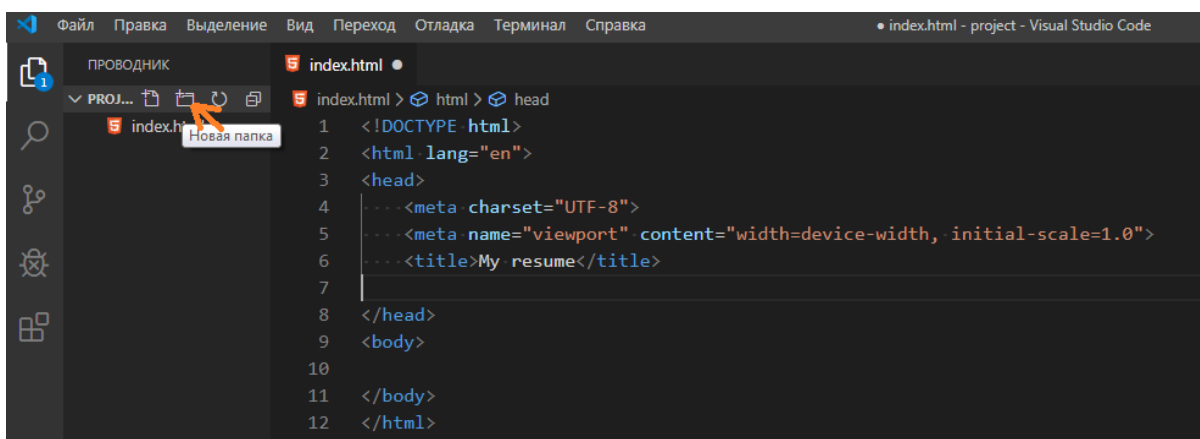
1.5. Коментарі

Для того, щоб закоментувати частина коду або ж залишити підказку для себе, необхідно просто натиснути *ctrl* + */*.

2. ЗАСТОСУВАННЯ СТИЛІВ

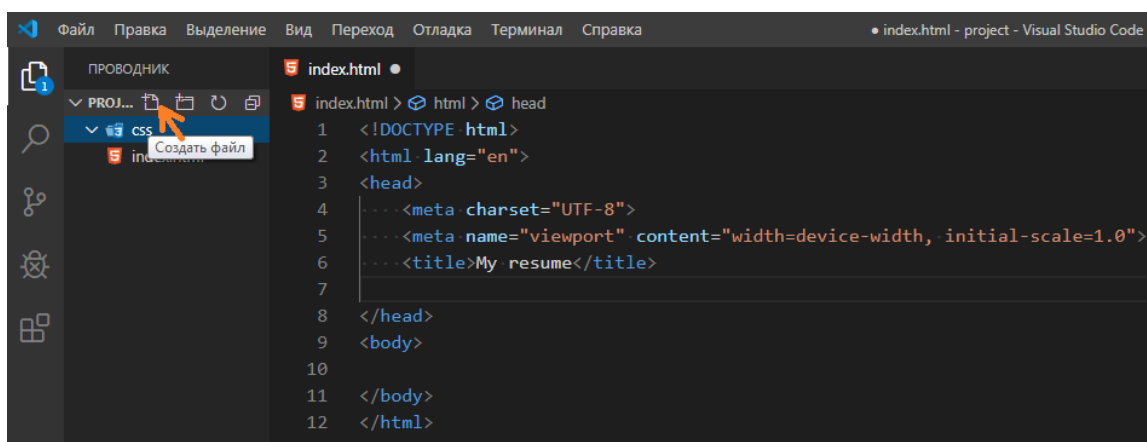
CSS (Cascading Style Sheets, каскадні таблиці стилів) - це мова, яка використовується для опису зовнішнього вигляду документа. HTML використовується для визначення структури і семантики вмісту документа, а CSS для його оформлення і позиціонування.

Спочатку необхідно створити папку *css*, а в ній файл *style.css*. Це зручно зробити в редакторі *Visual Studio Code*:



Вказуємо назву папки *css*.

