Верстка web-страниц

Селекторы

Содержание

1. Селекторы
2. Селекторы типов
3. Селекторы классов
4. ID-селекторы
5. Групповые селекторы
6. Универсальный селектор
7. Вложенные теги
8. Дерево html
9. Селекторы потомков
10. Создание модулей
11. Псевдоклассы и псевдоэлементы: ссылки, фрагменты абзаца, дополнительно
12. Селекторы атрибутов
13. Селекторы дочерних элементов, типов дочерних элементов, смежных элементов одного уровня
14. Селектор :not()

Селекторы

Для задания определенных свойств необходимо для начала обратиться к нужному элементу.

Данное обращение к тому или иному элементу в HTML-документе можно реализовать несколькими способами.

Селектор – контролирует дизайн веб-страницы, определяя элемент, который необходимо изменить

Именно **селектор** используется для форматирования множества элементов одновременно

Селекторы тегов

Селекторы типов (или селекторы тегов)

позволяют определить стиль всех экземпляров конкретного html-элемента

Для задания такого селектора необходимо просто указать название тега – p, h1, img, a,…

Селекторы тегов

Пример

```
h2 {
font-family: “Gill Sans”;
color:#554321;
}
```

Все фрагменты текста на веб-странице, оформленные в виде заголовка второго уровня, будут отображаться заданным шрифтом и цветом // если к ним не будет применено другое стилевое решение.

