**Модуль 8**. **Каскадные таблицы стилей. Позиционирование.**

**Позиционирование.**

Позиционированием называют определение конкретного месторасположения на странице того или иного элемента (бокса). Позиционирование бывает абсолютным, относительным, фиксированным и статическим.

Для того, чтобы позиционировать какой-либо элемент к нему применяют свойство **position** и одно из его возможных значений:

* absolute - Абсолютное позиционирование элемента.
* relative - Относительное позиционирование элемента.
* fixed - Фиксированное позиционирование элемента.
* static - Статическое позиционирование элемента. (Элемент отображаются как обычно.)
* inherit - Наследует значение элемента родителя.

Абсолютное позиционирование.

Абсолютно позиционированный элемент **(position: absolute)** выводится из общего потока и несмотря на другие элементы и их взаимное расположение, занимает указанное место на странице от края/краёв окна браузера. При таком способе позиционирования один элемент может накладываться поверх другого.

Для того чтобы позиционировать элемент от края/краёв окна браузера, нам понадобятся следующие свойства CSS:

* bottom - Расстояние от нижнего края окна браузера.
* left - Расстояние от левого края окна браузера.
* right - Расстояние от правого края окна браузера.
* top - Расстояние от верхнего края окна браузера.

Данные расстояния могут быть заданы в пикселях, процентах или любых других принятых единицах измерения CSS, значение по умолчанию - auto.

***Примечание (ранние версии браузера Internet Explorer игнорируют данное свойство и элемент выводится на странице так как будто его вовсе не позиционировали.)***

Пример в файле absolute.html

Относительное позиционирование.

Относительное позиционирование **(position: relative)** определяет место элемента относительно краёв элемента родителя и не выводится из общего потока.

Также, как и в случае с абсолютным позиционированием расстояния от края/краёв родительского элемента задаётся с помощью свойств: bottom, left, right, top.

Пример в файле relative.html

Фиксированное позиционирование.

Фиксированное позиционирование **(position: fixed)** похоже на абсолютное, элемент выводится из общего потока, его координаты рассчитываются от краёв окна браузера, но при прокрутке страницы элемент не меняет своего положения.

***Примечание (ранние версии браузера Internet Explorer игнорируют данное свойство и элемент выводится на странице так как будто его вовсе не позиционировали.)***

Пример в файле fixed.html

z-index

Как уже говорилось, выше позиционированные элементы могут накладываться один поверх другого, имитируя тем самым некую трёхмерность страницы, где каждый последующий наложенный друг на друга элемент является слоем.

Свойство **z-index** позволяет веб-мастеру управлять позициями этих слоёв в глубину экрана (по оси Z.), другими словами позволяет указывать браузеру какие элементы следует показывать на переднем плане, а какие на заднем.

Значения свойства **z-index**:

* auto - Элементы накладываются друг на друга в том порядке, каком они указаны в коде HTML. (по умолчанию).
* целое число - Чем выше данное значение, тем более высокую позицию занимает элемент по отношению к тем элементам, значение которых ниже.

Числовое значение **z-index** может быть и отрицательным, однако не все браузеры правильно интерпретируют отрицательные значения.

Еще следует отметить, что при равном значении **z-index**, на переднем плане находится тот элемент, который в коде HTML идет ниже остальных.

Это же правило действует при **z-index** равным **auto** или же в том случае если бы данное свойство и вовсе не применялось к тем или иным элементам.

Пример в файле z-index.html

**Вид маркера в списке.**

Если Вы помните, то в чистом HTML вид маркера в списке определял атрибут **type** и одно из его возможных значений, в CSS данную задачу берёт на себя свойство: **list-style-type** которое, в свою очередь, тоже имеет свои стандартные значения определяющие вид маркера как всего списка сразу, так и его отдельного "пункта".

Значения **list-style-type**:

* disc - Диск. (по умолчанию для <ul>)
* circle - Полый круг.
* square - Квадрат.
* decimal - Арабские цифры. (по умолчанию для <ol>)
* lower-roman - Строчные римские цифры.
* lower-alpha - Строчные буквы.
* upper-roman - Заглавные римские цифры.
* upper-alpha - Заглавные буквы.
* none - Маркер отсутствует.

Пользовательский маркер рисунок.

Наиболее интересным CSS инструментом для работы со стилем списка является, возможность вместо стандартных маркеров описанных выше использовать свои собственные нестандартные изображения - небольшие рисунки, вписывающиеся в общий дизайн Вашего сайта.

Эту задачу выполняет свойство **list-style-image** которое определяет в качестве маркера списка некое графическое изображение с указанием пути к нему.

Значений данного свойства всего два:

* none - Отменяет графическое изображение маркера.
* url - Путь к файлу с рисунком маркера.

Путь к рисунку после url указывается в круглых скобках.

Вот так:

**list-style-image: url(graphics/marker.gif)**

Стиль обтекания маркера списком.

Свойство **list-style-position** указывает браузеру на то, как следует отображать текст в списке относительно его маркеров. По умолчанию маркеры находятся в стороне от текста списка, но можно сделать так, что они будут обтекаться текстом.

Возможных значений свойства **list-style-position** всего два:

* outside - Маркер находится в стороне от списка.(по умолчанию)
* inside - Маркер обтекается текстом.

Пример в файле list-style.html

**Псевдоклассы.**

Псевдоклассы - это особые свойства, которые позволяют менять стиль элемента в зависимости от действий пользователя, а так же положения этого элемента (тега) в общем потоке документа, что позволяет разбавить дизайн страницы некой динамикой и логикой. Классическим примером применения псевдоклассов является ссылка, которая меняет свой цвет при наведении на неё курсором.

