# **HTML+CSS**

Полезные сайты:

Cubic-beizer (задаёт анимацию)

Voice Design (создаёт уникаль. голос)

Figma (фотошоп)

Фреймворки для создания сайтов:

Bootstrap

Брутфорс

Wordpress

cssgridgarden

                             HTML5

p\*3>lorem50

p>lorem50

<Теги></Теги>

<header>Заголовок</header>

<main>Структура сайта<main> (может быть только один на странице, он должен содержать основное содержание)

<footer>Нижняя часть сайта<footer>

<section>структурирует документ</section>

<nav>навигация</nav>

<noscript>выводит информацию, если у конечного пользователя отключён JavaScript</noscript>

<script>подключение файла JS</script>

<еm>Курсив</em> или <i></i>

<strike>перечеркивает</strike>

<del>перечеркивает текст</del>

<sub>ниже основного ряда</sub>

<sup>выше основного ряда</sup>

<br>(один) - перенос строки

<hr>(один) - горизонт линия

<abbr title="">подсказка</abbr>

<code>отображает код на стр</code>

<address>

по умолчанию курсив

</address>

<big>текст чуть больше</big>

<small>текст чуть меньше</small>

<pre>сохраняет все отступы<pre>

<ol>пронумированный список, внутри него <li>эл-нт списка</li> </ol>

<ul>непронумерованный список</ul>

<div>Просто<span>текст</span></div>

<a href = "ссылка" target = "\_blank" (новая вкладка)>текст для ссылки</a>

<а href = "#ссылка якорь">Вниз</a>

<a name = х"ссылка якорь"><а>

Scroll-behavior: - css свойство которое применяется к тегу html и применяет параметр auto (по умолчанию) и smooth (плавный скролл страницы к "ссылке якорь")

<a href = "mailto:(почта)">на почту</a>

<img src="URL﻿ картинки﻿" width="ширина px" heigt="высота px" alt="описание картинки">

<blockquote>цитата (отступ 40рх по умолчанию) </blockquote>

Подключение css файла в теге <head><link rel="stylesheet" href="(имя css файла)>

Подключение javascript файла прописывается в самом конце тега <body> <script scr="Полный путь к JS файлу">

</script>

<table border=(граница таблицы в px) width=(ширина рх) align="center">

   <thead>

      <tr>

         <th>Заголовок (жир шрифт) </th>﻿﻿

      </tr>

   </thead>

   <tbody>

      <tr> (Новая строка, пункт)

         <td>Описание</td>

      </tr>

   </tbody>

   <tfoot>

      <tr>

      </tr>

   </tfoot>

</table>

<form action="" method="post"> (сам тег ничего не создаёт, но служит родитель. тегом для создания форм)

<label for="указать имя с каким input будет работать тег label">Введите имя</label>

<input type="тип поля" placeholder="подсказка внутри поля" value="устанавливает текст по умолчанию внутри поля" maxlenght="максимальное кол-во символов" name="название поля" (для JS) id="для применения css стилей" hidden (скрывает поле) disabled (не активное поле) readonly (позволяет считать, но не записывать)>

<input type="text"; "password"; "email" "url"; "color (не везде поддерживается"; "date"; "time"; "datetime"; "search (поле поиска на сайте)"; "file"; "checkbox" (галочка для пользователя); "radio" (кружок для пользователя)>

Чтобы пользователь мог выбрать только один radio, нужно в input name="указать для всех radio одно имя"

<input type="button" (кнопка при которой не происходит перезагрузка страницы); "submit" (перезагружается страница и отправляются данные из тега form; "reset" (очищает данные из всех полей в теге form>

<тег АТРИБУТЫ = "параметры атр">

<тег title = "текст подсказка">

<ol start = "цифра с которой начинается список"  type = "тип пронумерованного списка">

<ul type = "circle" (пустые круги)  "square" (квадратики)>

Теги оформления списка dl, dt, dd

                       HTML+CSS

/\*комментарий \*/

<p class="название (N)"></p> (В CSS обращаемся к классу через точку)