Селекторы тегов

Пример

P - указываем, к какому элементу обращаемся

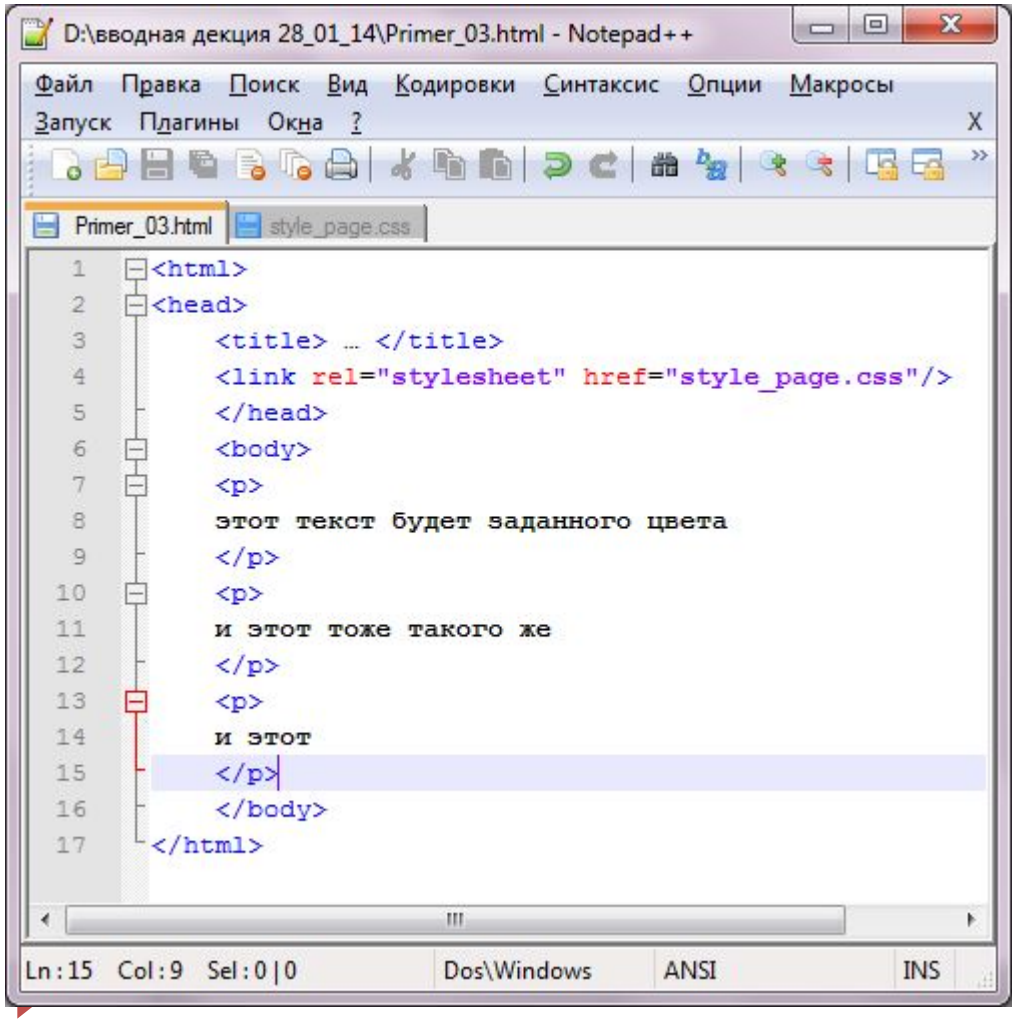
{

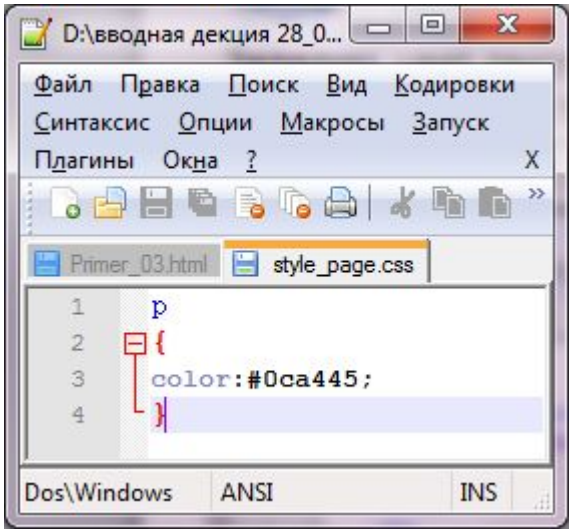
font-size:24px; - абзацы будут отображены шрифтом в 24px

}

Пример

Зададим цвет текста для всех абзацев #0ca445 – просто в описании стилей обратимся сразу к тегу абзаца.





Селекторы классов

Селекторы классов

Позволяют указать на конкретный элемент /группу из элементов/, не зависимо от тегов, с помощью которых эти элементы организованы в html-коде

Пример

- Из пяти абзацев (тег <p>) нужно два выделить синим цветом;
- Один абзац (теги <p> и <h1>) и два заголовка первого уровня нужно отобразить шрифтом Verdana

Селекторы классов

Правила для организации классов

- все названия селекторов классов должны начинаться с точки (.lives);
- при именовании стилевых классов разрешается использование только букв алфавита, чисел, дефиса, знака подчеркивания (.lives_98);
- название после точки всегда должно начинаться с символа – буквы алфавита (неверно: .9got, .-next; верно: .got9, .next);
- Имена стилевых классов чувствительны к регистру (.SIDEBAR и .sidebar рассматриваются как разные классы)

Селекторы классов

Обращение к HTML элементам:

<p **class=“contacts”**>Текст</p>

Пример CSS

. contacts - обращаемся к заданному классу (.имя класса)

