## Оформление текста с помощью CSS

|  |
| --- |
| [Главный текстовый тег — span](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/1) |
| [Свойство font-size: задаем размер шрифта](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/2) |
| [Свойство font-weight: толщина начертания](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/3) |
| [Свойство font-style: курсив](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/4) |
| [Свойство font-family: шрифт](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/5) |
| [Свойство color: цвет текста](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/6) |
| [Свойство text-decoration: подчеркивание и другие эффекты](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/7) |
| [Декоративное подчеркивание](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/8) |
| [Задаем регистр символов с помощью text-transform](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/9) |
| [Управляем пробелами: white-space](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/10) |
| [Горизонтальное выравнивание текста: text-align](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/11) |
| [Вертикальное выравнивание: vertical-align](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/12) |
| [Верхние и нижние индексы на CSS](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/13) |
| [Свойство line-height: управляем высотой строки](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/14) |
| [Вертикальный ритм текста](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/15) |
| [Испытание: рецепт на CSS](https://npoed.htmlacademy.ru/htmlcss110/course/10/run/16) |

# Главный текстовый тег — span [1/16]

При оформлении текста с помощью css чаще всего используют тег <span>. Он обозначает «просто текстовый блок». То есть особенного собственного смысла он не имеет. Также этот тег никак не изменяет отображение текста.

Однако, дополнительный смысл данному тегу добавляют с помощью классов. Например:

<span class="error"></span>

<span class="ok"></span>

А уже для класса с помощью css задают стили и тем самым изменяют оформление.

# Свойство font-size: задаем размер шрифта [2/16]

Cвойство font-size задает размер шрифта.

Размер шрифта лучше всего задавать в em — относительной единице измерения. 1em обычно равен длине буквы M в данном шрифте.

Другие единицы измерения для задания размеров шрифта:

* Пиксели: 20px
* Пункты: 15pt
* Проценты: 80%

Пиксели и пункты — это абсолютные единицы измерения, а проценты — относительные.

Также размер шрифта можно задавать с помощью ключевых слов: small, large и т.д. Но их обычно не используют.

# Свойство font-size: задаем размер шрифта [2/16]

Cвойство font-size задает размер шрифта.

Размер шрифта лучше всего задавать в em — относительной единице измерения. 1em обычно равен длине буквы M в данном шрифте.

Другие единицы измерения для задания размеров шрифта:

* Пиксели: 20px
* Пункты: 15pt
* Проценты: 80%

Пиксели и пункты — это абсолютные единицы измерения, а проценты — относительные.

Также размер шрифта можно задавать с помощью ключевых слов: small, large и т.д. Но их обычно не используют.

# Свойство font-weight: толщина начертания [3/16]

Полужирность текста можно задавать с помощью свойства font-weight, которое имеет два основных значения:

1. normal — обычное начертание;
2. bold — полужирное начертание.

На самом деле это свойство имеет много значений: bold, bolder, lighter, normal, 100, 200, 300, 400, 500, 600, 700, 800, 900. Эти значения задают степень толщины шрифта, от самого тонкого, до самого толстого.

Но большинство браузеров всё равно умеют отображать только два варианта толщины: обычный и полужирный. Поэтому и остальные значения свойства обычно не используют.

# Свойство font-style: курсив [4/16]

Начертание текста можно задавать с помощью свойства font-style. Его основные значения:

1. normal — обычное начертание;
2. italic — курсивное начертание.

Это свойство имеет и другие значения, но их почти не используют.

# Свойство font-family: шрифт [5/16]

Задать семейство шрифта можно с помощью свойства font-family. Можно задавать конкретное название шрифта: "Times New Roman". А можно задавать желаемый тип шрифта, например:

* serif — шрифт с засечками;
* sans-serif — шрифт без засечек.

Есть и другие типы, но они используются реже.

Обычно в качестве значения свойства задают список шрифтов, перечисляя их через запятую. В начале списка располагают самый редкий шрифт, затем похожий, но более распространенный, а в самом конце списка — желаемый тип шрифта. Пример:

body {

font-family: "PT Sans", "Arial", serif;

}

Браузер проходит по списку слева направо и использует первый найденный в системе шрифт.

# Свойство color: цвет текста [6/16]

Цвет текста задаётся с помощью свойства color. Значение цвета можно задавать разными способами:

* Шестнадцатеричным кодом, например #FF9900.
* Ключевым словом: red.
* В RGB-формате: rgb(255, 255, 0).

Чаще всего цвет задают в шестнадцатеричном формате.

# Свойство text-decoration: подчеркивание и другие эффекты [7/16]

Дополнительное оформление текста можно задать с помощью свойства text-decoration. Его значения:

1. underline — подчеркивание;
2. line-through — зачеркивание;
3. overline — черта сверху;
4. none — убирает вышеперечисленные эффекты.

К тексту можно одновременно применить несколько эффектов, если перечислить значения через пробел.

