**Правила оформлення CSS-коду**

Хороший код:

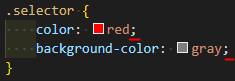
* це не той, який зручно писати, а той, який зручно читати і підтримувати.
* це простий код, прості короткі селектори і «говорящі» класи, за якими відразу можна зрозуміти до чого застосовуються стилі конкретного блоку.
* це акуратний код, в якому все зроблено однаково: скрізь використовується той самий порядок властивостей, відступи та пробіли, стиль запису кольорів та псевдоелементів.

1. Синтаксис

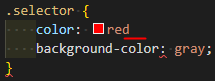
1.1 Наприкінці рядка має стояти крапка з комою

Після пари «властивість: значення» обов'язково ставиться крапка з комою. Без цього розділового знака не буде працювати правило в цьому рядку і наступне за ним.

Добре: після кожного значення стоїть крапка з комою



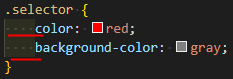
Погано: після першої властивості пропущено крапку з комою



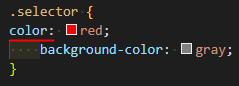
1.2 Для відступів всередині правил використовуйте два пробіли

Правила, що перераховуються всередині фігурних дужок, повинні бути відсунені від початку рядка. Для цього використовуйте 2 (або 4) пробіли. Це дозволяє відразу бачити блоки властивостей, що належать до одного селектора.

Добре



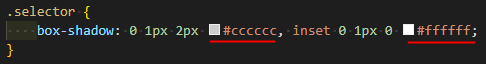
Погано



1.3 Значення кольорів не скорочується

Якщо колір заданий в шістнадцятковій системі, значення не скорочується, а пишеться повністю код з усіх шести символів. Для запису використовуйте малі літери. Наприклад, #e3e3e3.

Добре



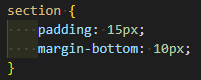
Погано



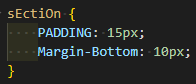
1.4 Все пишеться малими літерами

Всі назви тегів і властивості пишуться малими літерами. Хоча html та css регістро-незалежні.

Добре



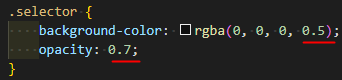
Погано



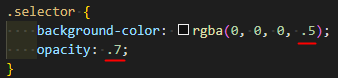
1.5 Нулі не пропускаються

Якщо число дробове і починається з нуля, він не опускається (наприклад, .5 замість 0.5). хоча деякі редактори дозволяють такий запис.

Добре



Погано



1.6 Використовуйте подвійні лапки

У стилях завжди використовуються подвійні лапки. Якщо лапки не обов'язкові, то вони пишуться однаково.

Добре



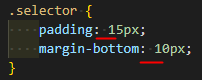
Погано



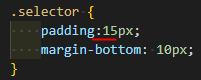
1.7 Пробіл після двокрапки

У правилах після двокрапки ставиться один пробіл (top: 10px;). При цьому перед двокрапкою пробіл не потрібен.

Добре



Погано



1.8 Пробіли після коми у кольорах

Після ком всередині значень rgb(), rgba(), hsl(), hsla() або rect() пробіли ставляться. Це покращує читабельність.

Добре



Погано



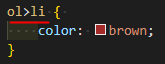
1.9 Пробіл до та після комбінатора

Між селекторами до та після комбінатора (наприклад, p > a) ставиться один пробіл.

Добре



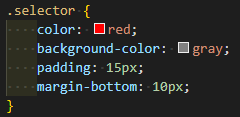
Погано



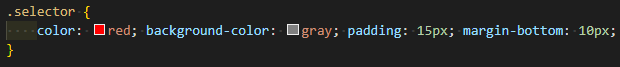
1.10 Кожна властивість пишеться з нового рядка

Одна властивість – один рядок. Кожне оголошення у правилі пишеться з нового рядку.

Добре



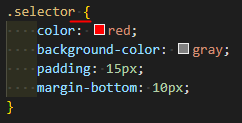
Погано



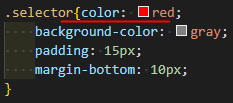
1.11 Пробіл перед фігурною дужкою

Після селектора і перед фігурною дужкою, що відкривається, ставиться один пробіл. Після дужки, що відкрилася, запис йде з нового рядка.

Добре



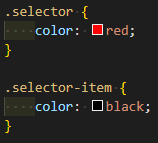
Погано



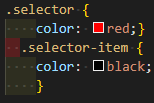
1.12 Фігурна дужка, що закриває, на новому рядку

Фігурна дужка, що закриває, після набору властивостей пишеться на новому рядку і без відступу. Вона має бути на одному рівні з селектором. Наступне після цього правило відокремлюється порожнім рядком.

Добре



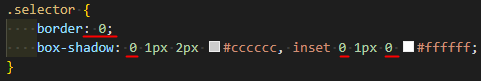
Погано



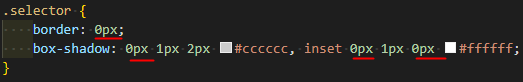
1.13 Опускайте одиниці виміру

Одиниці виміру не потрібно писати там, де без них можна обійтися. Наприклад, border: 0.