{

font-size:24px; - элементы указанного класса будут

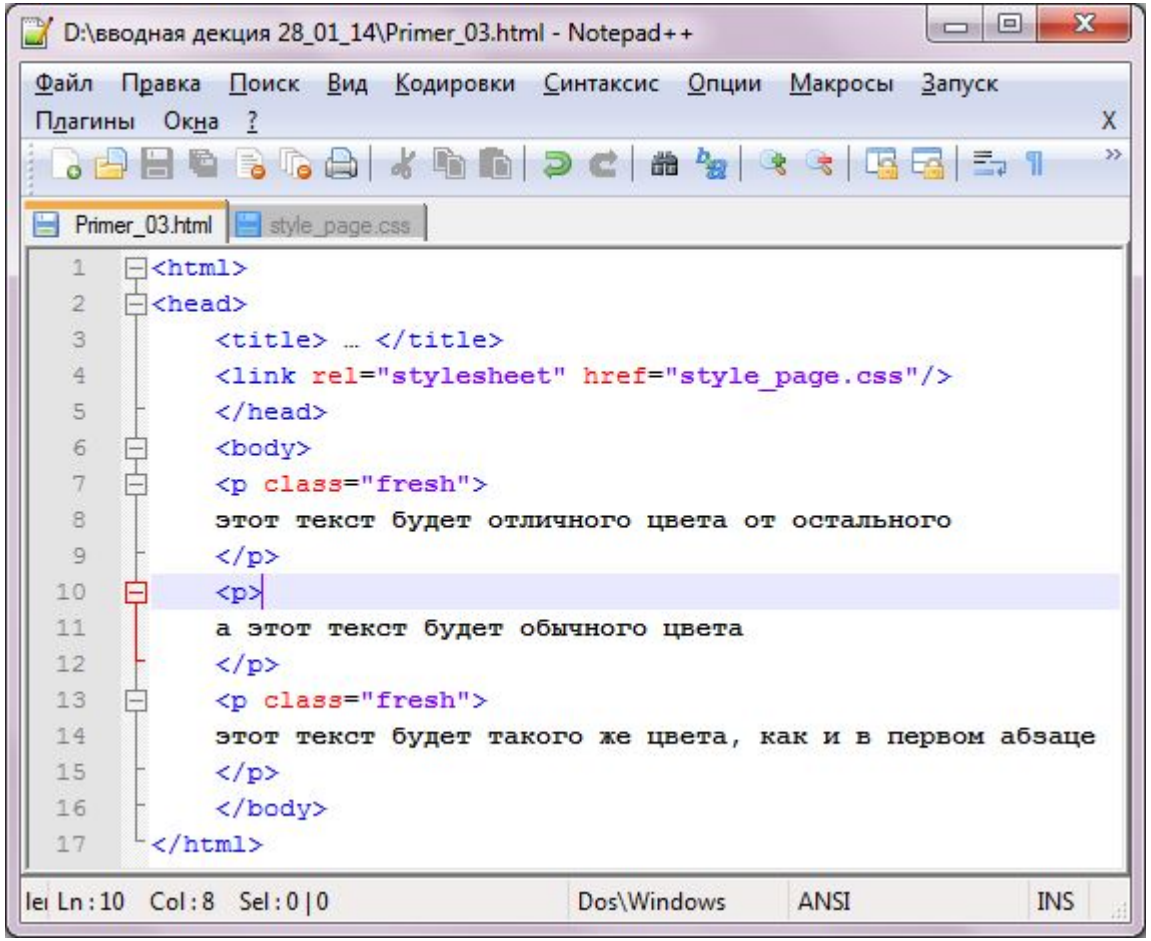
} отображены шрифтом в 24px

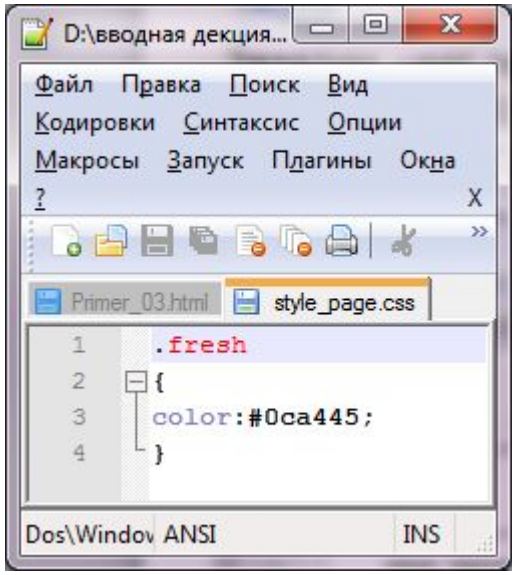
Селекторы классов

Создав класс, можно применить его практически к любому тегу, что дает огромную свободу действий при создании стилевого оформления страницы.

Пример

Зададим цвет текста для нескольких абзацев #0ca445 (этим абзацам мы присвоим class); а остальной текст оставим по умолчанию черным.





ID-селекторы

ID-селекторы

Предназначен для идентификации уникальных частей веб-страниц

Вначале id-селектора **вместо точки ставится знак #**; во всем остальном правила сохраняются те же, что и для классов.

! Более точно указывает на объект, чем селектор класса и чем селектор тега

ID-селекторы

Обращение к HTML элементам:

<p **id=“text”**>Текст</p>

Пример CSS

#text - обращаемся к заданному id (#имя id)

