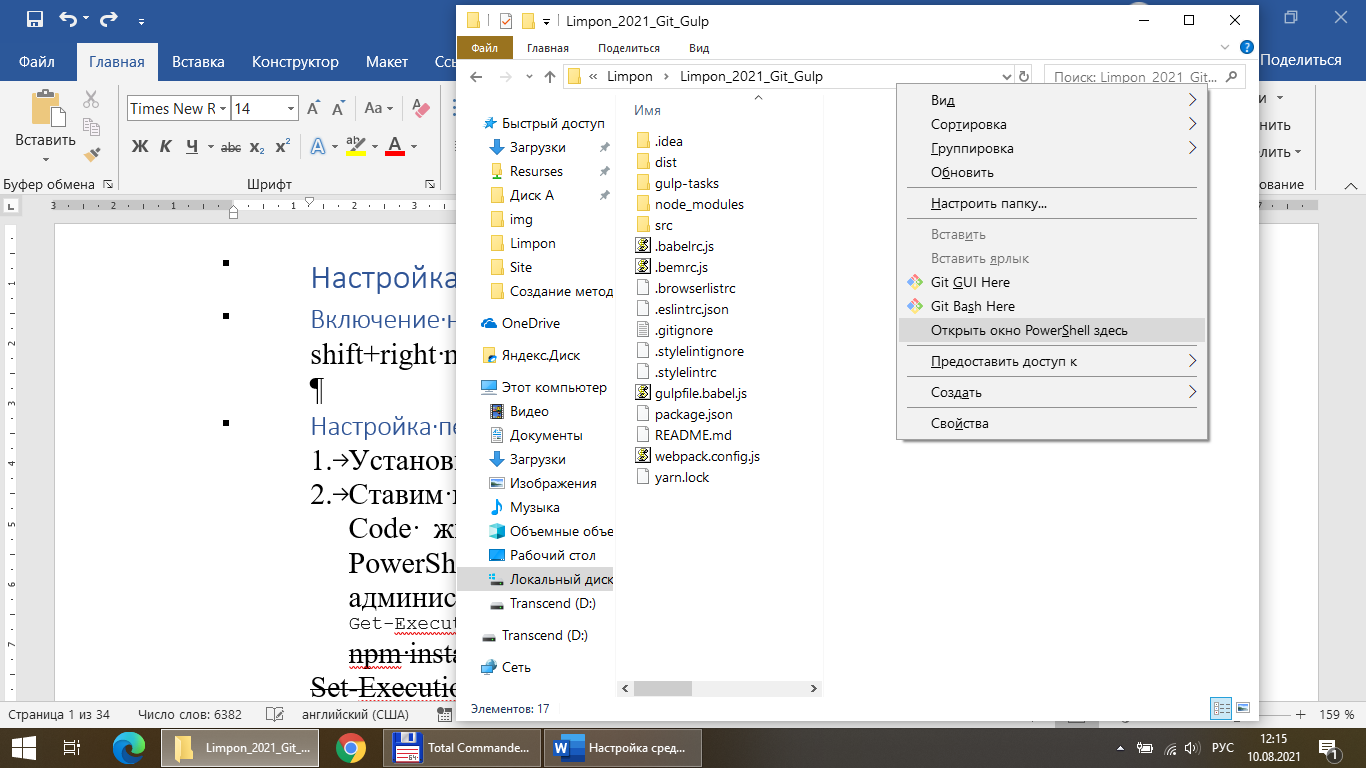
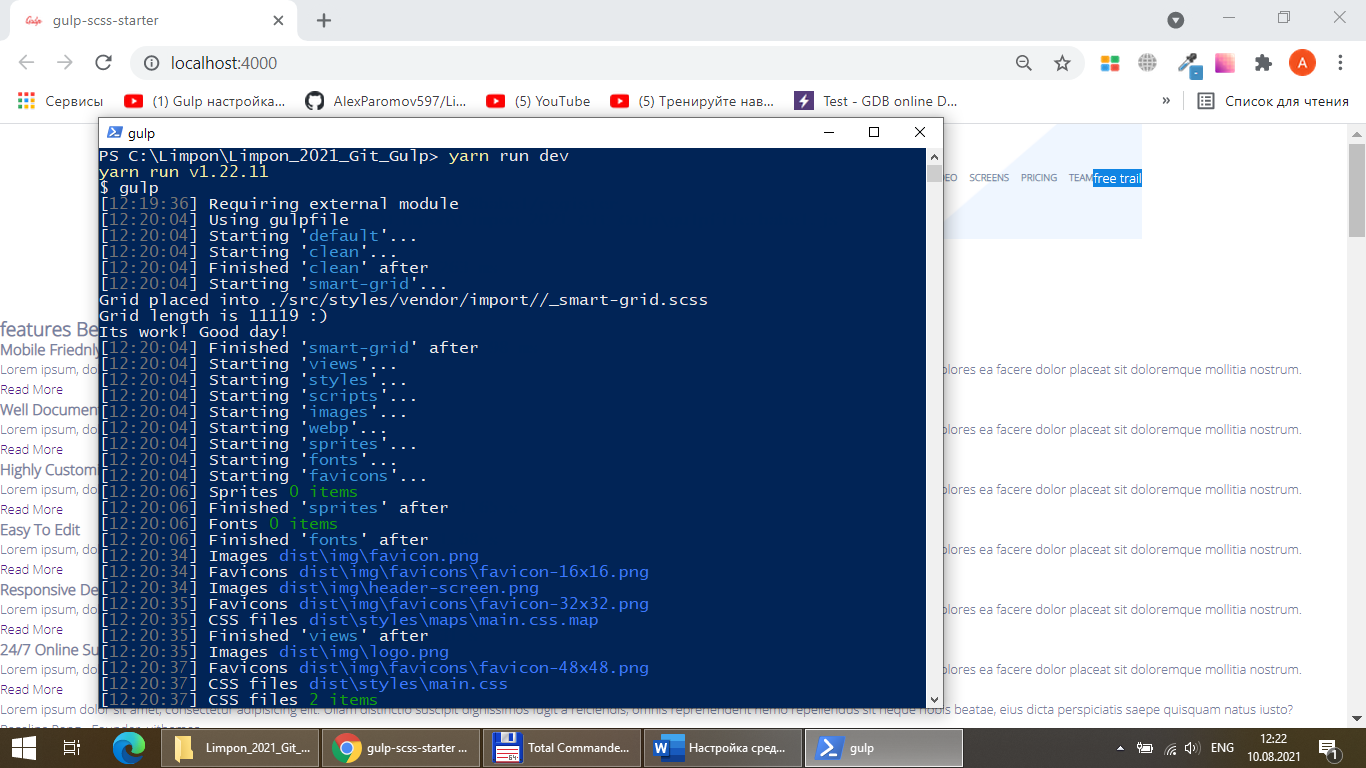
# Настройка среды разработки

