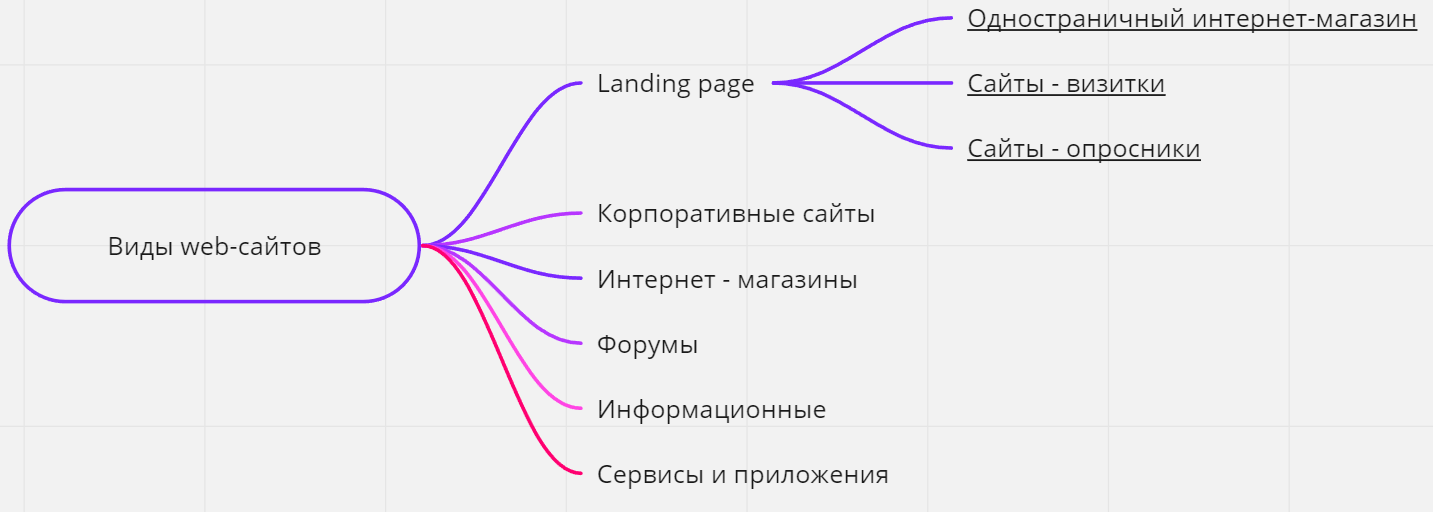
Классификация сайтов



Теги и атрибуты

Атрибуты бывают как специфические (для каждого тега свои), так и глобальные, которые можно применять к любому тегу.

<https://webref.ru/html/attr/common>

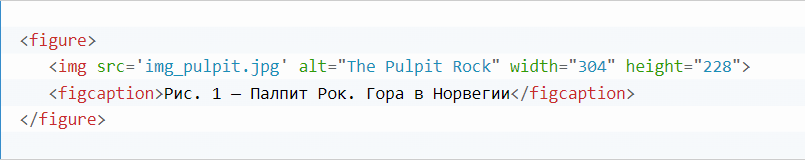
<https://developer.mozilla.org/ru/docs/Web/HTML/Global_attributes>

Ссылки на списки глобальных атрибутов

Семантические теги

Семантические элементы четко описывают, что они обозначают как браузеру, так и веб-разработчику. Их важно использовать в проекте, т.к. это повышает SEO-оптимизацию нашего сайта: наш сайт будет намного лучше ранжироваться поисковыми системами!