.название (N){

   нужные нам свойства

}

<p id="N"></p> (id - индетифткатор 1id=1N) (N (моя пометка) - это любое ваше название класса или id) (обращение к id через #)

#N {

   нужные нам свойства

}

Свойства в CSS:

color: rgba( 4-ое значение задаёт прозрачность цвета от 0 до 1)

transparent - прозрачный текст

background-color: (цвет фона)

background: linear-gradient(1-й цвет, 2-й цвет. Создаёт перелив страницы)

background-image: URL ("путь") - позволяет сделать фон из картинки

background-size: contain (картинка в зоне видимости); cover (картинка на всю длинну страницы) npx npx (первое значение - указывает ширину, а второе - высоту картинки)

background-position: npx npx (первое значение - вправо, второе значение - вниз)

background-repeat: no-repeat (отменяет повторение картинки)

-webkit-background-clip: text (позволяет наложить фон на сам текст, работает в паре с color: transparent)

❗-webkit- может поддерживаться не во всех браузерах❗

text-decoration: (подчеркивание) underline; (снизу), overline; (сверху), line-through; (перечеркнут)

border-bottom: 1px dashed black; (задаёт пунктирное подчеркивание), solid (обычная линия)

text-align: justify; (выравнивание по ширине)

text-decoration-color: (цвет линии)

text-decoration-style: (стиль линии)

text-decoration-line: (расположение л)

cursor: (курсор при наведении на объект) url ('курсор в любую картинку') auto;

margin;(внешние отступы) margin-top; margin-bottom; margin-left; margin-right (Отступы)

padding: (внутренние отступы)

width: (ширина) px, vh (масштабируется в зависимости от окна браузера) , %; fit-content (принимает размеры контейнера)

height: (высота)

max-height:

max-width:

display: inline; (делает div в строку)

display:inline-block;(обрщается к блоку)

line-height: (расстояние между строк)

font-size: (размер шрифта)

font-weight: (насыщенность шрифта)

font-style: (стиль шрифта)

word-spacing: (расстояние между слов)

letter-spacing: (между буквами)

text-transform: lowercase, uppercase, capitalize(каждое слово с большой бук)

writing-mode: vertical(написание текста по вертикали, горизонтали)

word-wrap: (перенос текста внутри блока)

white-space: (обработка пробелов)

text-shadow: 1рх 1рх 1рх цвет (тень текста)

box-shadow: 1рх 1рх 1рх цвет (затемнение блока)

border: (оформление границы)

border-color:

outline - тоже граница

outline-offset: npx (внутренний отступ между границей и объектом)

border-radius: (закругляет углы, применяет 4 параметра на каждый угол)

ol > li::marker {

     font-weight: bold; (делает жирными нумерацию списка)  }

box-sizing:(поведение размеров) border-box; (сохраняет заданные размеры, рамки и отступы отрисовываются внутрь)

rotate(n deg) - поворачивает объект

opacity: 0-1 прозрачность

z-index(n) - задаёт индекс объекту. Чем выше число, тем приоритетнее объект. z-indeх используется в паре с position: absolute; fixed; sticky; relative (смещает объект при этом сохраняя место в сетке)

transition: (плавно изменяет значения других  CSS свойств)

transition: property (какие свойства применяют изменения) duration (время перехода) timing-function(кривая скорости) delay (задержка)

transform: (позволяет изменить размер, положение, ориентацию эл-в).

translate (50px 100px) - перемещает объект по оси Х и Y (translateX только по Х, translateY только по Y)

scale(n)/scale(n, n) - увеличивает или уменьшает объект

rotateX/Y (n deg) - поворачивает объект

skew (n deg, n deg) - искажает по Х и Y

perspective (500px) rotateY(45deg) - устанавливает перспективу и поворот оси Y

transform-origin: (n%, npx) определяет исходную точку для трансформации

animation: (1)name (имя анимации, которая определяется с помощью @keyframes) (2)duration (время анимации) (3)timing-function (устанавливает кривую скорость анимации:

steps(кол-во шагов)

linear (одинаковая)

ease (стандартное смягчение)

ease-in (ускоряется)

ease-out (замедляется)

ease-in-out (ускоряется и замедляется)

cubic-beizer(n, n, n, n) (кастом кривая)