## Включение на рабочий день

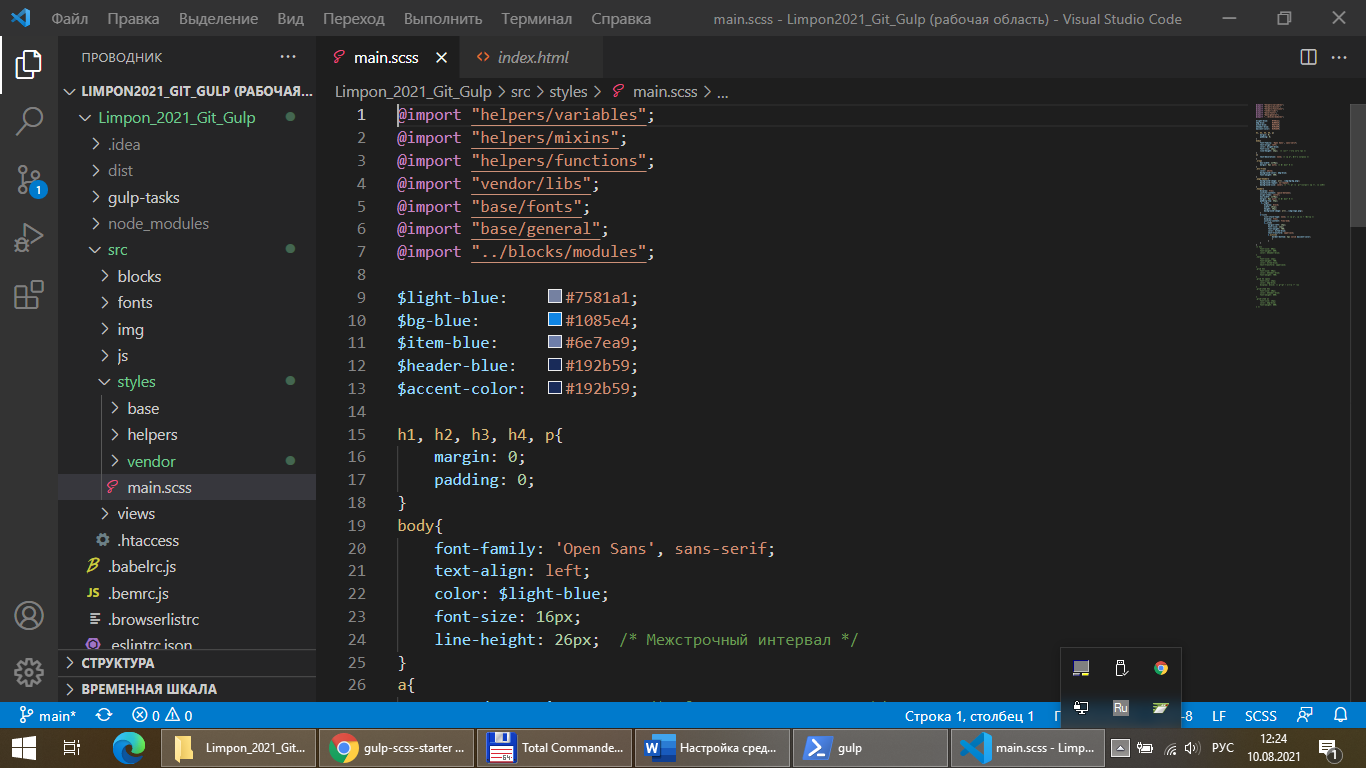
shift+right mouse открыть терминал PowerShell