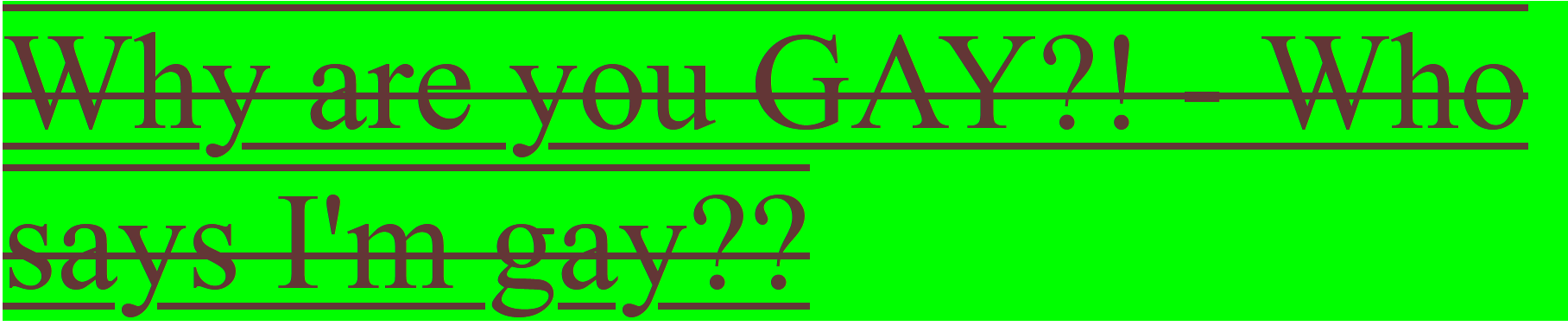
Несколько значений свойства указываются через пробел

text-decoration: overline underline line-through;



Как задавать цвет

**HEX -** #FFFFFF (RRGGBB) – 3 шеснадцатиричных числа, которые задают уровень красного, зеленого и синего

Возможно сокращение до 3х символов, если они повторяются парами #FFF

**RGB**

rgb(255, 0, 0)

rgb(100%, 50%, 50%).

rgba(255, 255, 255, 0.5) – с альфа каналом (прозрачность) 0-полностью прозрачен, 1 непрозрачен

**HSL (hue - оттенок, saturate - насыщенность, lightness - светлость)**

H(0-359) – определяется на цветовом круге, задается в градусах. 0° соответствует красному цвету, 120° — зелёному, а 240° — синему

S(0-100%) - Значение 0% обозначает отсутствие цвета и оттенок серого, 100% максимальное значение насыщенности.

L(0-100%) - Малые значения делают цвет темнее, а высокие светлее, крайние значения 0% и 100% соответствуют чёрному и белому цвету.

Hsl(240, 50, 50)

Hsla(240, 50, 50, 0.5) – с альфа-каналом (прозрачность)

**Через названия** (Red, blue, green)

**РАЗМЕРЫ**

Абсолютные –

* px (пиксели)
* pk
* cm (сантиметр) = 38px – раньше использовали для версий печати.
* mm (миллиметр) = 3.8px
* dm

Относительные –

* % - определение размеров относительно размеров родительского блока, либо величин заданных браузером.
* Rem – root element (вычисляется от размера шрифта корневого элемента html, если не указано – относительно браузера. Для использования rem можно установить базовый размер шрифта, а затем применять rem для определения цветов относительно базового размера.),
* em – относительно шрифта родителя. 1em – это размер шрифта родительского блока.
* Vh – высота в процентах относительно viewport (размеров области просмотра), 1vh = 1% от высоты окна.
* Vw – ширина в процентах от экрана, 1 vw = 1% от высоты окна.
* vmin – использует размер меньшей стороны, то есть если высота окна браузера меньше ширины, 1 vmin=1vh. Если ширина меньше высоты, 1vmin=1vw.
* vmax – используют большую сторону.

Исключения!

*При установке свойства margin и padding в % процент берется от ширины родительского блока, даже если задает вертикальные отступы.*

*При установке свойства line-height в % процент берется от текущего размера шрифта, а вовсе не от line-height родителя.*

*Для width/height процент берется обычно от ширины/высоты родителя, но при position: fixed процент задается от ширины/высоты окна (а не родителя или документа).*

Базовым размером шрифта, который по умолчанию применяется большинством браузеров является 16px.