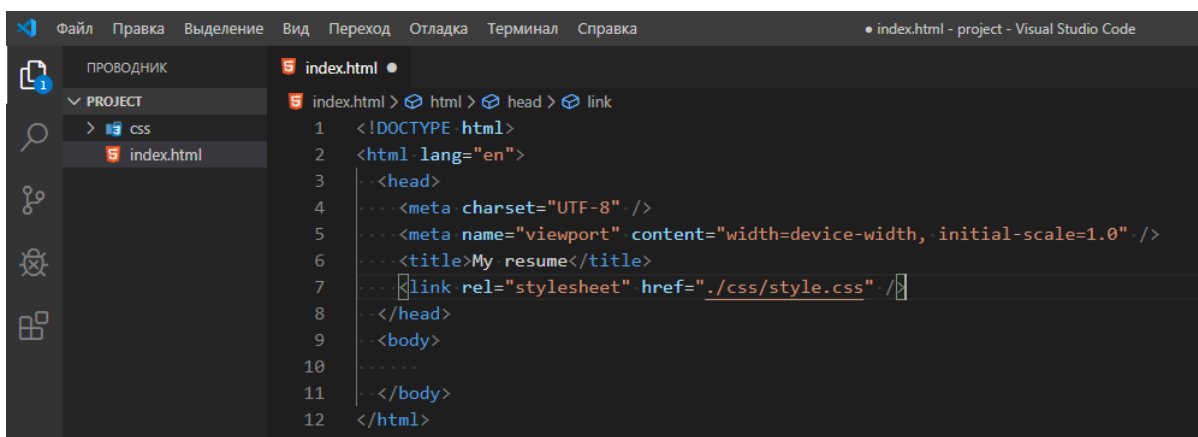
Після створення папки заходимо в неї, створюємо файл *style.css*



Після цього, необхідно підключити файл стилів. Для цього перед *</ head>* прописуємо:

```
<Link rel = "stylesheet" href = "../css / style.css">
```

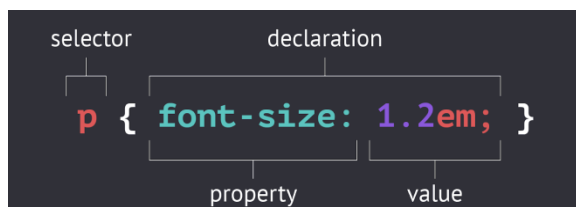
У нас вийде наступне:



Далі вже можна приступити до роботи в файлі `style.css`!

2.1. Синтаксис CSS

Блок CSS-коду називається правилом, яке складається з селектора і групи оголошень в фігурних дужках.



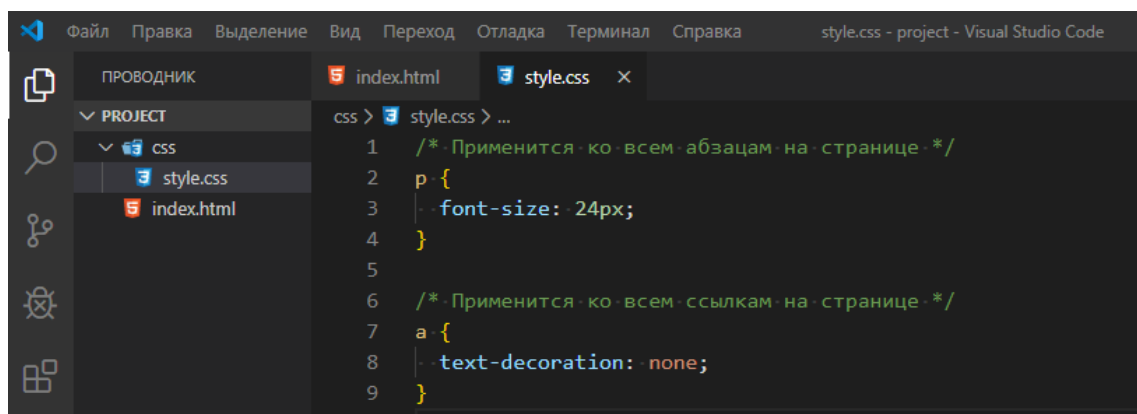
Селектор вказує браузеру на те, до яких елементів потрібно застосувати це правило. Пари властивість-значення називаються оголошеннями, розділяються двокрапкою з пробілом і обов'язково закінчуються крапкою з комою. Вони описують, що саме потрібно зробити зі стилями елемента.

2.2. Селектори

За допомогою різних селекторів можна вибрати той чи інший елемент або групу елементів. Селектор досить багато, ми познайомимося з селектором елемента (тега) і класу.