в папке проекта



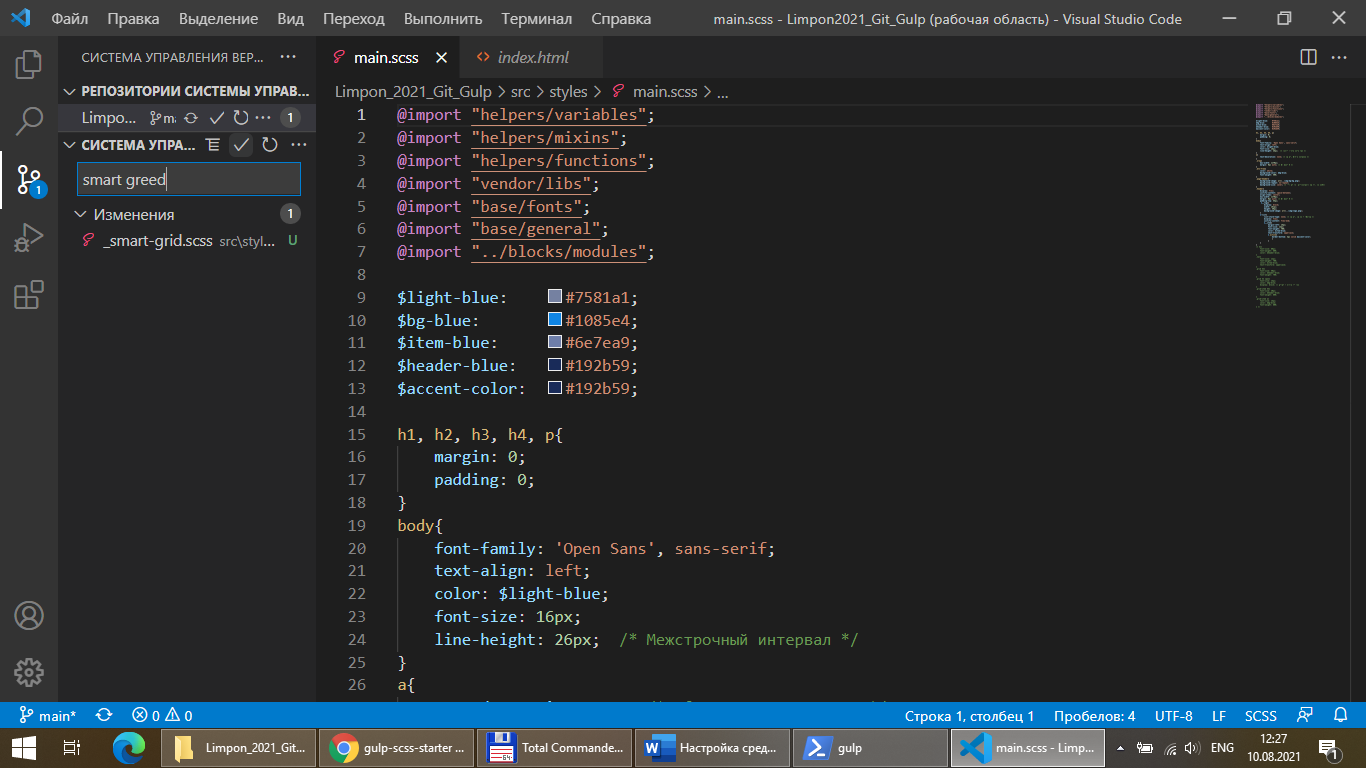
Запустить Gulp, и сервер браузера: yarn run dev