Cписок псевдоклассов:

* hover - Стиль элемента на который наведён курсор мыши.
* active - Стиль для ссылки которая становится активной, но переход по ней еще не совершен.
* visited - Стиль для недавно посещённой ссылки.
* link - Стиль для нечасто посещаемой ссылки.
* focus - Стиль элемента находящегося в фокусе.
* first-child - Стиль первого дочернего элемента.
* last-child – Стиль последнего дочернего элемента
* nth-child() - используется для добавления стиля к элементам на основе нумерации в дереве элементов

Псевдокласс focus.

Данный псевдокласс определяет стиль элемента, если тот находится в фокусе. Теперь собственно о том, что такое фокус?.. как в случае с оптикой и иллюзией, слово фокус в CSS обозначат, что некий объект, а точнее элемент, находится в центре внимания наблюдателя - пользователя. Такими элементами могут быть теги <a> <input> <select> и <textarea>.

Пример в файле focus.html

Псевдокласс first-child и last-child.

Все элементы можно определить, как родительские или дочерние и что элемент родитель может содержать в себе несколько дочерних элементов, ну например:

<div>- блок родитель

<p></p> - первый дочерний элемент

<p></p> - второй дочерний элемент

<p></p> - последний дочерний элемент

</div>

Пример в файле first-child.html

Псевдокласс nth-child()

Значения:

* odd - Все нечетные номера элементов.
* even - Все четные номера элементов.
* число - Порядковый номер дочернего элемента относительно своего родителя. Нумерация начинается с 1, это будет первый элемент в списке.
* Выражение - Задается в виде an+b, где a и b целые числа, а n — счетчик, который автоматически принимает значение 0, 1, 2...

Если a равно нулю, то оно не пишется и запись сокращается до b. Если b равно нулю, то оно также не указывается и выражение записывается в форме an. a и b могут быть отрицательными числами, в этом случае знак плюс меняется на минус, например: 5n-1.

За счет использования отрицательных значений a и b некоторые результаты могут также получиться отрицательными или равными нулю. Однако на элементы оказывают влияние только положительные значения из-за того, что нумерация элементов начинается с 1.

В табл. 1 приведены некоторые возможные выражения и ключевые слова, а также указано, какие номера элементов будут задействованы.

|  |  |  |
| --- | --- | --- |
| Значение | Номера элементов | Описание |
| 1 | 1 | Первый элемент, является синонимом псевдокласса :first-child. |
| 5 | 5 | Пятый элемент. |
| 2n | 2, 4, 6, 8, 10 | Все четные элементы, аналог значения even. |
| 2n+1 | 1, 3, 5, 7, 9 | Все нечетные элементы, аналог значения odd. |
| 3n+2 | 2, 5, 8, 11, 14 | — |
| -n+3 | 3, 2, 1 | — |
| 5n-2 | 3, 8, 13, 18, 23 | — |
| even | 2, 4, 6, 8, 10 | Все четные элементы. |
| odd | 1, 3, 5, 7, 9 | Все нечетные элементы. |

Пример в файле nth-child.html

**Псевдоэлементы.**

Псевдоэлементы - это особый вид свойств CSS, которые позволяют работать не над самим элементом, а над его отдельной частью.

Перечень всех псевдоэлеметов:

* :first-letter - Стиль первой буквы текстового блока
* :first-line - Стиль первой строки текстового блока
* :after - Добавляет содержимое после элемента.
* :before - Добавляет содержимое до элемента.
* ::selection - Стиль выделенного пользователем текста.

Стиль первой буквы.

Псевдоэлемент **first-letter** задаёт стиль первой буквы в каком-либо текстовом блоке, проще говоря, без особых усилий позволяет сделать "буквицу".

Стиль первой строки.

Псевдоэлемент **first-line** определяет стиль первой строки в текстовом блоке.

Пример в файле first-letter.html

Псевдоэлементы **after** и **before**

Псевдоэлементы **after** и **before** предназначены для "вставки" в страницу сайта контента который изначально неуказан в HTML документе. Вставляется содержание перед **(:before)** или после **(:after)** какого либо элемента с помощью свойства **content**, которое собственно и определяет содержимое для вставки.

Всё вместе пишется так:

p:after {content: "Конец, а кто слушал молодец!!"; }

Теперь после каждого параграфа будет добавляться надпись: "Конец, а кто слушал молодец!!"

Значением свойства content может быть:

* "текст" - собственно любой текст или символы.
* " " – пустой контент, когда текст нам не нужен
* "\0410" - юникод.
* url(путь)- адрес какого либо объекта.
* open-quote - открывающая кавычка.
* close-quote - закрывающая кавычка.
* no-open-quote - отменяет открывающую кавычку.
* no-close-quote - отменяет закрывающую кавычку.
* inherit - наследует значение элемента родителя.
* none - ничего не добавляется.
* normal - для псевдоэлементов before и after тоже самое что и none.

Пример в файле after-before.html

Псевдоэлемент **::selection**.

Псевдоэлемент **::selection** (я не опечатался, пишется именно с двумя двоеточиями) указывает на стиль выделенного пользователем текста.

Данный псевдоэлемент появился на свет только в спецификации CSS3 и к сожалению поддерживается не всеми браузерами так IE его полностью игнорирует, а браузер Firefox использует свой аналогичный псевдоэлемент ::-moz-selection.

Пример в файле selection.html