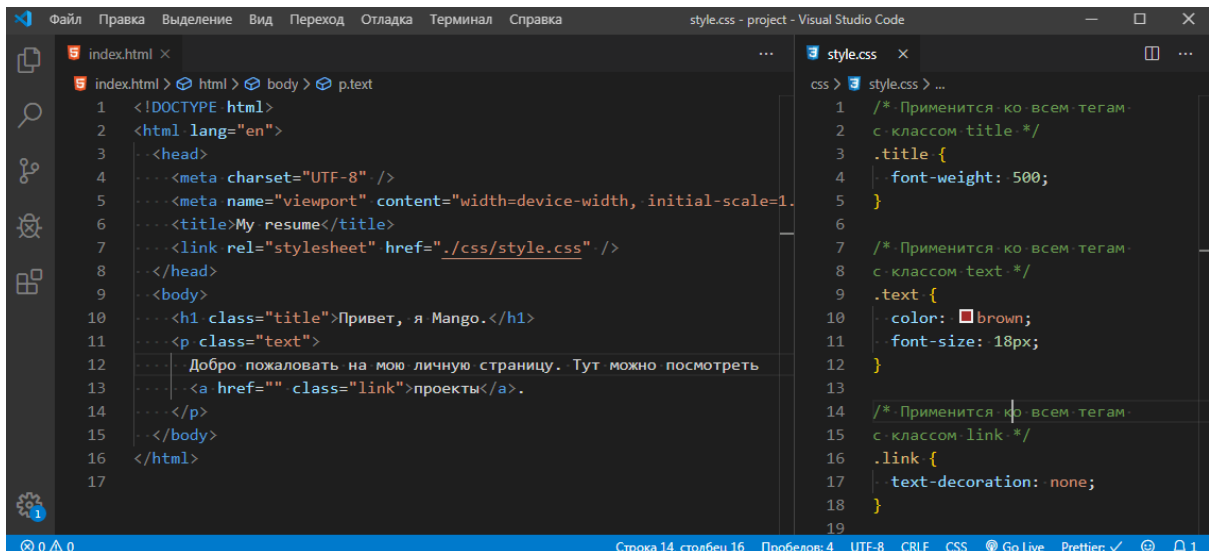
2.2.1. Селектор тега

Описує, до якого типу елемента буде застосовано CSS-правило. Найчастіше застосовується для оформлення всіх однотипних елементів базовими властивостями. Наприклад, прибрати підкреслення у посилань.



2.2.2. Селектор класу

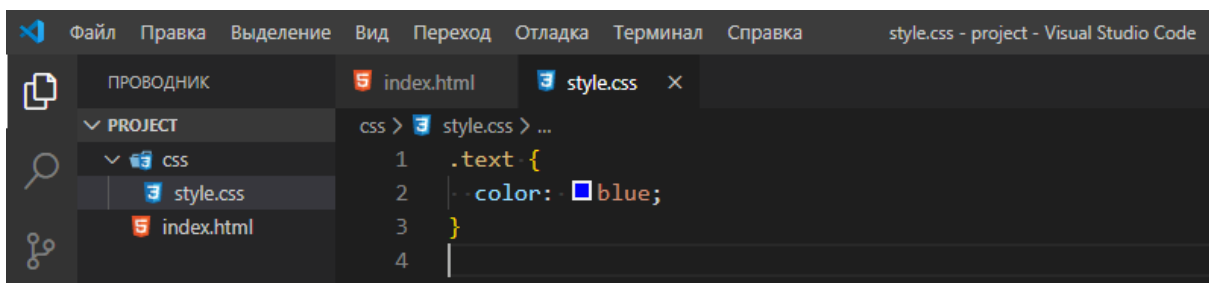
Основний селектор в сучасній розробці. Використовується в зв'язці з глобальним атрибутом *class*. Ім'я класу задається тільки англійською мовою, це обов'язково іменник описує, що це за елемент. Для поповнення словникового запасу доцільно використовувати довідник, наприклад, <https://github.com/yoksel/common-words> і словник термінів, наприклад, <https://github.com/web-standards-ru/dictionary/blob/master/dictionary.md>.



The screenshot shows the Visual Studio Code editor with two files open: `index.html` and `style.css`. The `index.html` file contains an HTML document with a title "My resume" and a paragraph with the text "Привет, я Mango." The `style.css` file contains CSS rules for the `.title` and `.text` classes. The `.title` class has a font-weight of 500, and the `.text` class has a color of brown and a font-size of 18px. The status bar at the bottom indicates the current line and column (14, 16) and the encoding (UTF-8).

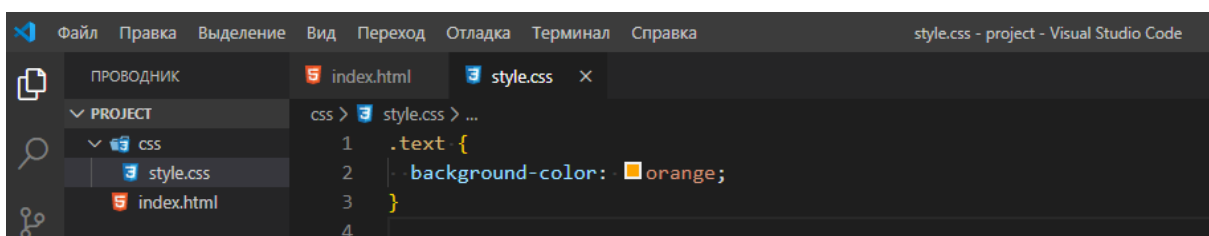
2.3. Робота з кольорами

За допомогою властивості `color` можна змінити колір тексту елемента. Сучасні браузерери підтримують кілька основних способів опису кольору: ключові слова, *rgb*.