# Декоративное подчеркивание [8/16]

Вы, наверное, уже много раз видели красивое пунктирное подчеркивание. Его используют для оформления ссылок и других динамических элементов. Технология следующая:

1. Убираем обычное подчеркивание с помощью text-decoration.
2. Задаем нужный цвет текста с помощью color.
3. Добавляем декоративное подчеркивание с помощью свойства border-bottom.

Также можно при наведении курсора скрывать такое подчеркивание с помощью псевдокласса :hover.

# Задаем регистр символов с помощью text-transform [9/16]

С помощью css можно управлять даже регистром символов: делать буквы строчными или прописными. Делается это с помощью свойства text-transform. Его значения:

1. lowercase — все строчные;
2. uppercase — все прописные;
3. capitalize — каждое слово начинается с прописной;
4. none — отменяет изменение регистра.

# Управляем пробелами: white-space [10/16]

Как вы уже знаете, браузер игнорирует множественные пробелы и переносы строк в HTML-коде. Изменить это поведение можно с помощью тега <pre>.

Однако, с помощью CSS управлять пробелами и переносами можно более гибко. За это отвечает свойство white-space, значения которого:

* nowrap — отображает весь текст одной строкой без переносов;
* pre — сохраняет пробелы и переносы как в исходном коде аналогично тегу <pre>;
* pre-wrap — работает как значение pre, но добавляет автоматические переносы, если текст не помещается в контейнер;
* normal — режим по умолчанию.

# Горизонтальное выравнивание текста: text-align [11/16]

Выравнивание текста по горизонтали задается с помощью свойства text-align. Его вы уже использовали в разделе про таблицы. Значения свойства:

1. left — выравнивание по левому краю;
2. right — по правому краю;
3. center — по центру;
4. justify — по ширине.

# Вертикальное выравнивание: vertical-align [12/16]

Выравниванием текста по вертикали можно управлять с помощью свойства vertical-align. Его действие хорошо заметно в ячейках таблицы. Внутри текстовой строки «работа» этого свойства заметна, если в ней есть фрагменты разного размера.

У данного свойства много значений, но самые часто используемые:

1. top — выравнивание по верхнему краю строки;
2. middle — по середине;
3. bottom — по нижнему краю;
4. baseline — по базовой линии (значение по умолчанию).

# Верхние и нижние индексы на CSS [13/16]

С помощью CSS можно имитировать теги <sub> и <sup>, которые применяются для создания нижних и верхних индексов. Делается это так:

1. Используем свойство vertical-align со значением sub или super.
2. Немного уменьшаем размер шрифта с помощью font-size.

# Свойство line-height: управляем высотой строки [14/16]

Высотой строки или, правильнее, межстрочным интервалом можно управлять с помощью свойства line-height. Значение этого свойства можно задавать следующими способами:

1. Множителем, например 1.5, 2.
2. В процентах: 150%.
3. С помощью любых других единиц измерения CSS: 12px, 2em.
4. Ключевым словом normal, которое задает автоматический расчет высоты строки.

Предпочтительнее задавать межстрочный интервал либо множителем, либо в относительных единицах измерения.

# Вертикальный ритм текста [15/16]

В веб-дизайне существует понятие «вертикальный ритм текста». Оно достаточно сложное и мы не будем глубоко в него вдаваться. Отметим лишь, что хороший вертикальный ритм улучшает восприятие текста.

Чтобы сохранить вертикальный ритм и хорошую читабельность текста при любом размере шрифта, нужно задавать размеры шрифта, межстрочный интервал и вертикальные отступы между заголовками и абзацами в относительных единицах.

Тогда, как бы пользователь ни изменял размер шрифта, ваш текст останется читабельным и удобным для восприятия. В этом задании вы потренируетесь использовать относительные единицы измерения.

Дополнительные материалы для тех, кто заинтересовался вертикальным ритмом: [О вёрстке вертикального ритма](https://peredelka.wordpress.com/2013/10/29/%D0%BE-%D0%B2%D0%B5%D1%80%D1%81%D1%82%D0%BA%D0%B5-%D0%B2%D0%B5%D1%80%D1%82%D0%B8%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE-%D1%80%D0%B8%D1%82%D0%BC%D0%B0/), [Compose to a Vertical Rhythm](http://24ways.org/2006/compose-to-a-vertical-rhythm/" \t "_blank), [Создание вертикального ритма на сайте](http://pixelgene.ru/tutorials/vertical-rhythm.html), [Генератор вертикального ритма](http://www.gridlover.net/app/).

# Испытание: рецепт на CSS [16/16]

В последнем испытании раздела «Разметка текста с помощью HTML» вам нужно было разметить рецепт каши с помощью подходящих тегов.

В этом испытании вам нужно оформить этот же рецепт. Однако, задача изменилась. Исходный HTML-код уже полностью готов и заблокирован для редактирования. Вам нужно применить изученные в этом курсе CSS-свойства к соответствующим элементам <span>.