1. Section – определяет раздел в документе, обязательно должен содержать в себе заголовки;
2. Header – предназначен для определения заголовочного блока или «шапки» документа или раздела. В одном документе разрешается определять несколько элементов header, например, «шапка» для статьи;
3. Nav – определяет набор ссылок навигации. НЕ ВСЕ ссылки в документе следует размещать внутри элемента nav. Элемент nav предназначен только для основного блока навигационных ссылок!
4. Article – определяет независимый, самодостаточный контент. Используется, например, при публикации на форуме, публикация в блоге, газетная статья;
5. Aside – определяет некий контент, находящийся в стороне от контента, внутри которого он расположен (как боковой блок страницы «сайдбар»). Контент внутри должен соотноситься с окружающим контентом.
6. Main – определяет основное содержимое документа. Содержимое внутри элемента должно быть уникальным для документа, здесь не должно быть ничего, что повторяется где-либо, вроде, сайдбаров, навигационных ссылок, инфы об авторских правах, логотипов сайта, поисковых форм и т.п. !В документе должен быть только один элемент main, и он не должен быть вложен в элементы article, aside, footer, header, nav;
7. Address –
8. Footer – предназначен для определения «подвала» документа или раздела. Обычно в «подвале» размещают информацию об авторе документа, ссылки на условия использования текста, информацию об авторских правах, контактные данные и т.п. В одном документе разрешается определять несколько элементов footer;
9. Time – определяет дату/время, также может использоваться для кодирования даты/времени в машинопонятной форме, чтобы пользовательские программы-агенты могли добавлять эту информацию в системы напоминания или в расписание событий пользовательского календаря;
10. Wbr – указывает браузеру, где можно сделать перенос строки в тексте, если этого требует ширина родительского элемента;
11. Details – определяет доп.инфу, которую пользователь может открывать и закрывать. Элемент может использоваться для создания интерактивного виджета. Внутри тега можно размещать любой контент. Содержимое должно быть скрыто до тех пор, пока не будет установлен атрибут open. Для определения видимого заголовка доп.информации в теге используется элемент summary (на заголовок можно нажимать, это откроет/закроет доп.инфу);
12. Summary – определяет заголовок для элемента details, на который пользователь может нажимать, чтобы открывать/закрывать доп.инфу;
13. Figcaption и Figure – назначение figcaption – добавление визуального пояснения к изображению. Изображение и пояснение к нему могут быть сгруппированы в элементу figure



1. Mark – определяет маркированный/выделенный текст;
2. Form – используется для создания HTML форм для ввода пользовательских данных;
3. Table – определяет HTML таблицу для представления табличных данных. HTML таблица состоит из элемента table и одного или более элементов tr(определяет строку таблицы), th(определяет верхний колонтитул/заголовок таблицы), td(определяет ячейку таблицы).

CSS-cелекторы:

**Основные виды селекторов:**

1. \* – любые элементы;
2. div – элементы с таким тегом;
3. #id – элемент с данным id;
4. .class – элементы с таким классом;
5. [name=“value”] – селекторы на атрибут;
6. :visited – “псевдоклассы”, остальные разные условия на элемент.

**Основные виды отношений между элементами:**

1. div p – элементы p, являющиеся потомками div;
2. div > p – только непосредственные потомки;
3. div ~ p – правые соседи: все p на том же уровне вложенности, которые идут после div;
4. div + p – первый правый сосед: p на том же уровне вложенности, который идёт сразу после div(если есть).

**Фильтр по месту среди соседей:**

* :first-child – первый потом своего родителя;
* :last-child – последний потомок своего родителя;
* :only-child – единственный потомок своего родителя, соседних элементов нет;
* :nth-child(a) – потомок номер a своего родителя, например, :nth-child(2) – второй потомок. Нумерация начинается с 1;
* :nth-child(an + b) – расширение предыдущего селектора через указание номера потомка формулой, где a, b – константы, а под n подразумевается любое целое число. Этот псевдокласс будет фильтровать все элементы, которые будут попадать под формулу при каком-либо n. Например, :nth-child(2n) даст элементы номер 2, 4, 6…

:nth-child(2n + 1) даст элементы номер 1, 3…

:nth-child(3n + 2) даст элементы номер 2, 5, 8…

**Фильтр по месту среди соседей с тем же тегом**

Есть аналогичные псевдоклассы, которые учитывают не всех соседей, а только с тем же тегом:

* :first-of-type
* :last-of-type
* :only-of-type
* :nth-of-type
* :nth-last-of-child

**Селекторы атрибутов**

На атрибут целиком:

* [attr] – атрибут установлен
* [attr=“val”] – атрибут равен val

На начало атрибута:

* [attr^=“val”] – атрибут начинается с val, например “value”
* [attr|=“val”] – атрибут равен val или начинается с val- , например равен “val-1”

На содержание:

* [attr\*=“val”] – атрибут содержит подстроку val, например “myvalue”
* [attr~=“val”] – атрибут содержит val как одно из значений через пробел. Например: [attr~=“delete”] верно для “edit delete” и неверно для “undelete” или “no-delete”

На конец атрибута:

* [attr$=“val”] – атрибут заканчивается на val, например, равен “myval”

**Другие псевдоклассы:**

* :not(селектор) – все, кроме подходящих под селектор
* :focus – в фокусе
* :hover – под мышью
* :empty – без детей (даже без текстовых)
* :checked, :disabled, :enabled – состояние INPUT
* :target – этот фильтр сработает для элемента, ID которого совпадает с анкором #... текущего URL. Например, если на странице есть элемент с id="intro", то правило :target { color: red } подсветит его в том случае, если текущий URL имеет вид <http://...#intro>.

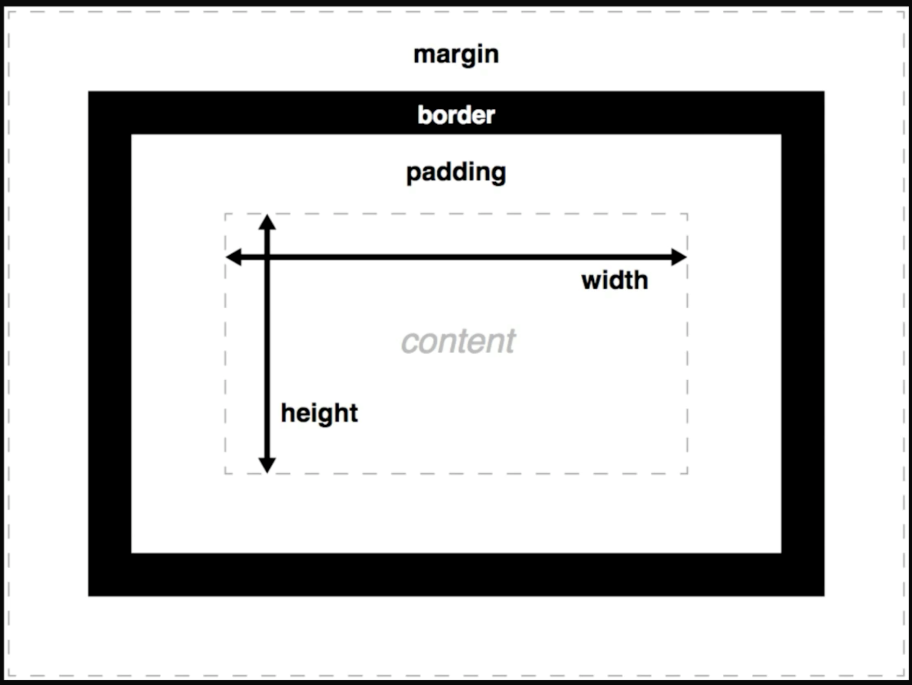
**Псевдоэлементы ::before, ::after**

«Псевдоэлементы» – различные вспомогательные элементы, которые браузер записывает или может записать в документ.

При помощи псевдоэлементов ::before и ::after можно добавлять содержимое в начало и конец элемента

Блочная система элементов

Все элементы, которые мы видим на странице, имеют прямоугольную форму!!! У любого прямоугольника есть две главные величины: ширина и высота.



Все элементы делятся на три вида по своей структуре:

* Block – формируют наш каркас на странице, т.е. разделяют страницу на какие-то области, это скелет нашего сайта. Блочные элементы очень жадные, они захватывают всю данную им область. Блочные элементы располагаются друг под другом.
* Inline – занимаются наполнением сайта. Данные элементы занимают ровно столько места, сколько контента у них внутри. Располагаются горизонтально. Строчным элементам нельзя задать ширину и высоту.
* Inline-block – комбинация строчных и блочных элементов. От блочных элементов взяли все параметры, которые есть в блочной модели css (можно задавать ширину, высоту, применять margin, padding, border), а от строчных элементов они взяли горизонтальное расположение, данные элементы не такие жадные, как блочные.

Важно при формировании структуры сайта использовать свойство box-sixing: border-box; Таким образом, изменяется алгоритм расчёта ширины и высоты элементов, теперь ширина и высота блока не будет складываться с margin, padding, border и будет иметь чётко заданные нами параметры.