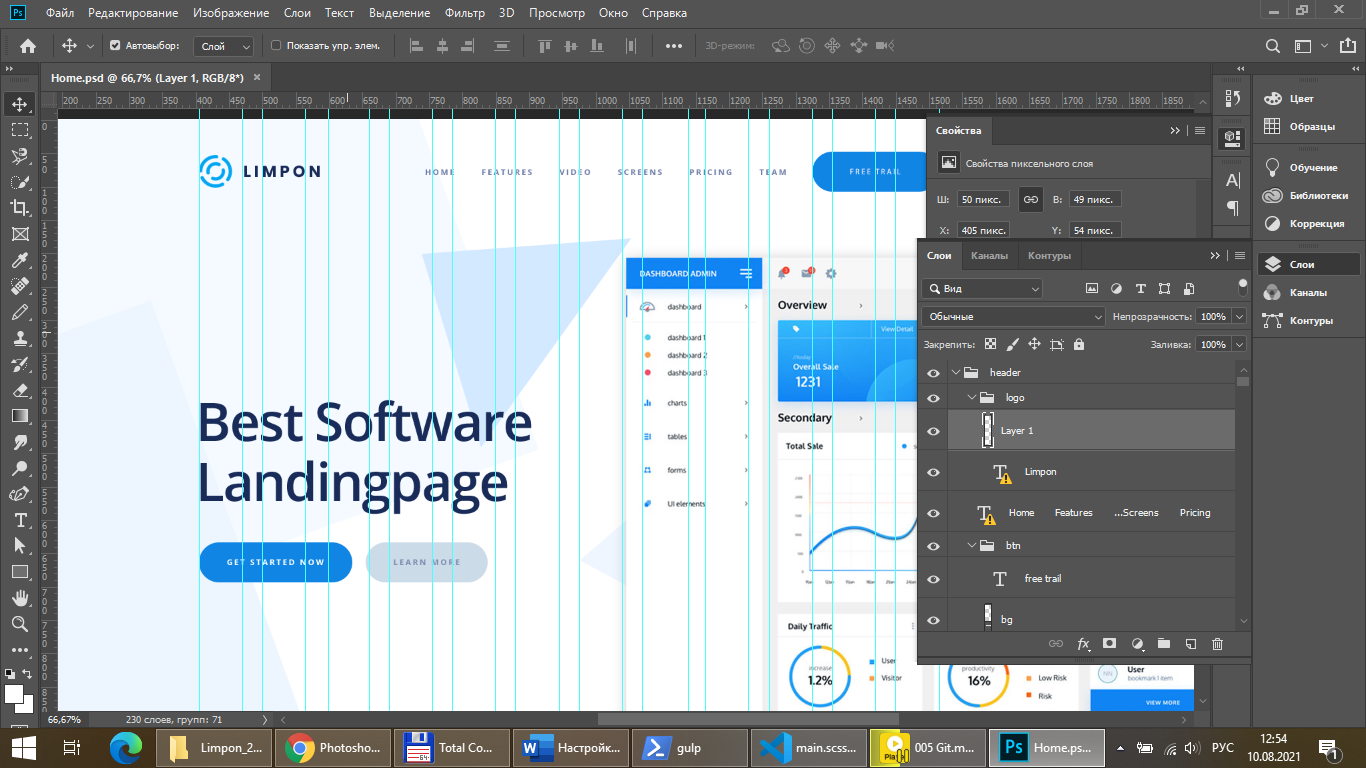
Запускаем VS Code



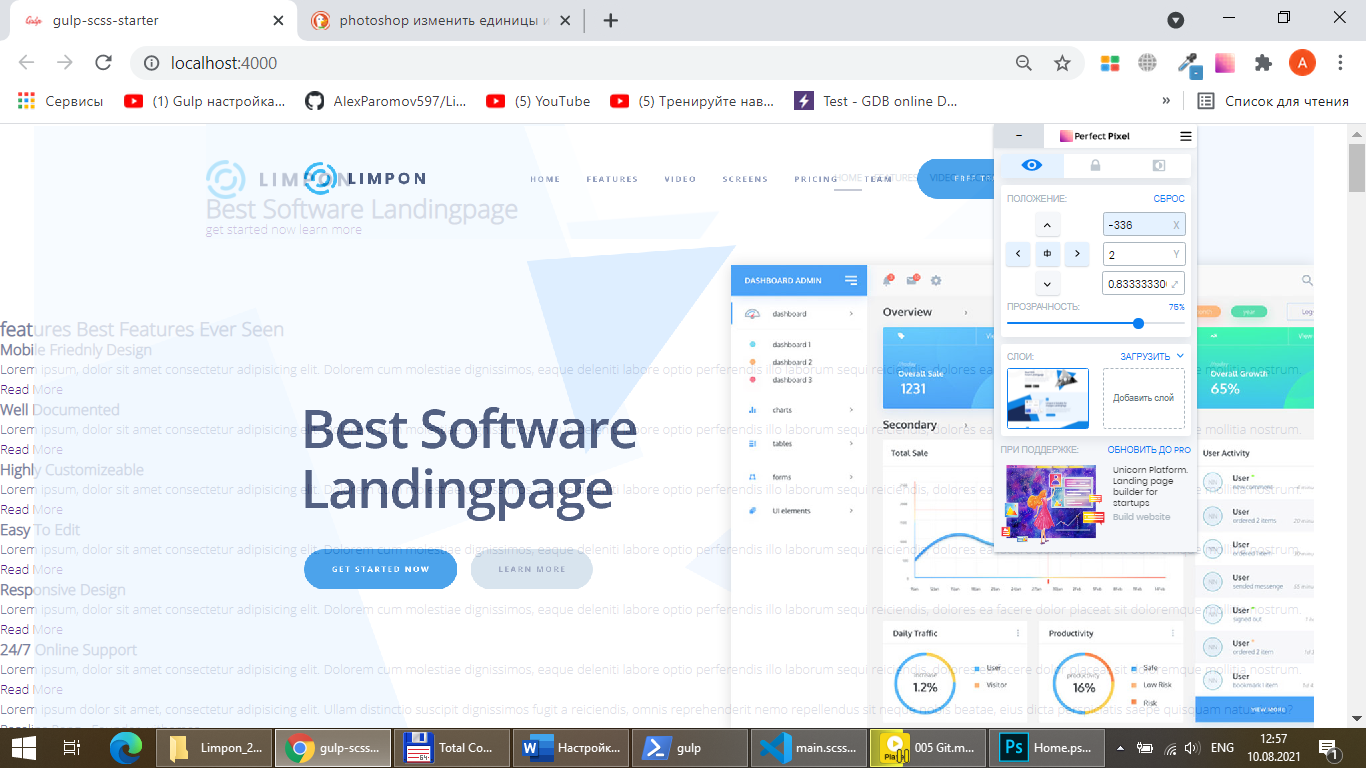
Если необходимо, Commit (Фиксация), Push (Отправка фиксаций)



Открываем Photoshop (Просмотр, Линейки, Пиксели)



Включаем пиксель PerfectPixel



## Настройка первый раз на компьютере

1. Установить node.js
2. Ставим глобально Gulp или Yarn (В проводнике на папке например, Code жмем shift+right mouse открыть терминал cmd, GitBash, PowerShell – Установить Set-ExecutionPolicy RemoteSigned с правами администратора)

Get-ExecutionPolicy -List

~~npm install --global gulp-cli~~

~~Set-ExecutionPolicy Unrestricted LocalMachine~~

cd c:\Limpon\Limpon\_2021

1. Создаем репозиторий, подключаем и синхронизируем его в декстопе его каталог, подключаем его через мастера в ВС коде, она запрашивает авторизацию вс кода. Закидываем файлы в каталог и пуш.