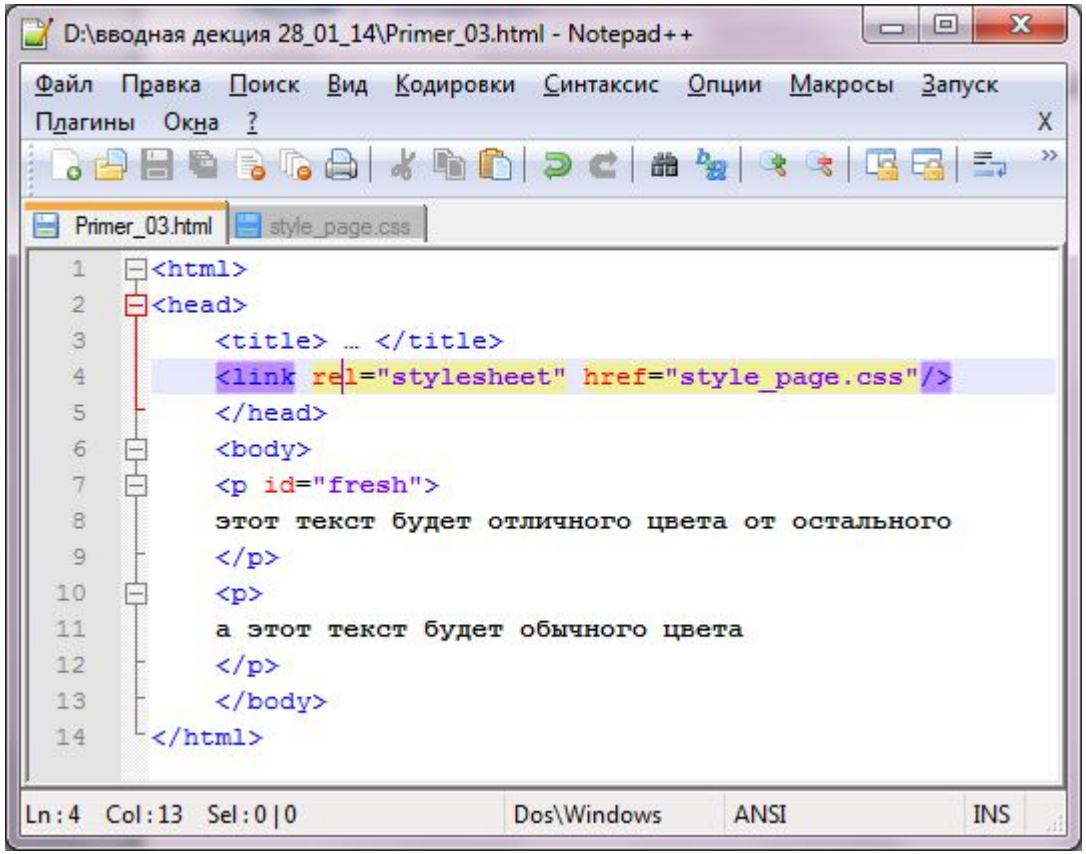
{

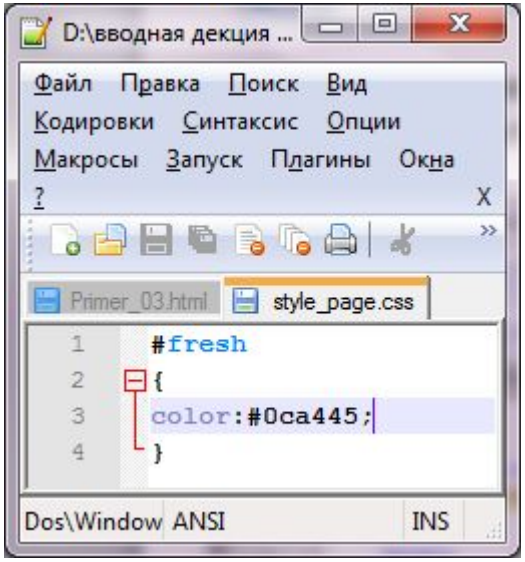
font-size:24px; - для данного элемента шрифт будет

} указанного размера в 24px

Пример

Зададим цвет текста для одного абзаца #0ca445 (этому абзацу мы присвоим id); а остальной текст оставим по умолчанию черным.





ID-селекторы

Для ID есть случаи специального применения:

- при использовании JavaScript позволяют определить место и манипулировать частями страницы (например, id для элементов форм на странице);
- позволяют создавать маркеры на странице и быстро по ней перемещаться (по аналогии с якорным тегом).

Групповые селекторы

Групповые селекторы

Иногда нужно применить одинаковое форматирование к абсолютно разным элементам, причем создание отдельного стиля для каждого не подходит (или дополнение уже созданных стилей одним\двумя свойствами не вариант – нужно прописать это свойства для, к примеру, 20 уже имеющихся стилевых групп) – **тогда создаем групповой селектор**

Групповые селекторы

Групповые селекторы

Создаем список, в котором один селектор отделен от другого запятыми, а дальше по установленному ранее правилу указываем свойства

Пример

```
h1, p, .copyright, #banner {color: #f1cd33;}
```

Универсальный селектор

Универсальный селектор - *

Предоставляет возможность выбора **всех** тегов веб-страницы.

Пример

```
* {color: #f1cd33;}
```

Универсальный селектор

Универсальный селектор - *

Можно использовать для выбора **всех тегов внутри некоего определенного селектора**

Пример

```
.banner * {color: #f1cd33;}
```

– для всех тегов внутри тега с классом banner

Вложенные теги

Стилизация вложенных тегов – селекторы потомков

Их применяют для того, чтобы единообразно отформатировать целый набор тегов в том случае, когда они находятся в определенной структурной области страницы.