Добре



Погано



2. Порядок властивостей

Порядок логічно пов'язаних властивостей має бути згрупований наступним чином:

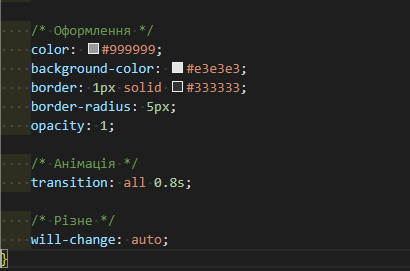
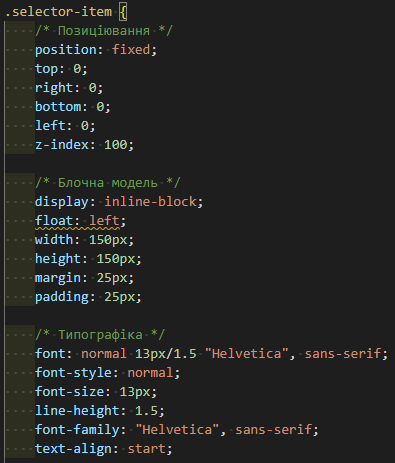
* Позиціювання
* Блокова модель
* Типографіка
* Оформлення
* Анімація
* Різне

Позиціонування слідує першим, оскільки воно впливає на положення блоків у потоці документа. Блокова модель визначає розміри та розташування блоків і йде наступною.

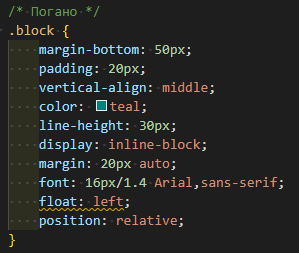
Всі інші оголошення, які змінюють вигляд внутрішніх частин блоків та не впливають на інші блоки, йдуть в останню чергу.

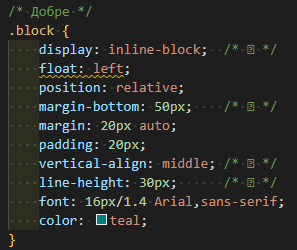
Згруповані властивості у правилі відокремлюються одна від одної порожнім рядком.

Порядок оголошення деталізованих правил, таких як font-size, font-family, line-height повинен відповідати порядку в скороченій версії правила. У разі спільного використання деталізованих і скорочених правил першою повинна йти скорочена версія.



Властивості слід групувати за змістом, і ось чому це важливо:





Якщо взяти перший код і розставити в ньому властивості по порядку, відразу виявляється купа помилок: display: inline-block не працюватиме з float: left, тому що в цьому випадку елемент стає блоковим. З цієї ж причини не працюватиме vertical-align. margin-bottom не спрацює, тому що нижче перезаписується скороченим записом властивості margin. line-height не працює, тому що перезапишеться коротким записом font.

Достатньо згрупувати властивості за змістом, щоб ці помилки стали видно як на долоні, тому намагайтеся завжди дотримуватись певного порядку властивостей.

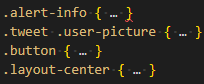
3. Імена класів

Імена класів пишуться малими літерами, між кількома словами використовується дефіс (але не знак нижнього підкреслення або camelCase). Дефіси є роздільниками у взаємопов'язаних класах (наприклад, .button і .button-cancel).

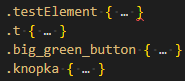
Імена повинні бути такими, щоб по них можна було швидко визначити, якому елементу на сторінці заданий клас: уникайте скорочень (єдиний виняток – .btn для кнопок), але не робіть їх занадто довгими (не більше трьох слів).

Для іменування класів використовуються англійські слова та терміни. Не використовуйте трансліт для назви класів та атрибутів.

Добре



Погано



4. Правило @import

Правило @import працює повільніше, ніж тег <link>. У стилях @import використовувати не бажано.

Добре: підключення тегом link



Погано



5. Варіанти шрифту

Альтернативні варіанти шрифту та тип сімейства вказуються наприкінці перерахування значень font-family.

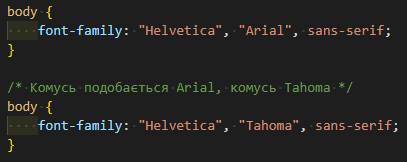
У разі використання нестандартних шрифтів обов'язково вказувати альтернативний веб-шрифт і тип сімейства, щоб у разі відсутності нестандартного шрифту в системі зміни зовнішнього вигляду сторінки були мінімальними. Нестандартний та альтернативний шрифти має бути одного типу.

Порядок шрифтів наступний:

* нестандартний шрифт;
* веб-безпечний;
* тип сімейства шрифту.

Список веб-безпечних шрифтів можна переглянути тут (<https://www.cssfontstack.com/>).

Добре: вказано альтернативний веб-безпечний шрифт та його тип сімейства



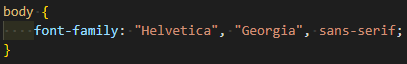
Погано: вказано лише нестандартний шрифт



Погано: вказано лише нестандартний шрифт та тип сімейства, альтернативний веб-безпечний шрифт відсутній



Погано: Georgia – шрифт із засічками, а нестандартний шрифт – без засічок



Детально можна глянути (<https://codeguide.academy/html-css.html#css>).