The screenshot shows the Visual Studio Code editor with the `style.css` file open. The file contains a CSS rule for the `.text` class, setting the `color` property to `blue`. The status bar at the bottom indicates the current line and column (14, 16) and the encoding (UTF-8).

Властивість `background-color` дозволяє змінити колір фону елемента.



The screenshot shows the Visual Studio Code editor with the `style.css` file open. The file contains a CSS rule for the `.text` class, setting the `background-color` property to `orange`. The status bar at the bottom indicates the current line and column (14, 16) and the encoding (UTF-8).

2.4. Робота з текстом

Шрифт – це набір друкарських літер, цифр і спеціальних знаків певного стилю для набору тексту.

Шрифти ділять на чотири основні групи (сімейства): шрифти із зарубками (*serif*), шрифти без зарубок (*sans-serif*), декоративні та рукописні. Mac, Linux і Windows містять

набір стандартних шрифтів з числа дітей. За замовчуванням, якщо жоден інший шрифт не заданий, браузер використовує Times New Roman з сімейства із зарубками (serif).

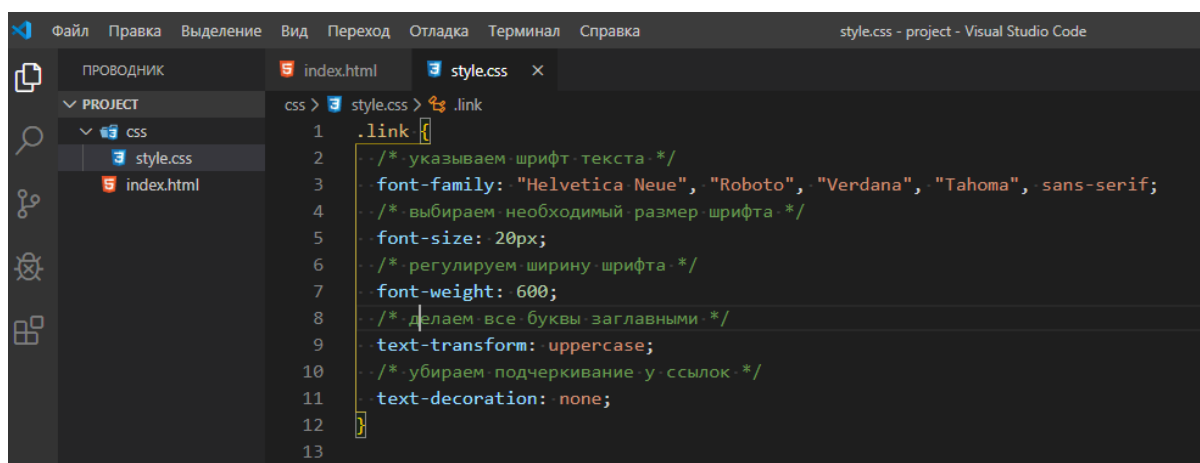
Властивість *font-family* задає шрифт тексту елемента. Через кому можна перерахувати довільну кількість шрифтів, які браузер, в порядку зліва направо, спробує знайти в системі користувача і застосувати перший знайдений. В кінці списку завжди вказується ціле сімейство шрифтів, на той випадок якщо жодного перерахованого шрифту не знайдено. Тоді застосується стандартний шрифт з цього сімейства який є в системі користувача.

font-size задає розмір шрифту елемента в абсолютних або відносних одиницях виміру. Поки що, обмежимося абсолютними одиницями – *px* (пікселями).

font-weight контролює товщину накреслення (жирність, вага) букв шрифту. Значення властивості можна задавати набором зарезервованих слів або цифрами від 100 до 900 з кроком в 100, які задають ступінь товщини гліфа від тонкого 100 до товстого 900.

Text-transform управляє капіталізації тексту в тексті. Наприклад, зробити всі букви великими або прописними. У розмітці текст завжди набирається згідно з правилами мови. Після чого, за допомогою *text-transform*, приводиться до потрібного вигляду, тому що все оформлення повинно бути в CSS: *text-transform: none | uppercase | lowercase | capitalize* (вибираємо потрібне).

Text-decoration – задає різне оздоблювальні підкреслення і інші ефекти тексту: *text-decoration: none | underline | line-through | overline* (вибираємо потрібне).



Шрифти за адресою, наприкладі шрифту "Montserrat":
<https://fonts.google.com/?sidebar.open&selection.family=Montserrat:wght@300;400>.

Для безкоштовного хостінгу, з метою забезпечення вільного доступу користувачів мережі до розробленої сторінки, можна використати ресурс <https://app.netlify.com>.