Позиционирование элементов

Позиционирование позволяет вам изымать элементы из нормального потока макета документа и заставить их вести себя по-другому; например, располагаться друг на друге или всегда оставаться на одном и том же месте внутри окна просмотра браузера. Для этого используется свойство position.

По умолчанию, у свойства position стоит static, то есть элемент отображается в потоке так, как нужно.

* Relative – теперь можно располагать элемент так, как нам вздумается для этого используем значения css-координат: top, right, bottom, left.
* Absolute – позволяет любой абсолютно элемент располагать в том места, где нам хочется. При абсолютном позиционировании элемент уже выпадает из списка HTML, находится не в потоке макета документа, становится не по порядку. Располагается относительно своего ближайшего родителя. Родителю важно задать position: relative;
* Fixed – «липкое» меню, которое будет оставаться неизменным в положении при скролле страницы
* Inherit – наследует свойство от родителя
* Sticky – это гибрид абсолютной и фиксированной позиции, который позволяет позиционируемому элементу вести себя как будто он абсолютно позиционирован, до тех пор пока он не будет прокручен до определённой пороговой точки (например, 10px от вершины окна просмотра), после чего он становится фиксированным. Это можно использовать, например, чтобы заставить панель навигации прокручиваться вместе со страницей до определённой точки, а затем задерживать в верхней части страницы.

**Принцип карточной колоды:**

Для того, чтобы расположить элементы на оси Z, мы используем свойство z-index. Чем больше значение данного свойства, тем выше вероятность, что элемент будет к пользователю ближе (будет на переднем плане). НО! Если элементы различаются по позиционированию, то чтобы сместить спозиционированный элемент по оси Z, свойству z-index нужно задать отрицательное значение.

Свойство Float

Свойство float CSS указывает, что элемент должен быть взят из нормального потока и помещён вдоль левой или правой стороны его контейнера, где текст и встроенные элементы будут обтекать его (Текст и Inline-элементы). Элементы при наличии float автоматически получает display:block, кроме display: inline-table; run-in;

Float: left; right; none; inherit;

Плавающий элемент - это любой элемент со свойством float отличным от none.

Единицы измерения CSS

Единицы измерения подразделяются на относительные и абсолютные, разница в том, что относительные величины зависят от какого-то другого параметра.

* Относительные единицы:

1. em – высота шрифта текущего элемента
2. ex – высота символа x
3. px – пиксель(мельчайшая точка на экране) зависит от размера и разрешения экрана
4. % – процент (нельзя использовать в свойстве height, если высота родительского элемента не назначена)
5. rem – размер шрифта корневого элемента в документе
6. vw – ширина доступного нам размера экрана (1/100 ширины экрана)
7. vh – высота размера экрана (1/100 высоты экрана)
8. vmin, vmax – определяются минимальным или максимальным значением высоты или ширины. Например, если ширина окна браузера равна 1200px, а высота 700px, то vmin будет равен 7px, а vmax — 12px.

* Абсолютные величины:

1. In – дюйм (1 дюйм = 2,54 см)
2. cm – сантиметр
3. mm – миллиметр
4. pt – пункт (1 пункт = 1/72 дюйма)
5. pc – пика (1 пика = 12 пунктам)

Полезные ресурсы:

1. <https://stackoverflow.com/> - форум программистов

<https://ru.stackoverflow.com/> - российский

1. <https://www.markupeasy.ru/> - сервис для работы с Psd-макетами (аналог AvoCode)
2. <https://docs.emmet.io/cheat-sheet/> - сокращения Emmet
3. <https://convertio.co/ru/> - конвертер файлов
4. <https://webref.ru/html> - справочник HTML
5. <https://htmlreference.io/> - справочник HTML
6. <https://nikomedvedev.ru/other/vscodeshortcuts/hotkeys.html> - Горячие клавиши VS-code
7. <https://msiter.ru/> - самоучитель и справочники по HTML, CSS, XML
8. <https://cssreference.io/> - справочник css
9. <https://sitehere.ru/knigi-po-sajtostroeniyu> - книги по сайтостроению
10. <https://sitehere.ru/tablica-stilej-css-v-udobnom-vide> - таблица стилей CSS со шпаргалками