Пример

Все теги a (ссылки) из панели навигации

Все изображения из правого вертикального блока

Дерево HTML

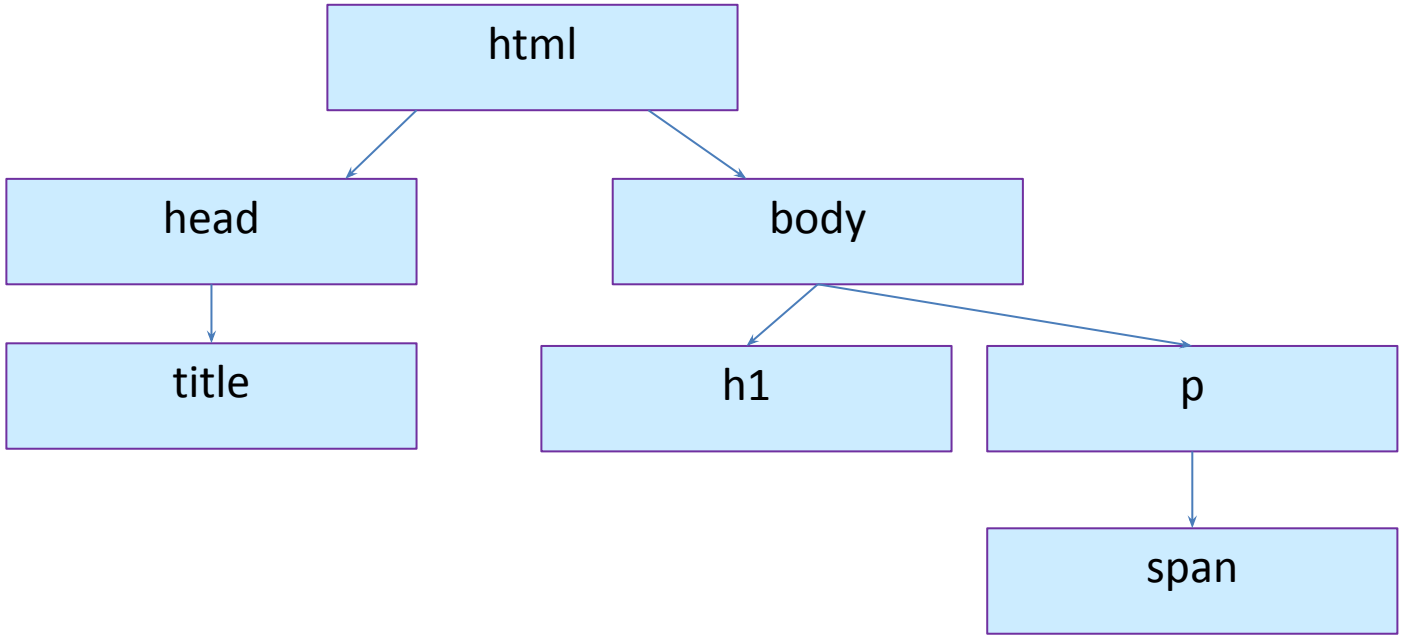
Дерево HTML

Код HTML, на котором написана любая веб-страница, напоминает генеалогическое дерево

Схемы страницы в форме дерева позволяют выяснить и проследить, как CSS видит взаимодействие элементов страницы