yarn --version

yarn

yarn run dev

## NPM

### Один раз на машине:

1. npm ls -g --depth=0 (empty)

2. npm i gulp-cli -g

3. npm ls -g --depth=0 gulp-cli@2.0.1

4. gulp Local gulp not found - error

/////////////////////////////////////////////////////////

npm run dev package.json

npm run build package.json

**Каталоги только на английском языке иначе работать не будет !!!**

npm i $name установить модуль локально, $name- имя модуля

npm i $name -g установить модуль глобально

npm ls список модулей, установленных локально

npm ls -g список модулей, установленных глобально

npm ls -g --depth=0 список модулей, установленных локально

npm r $name удалить модуль локально

npm r $name -g удалить модуль глобально

npm init -y создать package.json пустой стандартный

/////////////////////////////////////////////////////////

### Для старта каждого проекта:

1. cd [/d] $path

2. вставить gulpfile, package, package-lock

3. npm i

npm i gulp -D !!!!!!

///// 2 типа модулей 1 для разработки 2-ой в продакшн попадать в devDependencies

"dependencies": {

"normalize.css": "^8.0.1" используются только в финальном коде

},

////// --save-dev или -D

"devDependencies": {

"browser-sync": "^2.26.3",

"del": "^3.0.0",

"gulp": "^4.0.0",

"gulp-autoprefixer": "^6.0.0",

"gulp-clean-css": "^4.0.0",

"gulp-concat": "^2.6.1",

"gulp-if": "^2.0.2",

"gulp-sourcemaps": "^2.6.4"

}

npm config get cache и удалить папку ручками

## Git Bash /////////////////////////////////////////////////////////

входит в дистрибутив Git for Windows (msysGit).

Дистрибутив Git для Windows называется msysGit или же просто Git for Windows.

Это один и тот же дистрибутив. Чтобы ещё больше запутать ситуацию, для его сделали 2 разных сайта:

http://git-scm.com/download/win 160

https://msysgit.github.io/ 42

Это один и тот же дистрибутив.

cd c:\Users\Паромов\Documents\GitHub\BukeeSRC

cd /d/\_3/ переход d:/\_3/

cd /d d:\\_3

.. - переместиться из текущего каталога на одну папку вверх (из текущего подкаталога в родительский каталог)

Node.js с сайта

node -v

npm -v

### Gulp с сайта https://gulpjs.com/

(sudo) npm install gulp-cli -g (sudo) с правами администратора

npm install gulp -D

npx -p touch nodetouch gulpfile.js

gulp --help

gulp -v

npm install --save-dev gulp установили Gulp

https://gulpjs.com/plugins/

### Emmet https://docs.emmet.io/ - док по emmet

ul>li\*4 <tab>

.fsfddsf <tab> <div class="fsfddsf"></div>

bgc <tab> background-color: #fff;

ul>li\*5

!

link <link rel="stylesheet" href="">

.page <div class="page"></div>

a.btn <a href="" class="btn"></a>

m-

p10

bgc

fc

fw

# Visual Studio Code (VS Code)

### Hotkeys

**Мультикурсор**Основные команды:

Ctrl-D - добавлять курсоры по одному, на каждое совпадение (с выделенным куском)

