Появился **html** в 1994 с версии 2.0. В настоящее время последняя версия **html** 5.3 и состоит язык из открывающих и закрывающих тегов

Раздел **<head>** содержит в основном техническую информацию

**Семантическое ядро** — встречается при настройке рекламы, а именно — отображения сайта в топах по запросам

Тег **<title>** — должен содержать не более 60 символов

Тег **<body>** - основная часть документа

В html версии 5 если нет тега **<body>** содержимое будет отображаться, но гарантий нет

Тег **<div>** — контейнер, универсальный блочный элемент и используется для блочной верстки. Она позиционируется абсолютно (крайний верхний левый угол страницы, что структурно для каждого элемента блочного типа)

Основной тег таблицы — **<table>**, **<tr>** — строка, **<td>** - ячейка (**TR - table row**, строка таблицы, **TD - table data**, данные таблицы)

**border** — параметр границ таблицы (можно прописать внутри тега **<table>** (как и внутри многих других тегов)). Длину проще подключать в процентах для удобства использования и отображения в различных браузерах. Для закрепления длинны ячеек её нужно записать внутри тега **<td>** — **width = «50%»**. Процент должен быть целым числом, а иначе браузер все равно преобразует его в целое число. Тег **<br>** - перенос текста на новую строку. Не требует закрывающего тега и вставляется в то место где нужен перенос

Высоту любого объекта лучше задавать автоматическую (**height = «auto»**)

Тег **<p>**- текст блочного типа. По умолчанию использует Times New Roman.

Тег **<h1>** — заглавный текст 1 типа (как заголовок в ворде)

Для задания шрифта текста используется параметр **font**= «…» и браузер будет брать первый, который сможет взять из списка указанных в коде

Есть кодировки «безопасных» цветов. Для их использования обязательно нужно использовать символ решетки. Можно и через английское слово (blue, green и т.д)

Для изменения текста внутри кода **html** используется тег **<font>**. Проценты размера текста будут зависеть от размера пикселя от шрифта. Но в документации VS Code говорится, что скоро тег не будет более использоваться(

Для вставки картинок используется тег **<img src=“/.../...“>** . Можно не использовать src, но тогда файл должен находиться с кодом в одной директории

Для позиционирования по горизонтали тег **align**, **valign** — для позиционирования по вертикали

**<!--...-->** - для задания комментариев

**colspan** — объединение по горизонтали

**rowspan** — объединение по вертикали

И использовались эти методы во времена **html** 2.0 и в наше время не применяется

Значок сайта во вкладке загружается через тег **<img>** внутри **<head>** с размерами 16 на 16 пикселем и форматом **.ico**

Любой **тег** — это объект со своими параметрами (например высота, ширина и т.д).

Листы стилей (или файл **.CSS**) нужны для применения на огромном количестве страниц сайта одинаковые стили

**<iframe>** — позволяет открывать внутри определенного места на странице другие страницы или видео/пдф файлы. Как пример в гугл картах можно выбрать интересное место, выделить код и встроить этот код себе на страничку и там-же настроить её

Для подключения **CSS** файла нужно создать файл с расширением **.CSS** и поместить в одну директорию с файлом **html**, а далее в теге **<head>** нужно разместить следующий код:

**<link rel = «stylesheet» type = «text.css» href = «main.css»>**

Все классы в **CSS** начинаются с . но в **html** указываются уже без неё. Названия у классов могут быть любыми, но желательно их делать логически понятными

В **CSS** для комментарием используется /\* следующая штука \*/

В **CSS** для контура используется **border: solid 1px**, где **solid** — тип контура, а 1px размер контура

**border-right-color:black;** - для задания цвета контура можно и без параметра **right**)

У атрибута **border** огромное количество параметров. Можно открыть **VS Code** и почитать, что вам предложит **intellisense**

**border-radius: 15px;** - это радиус контура по углам

ПС, если поставить его 100% - то скорее всего у вас получится овал из-за изначальной «квадратности» объектов

Для указания цвета фона используется **background-color: blue;**

Для указания стиля шрифта используется **font**-… и можно также посмотреть что предложит **VS Code**

**transition** — можно использовать для «замедления» отображения либо чего-то подобного

**a:link{**

Общий стиль ссылок. Можно конечно и таким образом стилизовать все ссылки, а можно и стилизовать каждую ссылку. Например:

У нас уже есть **<div class = «example»>** и внутри есть какая-то ссылка. Для стилизации ссылок для конкретного **div**

**.example a** {

Стили ссылок именно для **example**

}

**}**

**a:visited** {

Стиль посещенной ссылки

}

**a:hover** {

При работе с наведением на ссылку. Сделает плавное появление ссылки при наведении

**transition: 1s;**

}

**a:active** **{**

Стиль активной ссылки

**}**

**a:before** {

Отображает то, что находится до какого-то элемента

**content: «...»;** - вывод некопируемого текста

}

**a:after** {

Отображает то, что находится после какого-то элемента

**content: «...»;** - вывод некопируемого текста

}

Для отображения цвета лучше всего использовать не «безопасные» цвета, а **rgb(0.0.0);**

**<span>** можно использовать как заменитель других тегов (допустим когда другие закончились, а стилизовать текст еще как-то нужно)

**<div class = «...»>** нужен для стилизации определенного участка кода. Указанный в **div** класс далее можно будет стилизовать в CSS.

В фон можно встраивать не только цвет, но и картинку. В **css** это выглядит так **backgroud-image: url(…);** и по умолчанию **background** старается занять всю площадь объекта и для того, что картинки не дублировались можно поставить свойство **background-repeat:no-repeat;** можно и указать размер фона через **background-size:75%;** и еще куча других свойств, которые можно будет найти в **intellisense VS Code**

**float** — обтекание текста. Допустим **float:left;** - Элемент будет прижат к левому краю и весь текст будет обтекать его, а наследование будет идет от родительского элемента

**margin** — отступ между объектами от родительского элемента

**display:block;** нужен для указания «блочности» элемента и благодаря этому элементы, которые находятся рядом друг с другом так и будут находиться

Псевдокласс — по сути нужен для задания дополнительных параметров. Например **:hover, :before, :target**