Дерево HTML

Дерево HTML. Пример

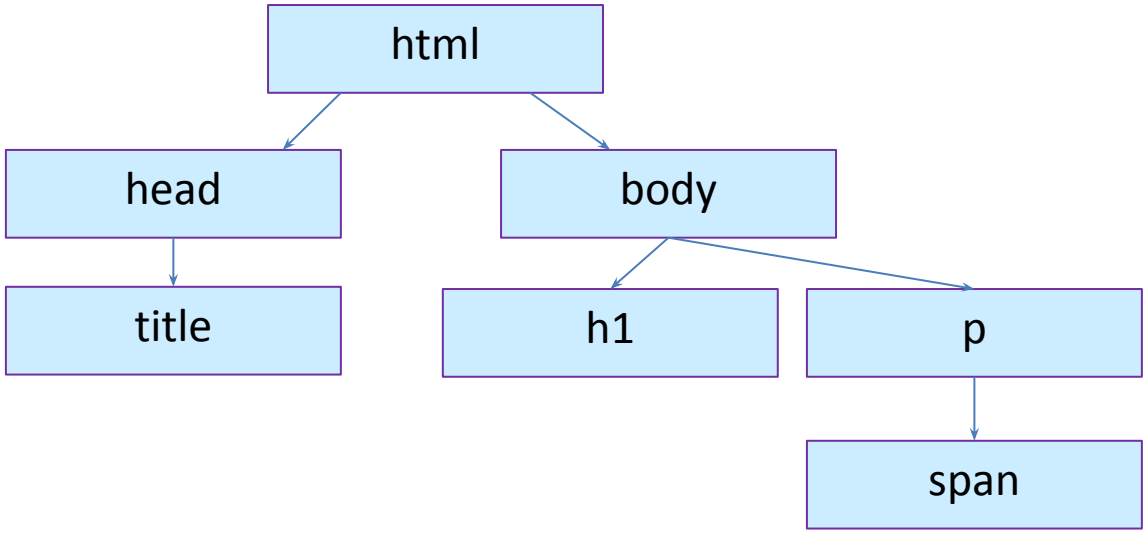


Дерево HTML

Предок – html-элемент, который заключает в себе другие элементы

Пример

body – предок для h1, p, span – всех, содержащихся в нем элементов

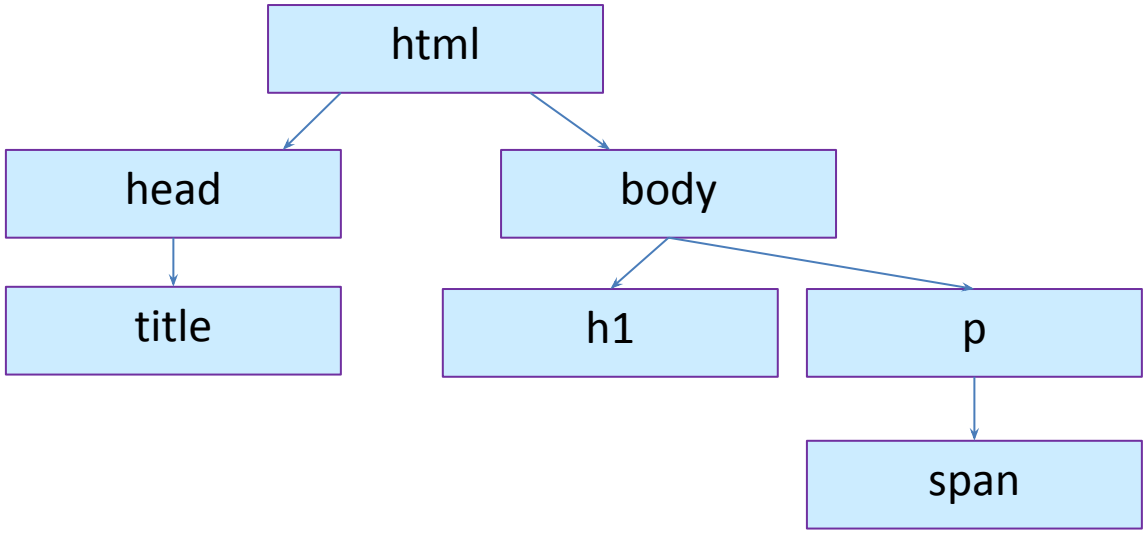


Дерево HTML

Потомок – элемент, который расположен внутри одного или более тегов

Пример

body – потомок для html, p – потомок для body и для html одновременно



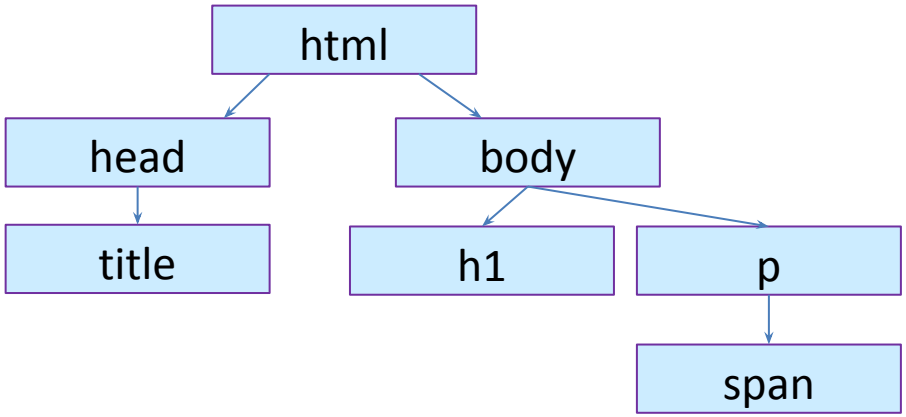
Дерево HTML

Родительский элемент – элемент, который связан с другими элементами более низкого уровня и находится выше на дереве на один уровень

Пример

html – родительский только для head и body

p – родительский по отношению к span



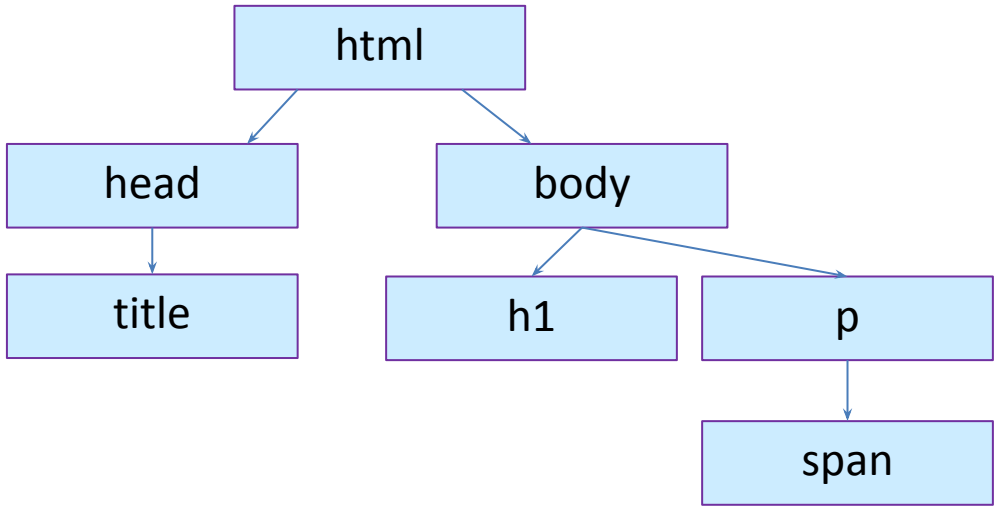
Дерево HTML

Дочерний элемент – элемент, непосредственно подчиненный другому элементу более высокого уровня (сам дочерний элемент находится на уровень ниже родительского)

Пример

h1, p – дочерние по отношению к body

span – не является дочерним для body



Дерево HTML

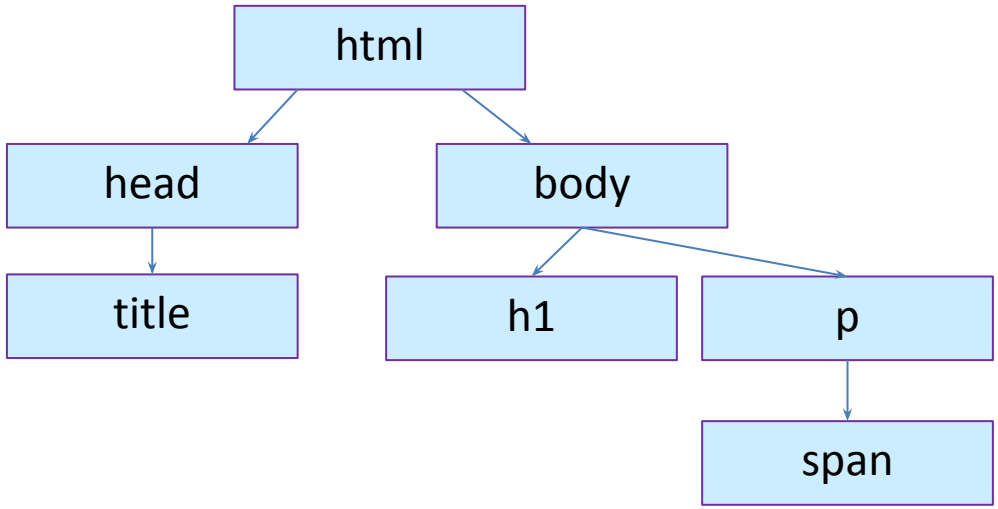
Сестринский элемент – элементы, являющимися дочерними для одного и того же родительского тега (расположены на одном уровне, называются еще соседскими)