Для определения **размера шрифта** css также предлагает 7 ключевых слов, которые задают размер относительно базового:

* xx-small
* x-small
* small
* medium (базовый 16px)
* large
* x-large
* xx-large

**СВОЙСТВА ДЛЯ ТЕКСТА**

* **font-family –** шрифт:Times New Roman, Georgia, serif, sans-serif
* **font-size –** размер шрифта, **inherit –** наследование от родителя, **initial** – по умолчанию.
* **font-weight** – жирность: 100-200-900, normal – 400, bold – 700, initial, inherit.
* **font**-**style –** стиль шрифта. normal, italic(курсив)
* **letter**-**spacing –** расстояние между буквами: normal, 10px, можно отрицательные -10px, em.
* **word-break** - переносы: normal(стандартный), break-all(при превышении пойдет перевод, между символами) break-word(словами)

**text-decoration (доп оформление):**

* **underline** –подчеркнутый текст внизу(пример)
* **overline** – подчеркнутый текст сверху (пример)
* **line-through** – перечеркнутый текст (~~пример~~)
* **none** – убирает подчёркивание (у ссылок)

**Text-transform(работа с регистрами):**

* **lowercase –** символы текста строчные (нижний регистр).
* **uppercase** – символы текста прописные (верхний регистр).
* **white-space(**обработка пробелов): normal(несколько в один), no-wrap(без переноса)
* **none** – не меняет регистр символов.

**Text-align(выравнивание)**

• left/center/right/justify

**line-height - межстрочное расстояние.**

**Text-indent(ОТСТУП),** ПО ДЕФОЛТУ ТОЛЬКО ПЕРВАЯ СТРОЧКА,

* text-indent: 5em each-line;(ВСЕ СТРОКИ ПОСЛЕ ENTER И BR)
* text-indent: 5em hanging; (ВАЩЕ ВСЕ СТРОКИ КРОМЕ ПЕРВОЙ)

**CЕЛЕКТОРЫ**

Селекторы могут **группироваться** в одной строке через запятую. Тогда стили применяются к каждому из них.

**Виды селекторов:**

* *\* {}* универсальный селектор. Правило применяется к любому элементу. Может использоваться в комбинации с псевдоклассами и псевдоэлементами.
* *p {}* селектор элемента. Применяется ко всем элементам данного типа.
* *.class {}* селектор класса. Применяются к элементам с совпадающим атрибутом class. Если элемент имеет несколько классов, то они разделяются пробелом.
* *#id {}* селектор идентификатора. Применяется к элементу с указанным идентификатором.
* *ul li {}* селектор потомков. Применяется к элементам, расположенным внутри элемента контейнера.
* *p > b* дочерний селектор. Применяется ко всем элементам, являющимися прямыми потомками контейнера.
* *h1 + p* сестринский селектор. Сестринские элементы – те которые имеют общего родителя. Выбирается при этом только первый элемент, следующий после h1.

*h1 ~ p* выберет все сестринские элементы, следующие за h1.

* *p[attribute=”value”]* селектор атрибута. Выбирает элементы на основе имени атрибута или его значения
* *a:active* селектор псевдокласса. позволяют применить CSS-правила к элементам при совершении события или подчиняющимся определенному правилу.
* *p::first-letter* селектор псевдоэлемента.

Псевдоклассы определяют динамическое состояние элементов, которое изменяется с помощью действий пользователя, а также положение в дереве документа.

Условно все псевдоклассы делятся на три группы:

* **определяющие состояние элементов;**

1. active – активация элемента (щелчок мыши);
2. link – непосещенная ссылка;
3. focus – элемент получивший фокус (например курсор в поле для ввода);
4. hover – при наведении мышки на элемент;
5. visited – посещенная ссылка;

* имеющие отношение к дереву элементов;

1. first-child – первый дочерний элемент селектора.

* указывающие язык текста.
  1. lang() – определяет язык, который используется в документе или его фрагменте.

**Псевдоэлемент** в CSS — это ключевое слово, добавляемое к селектору, которое позволяет стилизовать определённую часть выбранного элемента.

* 1. after – применяется для вставки контента после содержимого элемента. Вставляемый контент определяется свойством content в css правиле.
  2. before – для вставки контента до содержимого элемента.
  3. first-letter – определение стиля первого символа в тексте.
  4. first-line – стиль для первой строки блочного текста.

**Наследование –** механизм, с помощью которого значения свойств элемента-родителя передаются его элементам-потомкам. Потомки могут переопределять свой стиль.

Не все свойства наследуются. Чтобы унаследовать ненаследуемые свойства используется слово inherit. Чтобы запретить наследование используется *initial*. В этом случае элемент применит свойства браузера.

*Наследуемые свойства:* отображающее параметры текста (fon-size, font-family, color, text-align, direction), cursor, visibility, list-style

**Каскадирование стилей –** заключается в том, что если к элементу применяется несколько css-правил, то результирующий стиль будет сформирован из всех присутствующих в этих правилах свойств.

В случае конфликтов, будет применено свойство с наибольшим приоритетом.

* + 1. Стиль браузера;
    2. Стиль пользователя (устанавливается в настройках браузера).
    3. Стиль автора страницы

Стили унаследованные от родителей.

Стили заданные во внешних таблицах.

Стили внутренних таблиц в атрибуте <style>.

Инлайн-стили.

Стили помеченные !important.

В случае конфликта на одном уровне, конфликт решается с помощью сравнения специфичности селекторов. Приоритетнее те, что имеют больше

1. идентификаторов;
2. затем классов, атрибутов и псевдоклассов;
3. элементов и псевдоэлементов.

В случае одинаковой специфичности, применяется стиль, расположенный ниже в таблице.