(4)delay(задержка перед выполнением) (5)iteration-count (кол-во повторений, целое число или infinite); (6)direction (направление анимации):

normal - обычное направление

reverse - обратное

alternate - вперёд и обратно

alternate-reverse

(7)fill-mode - определяет, как стили анимации будут применять я до начала анимации и после её окончания:

none (нет стилей до и после)

forwards(сохраняются после окончания)

backwards (применяются до начала)

both (применяются до и после)

(8)play-state (состояние анимации running/paused)

filter: blur(radius) - размытие элемента. пример filter: blur(10px)

filter: brightness(factor) - яркость эл-та.

меньше 1 - темнее, больше 1 - ярче.

filter: contrast(factor) аналогично bright

filter: drop-shadow (1рх 1рх 1рх цвет)

filter: grayscale (factor) - серость

filter: hue-rotate(deg) изменяет оттенок цвета элемента deg - угол вращения.

filter: invert(factor)от 0 до 1 инвертирует

filter: opacity(factor)от 0 до 1 непрозрач

filter: sturate(factor) - насыщенность меньше 1 уменьшает, больше 1 увелич.

filter: sepia(factor) эффект серии 0-1

display:flex (имеет две оси)

align-content:

align-items: flex-start; flex-end; center; baseline (Эл-ты отображаются на базовой линии контейнера); stretch (Эл-ты растягиваются, чтобы заполнить контейнер)

flex-direction: (устанавливает главную ось) column (расположение в колонку; row( в строчку (по умолчанию)) row-reverse; column-reverse;

flex-flow: row wrap(совмещает flex-direction: и flex-wrap:

flex-wrap: wrap (переносит Эл-ты, которые не поместились в строку)

flex-basis: n% (растягивает ширину всех элементов на заданную ширину)

flex-shrink: (n) (во сколько раз бвстрее при уменьшение браузера, будет сжиматься объект)

flex-grow: (n) (сколько пространства будет занимать элемент на экране)

justify-content:(выравнивание по главной оси) center; flex-end; flex-start; space between (отодвигает Эл-ты на одинаковое расстояние между друг другом); space-around (одинаковые отступы со всех сторон)

align-items: (выравнивание по второстепенной оси) имеет те же параметры, что и justify-content.

gap: npx/n% (отступы)

align-self: (позволяет выравнивать отдельно взятый эл-т)

order: (n)-число (задаёт порядок эл-ов. По умолчанию order для всех элементов 1. Эл-т с самым большим числом уходит в конец контейнера)

order: +1/-1

ПСЕВДОКЛАССЫ:

.N:hover (hover псевдокласс, применяет стили к объекту при наведении на него) { }

.N:active (псевдокласс будет срабатывать каждый раз, когда мы будем нажимать на объект)

.N:visited (добавляет стили к ссылкам по которым мы уже переходили)

input:focus {срабатывает, когда мы навели мышку внутрь тестового поля}

input {color: blue}

input:focus (когда активно) {color: red}

input:disabled {стили будут применнены только к тем input, которые имеют атрибут disabled в HTML}

Input:enabled {стили применяются к активным input}

input:required {обязательное поле для заполнения}

p.text:empty (пустой) {

   Обращаемся к тегам p, которые содержат класс text:empty обращается к пустым абзацам

}

p.text:not (:emty) {к не пустым}

p:first-of-type {применяет стили только к первому абзацу}

ПСЕВДОЭЛЕМЕНТЫ:

При помощью них мы можем работать с определённой частью текста внутри HTML объекта

#id::after (после) {

   content: 'любой текст' добавляет текст после объекта (добавленный текст не выделяется) ::before (до) }

p.N::first-letter {обращается к первому символ абзаца}

::first-line {к первой линии абзаца}