Ctrl-Shift-L - добавить курсоры на все совпадения

Alt-Enter (в окошке поиска) - добавить курсоры на все совпадения к запросу

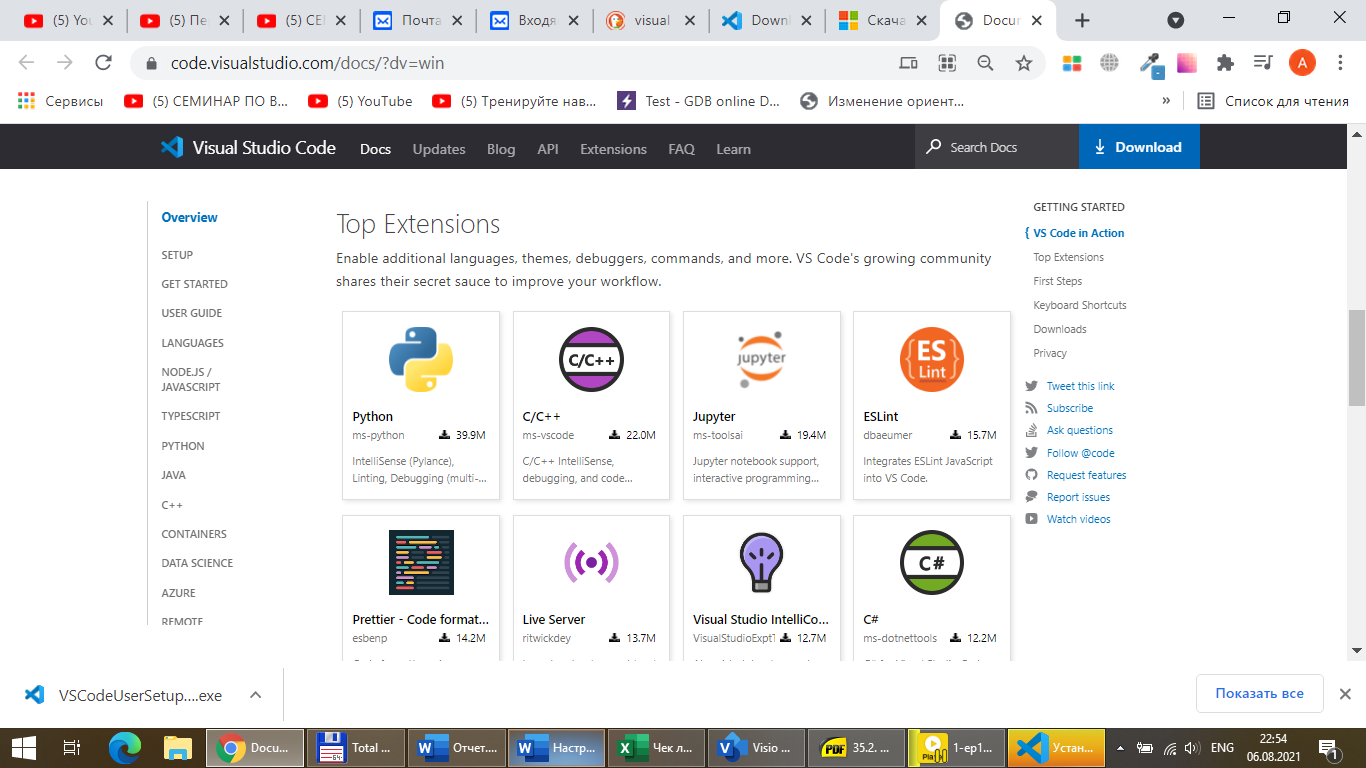
## Install

Downloads

[Download VS Code](https://code.visualstudio.com/download) - Quickly find the appropriate install for your platform