Пример

h1, p – сестринские элементы

head, body – сестринские элементы

title, h1, p - не являются сестринскими, т.к. в роли родительского выступают разные теги



Селекторы потомков

Для создания **селектора потомков** указываем структуру вложенности – объединяем селекторы вместе согласно ветви дерева, которую форматируем, **помещая самого старшего предка слева, а форматируемый тег справа**

h1 strong {color: red;}

- изменит цвет только в тех тегах strong, которые находятся внутри тега h1

Вместо тегов при построении селекторов потомков **можно использовать другие селекторы:**

.intro a {color: yellow;}

только для ссылок из объектов класса intro

Создание модулей

Если на странице есть **блок html-элементов**, выполняющих одну функцию, то **для форматирования данного модуля** можно использовать селекторы потомков

Пример, есть фрагмент кода, создающий новости

```
<div class="news">
    <h2>Our company is great!</h2>
    <p>More information about why our company is so great</p>
    <h2>Another news item</h2>
    <p>Information about the other news item...</p>
    <h2>.. and so on...</h2>
    <p>... and so on...</p>
</div>
```

Создание модулей

```
.news h2 {color: red;}
.news p {color: blue;}
```

Создание селекторов потомков позволит для элементов одного модуля новостей задать разное форматирование

Создание модулей

Можно усложнить цепочку селекторов

Пример, есть фрагмент кода, содержащий контактную информацию

```
<div class="contact">
    <p class="name">John Smith</p>
    <p class="phone">555-555-1234</p>
    <p class="address">1234 Elem St</p>
</div>
```

```
.contact .name {font-weight: bold;}
.contact .phone {color: blue;}
.contact .address {color: red;}
```

Создание модулей

ВАЖНО: правильно строить/читать селекторы во избежание неверных результатов в работе

p.intro - **не является** селектором потомков! Между тегом p и классом .intro **НЕТ пробела**

т.е. стили будут применены к тем тегам p, для которых прописан класс .intro (<p class="intro">)

Псевдо

Если есть необходимость выбрать фрагмент веб-страницы, в котором нет тегов, но который достаточно просто идентифицировать – например, первая строка абзаца, ссылка при наведении указателя мыши и др. – тогда можно использовать **псевдоклассы и псевдоэлементы**

Стилизация ссылок

a:link – обозначает любую ссылку;

a: visited – посещенные ссылки;

a: hover – ссылка при наведении;

! Псевдокласс :hover может быть применен к другим объектам веб-страницы, стиль которых должен меняться при наведении указателя – другие теги, классы

a: active – ссылка при ее выборе;

Стилизация фрагментов

Два псевдоэлемента **:first-letter** и **:first-line** позволяют обратиться к фрагментам абзаца – первой букве или первой строке.

:first-letter – позволяет создать буквицу, начальный символ абзаца, который выделяется из остального контекста, как в начале книжной главы

:first-line – позволяет задать стилизацию первой строки

Стилизация фрагментов

ВАЖНО:

В CSS3, чтобы отличить псевдоклассы от псевдоэлементов, было добавлено еще одно двоеточие

::first-letter

::first-line

Однако, для выполнения условия кроссбраузерности, оставляем одно двоеточие

Дополнительно

:focus

Функционирует подобно :hover с той лишь разницей, что применяется **при нажатии клавиши табуляции или выполнении щелчка мыши на текстовом поле** – т.е. пользователь выполняет фокусировку на объекте формы

Преимущественно предназначен для обеспечения обратной связи с посетителями сайта – например, смена цвета заполняемого поля:

```
.FormaS:focus {background-color: #ffffcc;}
```

Дополнительно

:focus

Задает стилевой эффект **только на время**, пока элемент находится в фокусе

Дополнительно

:before

Позволяет добавлять сообщение, предшествующее определенному элементу веб-страницы

Для работы с данным псевдоэлементом можно создать класс (например, .tip) и применить его к тем элементам, которым должно предшествовать сообщение

Пример,

<p class="tip">…

p.tip:before {content: "Это важно знать!";}

Дополнительно

:before

текст, который добавляется данным псевдоэлементом-селектором, еще называют **сгенерированным содержимым**, поскольку браузер создает его.

! Поддерживается IE 8 и выше

Дополнительно

:after

Позволяет добавлять сообщение, **следующее за** определенным элементом веб-страницы

Дополнительно

:after и :before

- также в CSS3 рассматриваются с двойным двоеточием как псевдоэлементы, но для соблюдения условия кроссбраузерности допустимо наличие одного двоеточия

Дополнительно

::Selection

Ссылается на элементы, которые посетитель выбрал на странице.

Пример, выделение текста для последующего копирования

Единственные свойства, доступные для изменения с помощью данного селектора – это color и background-color – чтобы исключить безумства с форматами выделенного фрагмента

Дополнительно

::Selection

Работает только версия с двумя двоеточиями

Не поддерживается IE 8 или Firefox

Для Firefox можно добавить через префикс

```
::-moz-selection {
color: #ffffff;
background-color: #993366;
}
```

Селекторы атрибутов

Есть возможность форматирования тегов **на основе выборки** любых содержащихся в них **атрибутов**

Причем вместо имен тегов можно использовать классы или не просто атрибут, а атрибут с конкретным значением

Пример

```
a[href="http://www.cosmofarmer.com"] {
    color: red;
    font-weight: bold;
}
```

Селекторы атрибутов

Символ ^= означает «начинаться с» позволит усложнить процесс выбора по атрибутам

Пример

img[title^="Content"]

.photo[title^="Autumn"]

a[href^="http://"] – внешние ссылки

a[href^="https://"] – внешние ссылки для защищенного SSL-соединения

Селекторы атрибутов

Символ $= означает «заканчивается на» позволит усложнить процесс выбора по атрибутам

Пример

a[href$=“.pdf”]

Символ *= - позволит выбрать все возможные включения в значение атрибута

Пример

img[src*=“headshot”] – выберет все графические файлы, имеющие в названии сочетание headshot

Дочерние элементы

Селектор дочерних элементов позволяет определить конкретные дочерний и родительский элементы

body **>** h1

p **>** strong

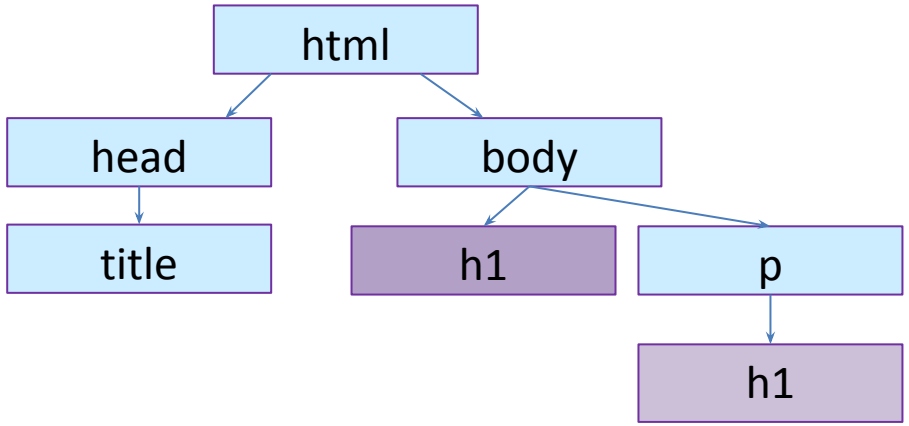
Для дочерних элементов есть несколько **псевдоклассов**

Дочерние элементы

:first-child – позволяет выбрать и форматировать первый дочерний элемент, вне зависимости от того, сколько дочерних элементов есть на самом деле

Пример,

h1:first-child - выберет первый тег h1 на всех уровнях вложенности



Дочерние элементы

:last-child – позволяет выбрать и форматировать последний дочерний элемент

:nth-child – групповой псевдоэлемент; например, с его помощью можно выбрать каждую вторую строку в таблице или каждый второй элемент списка: odd – нечетные, even – четные

tr:nth-child(**odd**) {background-color: #d9f0ff;}

tr:nth-child(**3n**) {background-color: #d9f0ff;} – каждый 3-й

tr:nth-child(**3n+2**) {background-color: #d9f0ff;} – каждый 3-й, начиная со второго

Дочерние элементы

Селекторы типов дочерних элементов

:first-of-type

:last-of-type

:nth-of-type

ЗАДАНИЕ: изучите особенности данных селекторов самостоятельно

Смежные элементы

Иногда есть необходимость выбрать тег, относящийся к группе смежных элементов одного уровня и имеющих общего родителя – **смежный элемент того же уровня**

Селектор смежных элементов использует знак + для соединения одного элемента с другим

h2 + p – выберет все первые абзацы, следующие за любым h2

ЗАДАНИЕ: особенности применения изучите самостоятельно

:not()

Селектор :not() или псевдокласс отрицания

Пример

p:not(.class) {color: blue;} – задаст тексту синий цвет во всех абзацах, к которым не применялся класс .class

a[href^="http://"]:not([href*="mysite.com"]) – позволит выбрать все ссылки, начинающиеся http://, исключив те, что связаны с mysite.com

:not()

Правила использования

- можно использовать только с простыми селекторами – селекторы элементов, универсальный селектор, классы, ID, псевдоклассы
- нельзя использовать селекторы-потомки (div p a), псевдоэлементы (::first-line), групповые селекторы или комбинации
- нельзя в одной строке применять несколько :not()-селекторов

:not()

Пример

.footnote:not(div)

img:not(.portrait)

div:not(#banner)

Спасибо за внимание