<https://code.visualstudio.com/docs/?dv=win>

<https://code.visualstudio.com/download>



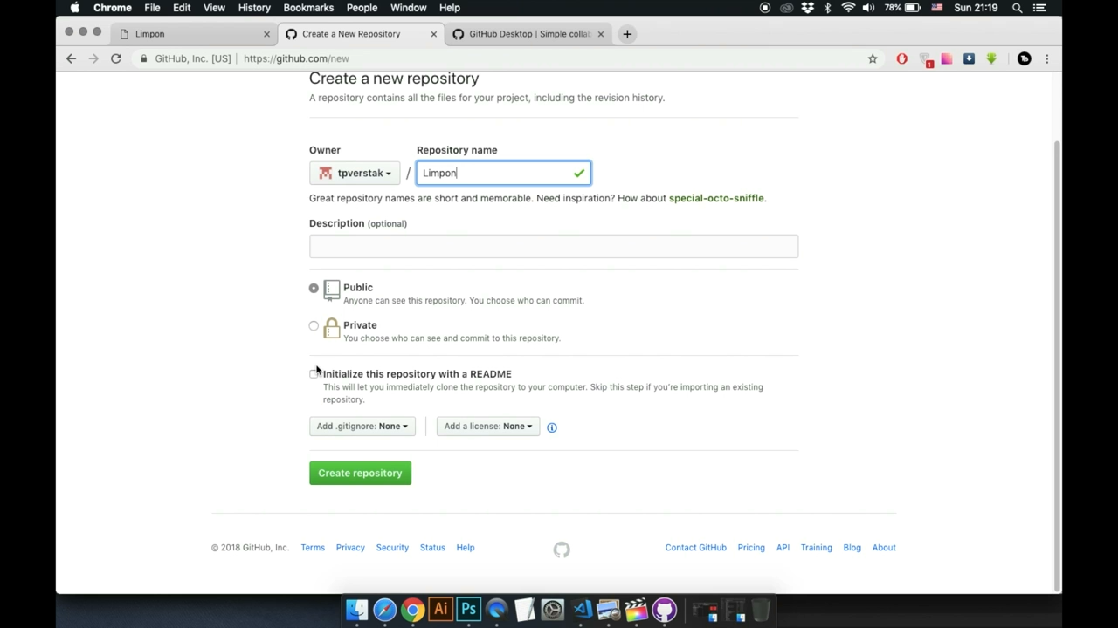
Russian language Pack

ctrl+K ctrl+T Цветовая схема

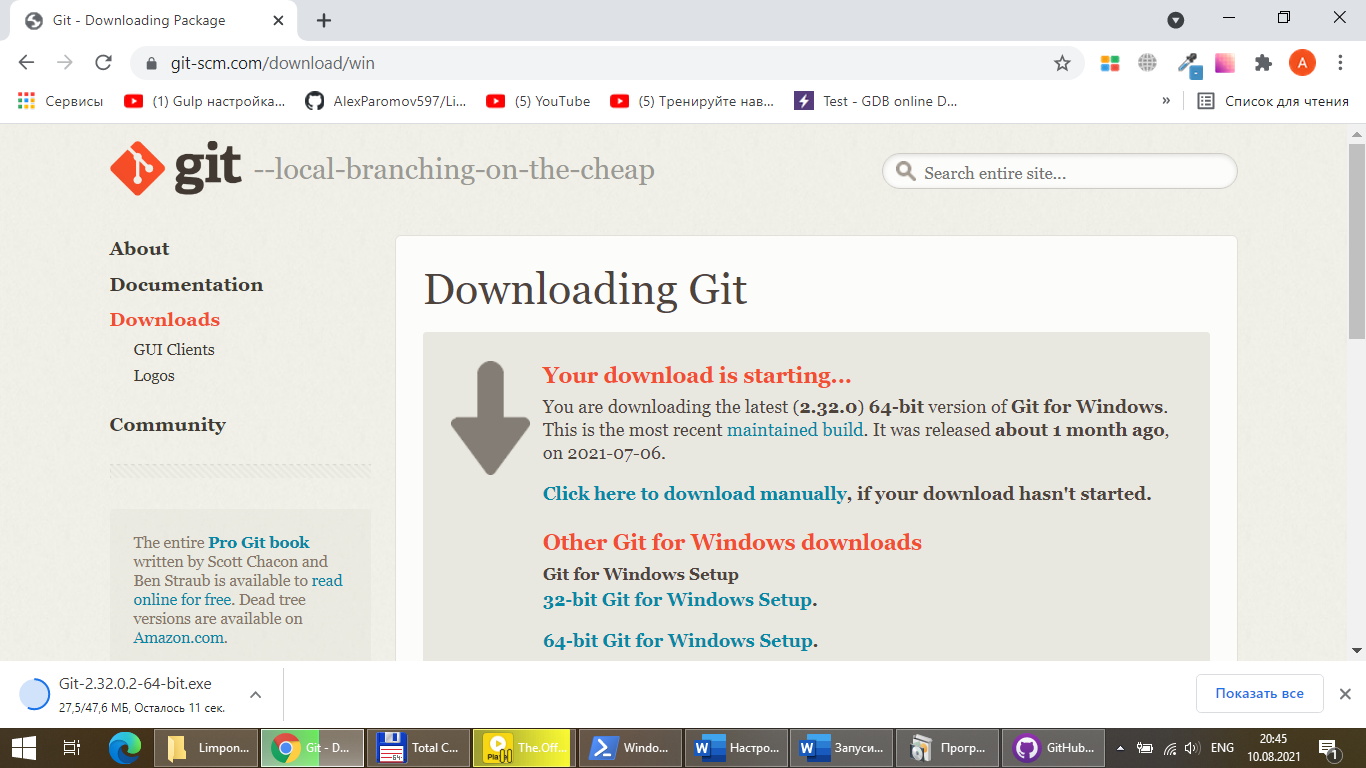
ctrl+shift+P Показать все команды

## GitHub

Создаем аккаунт

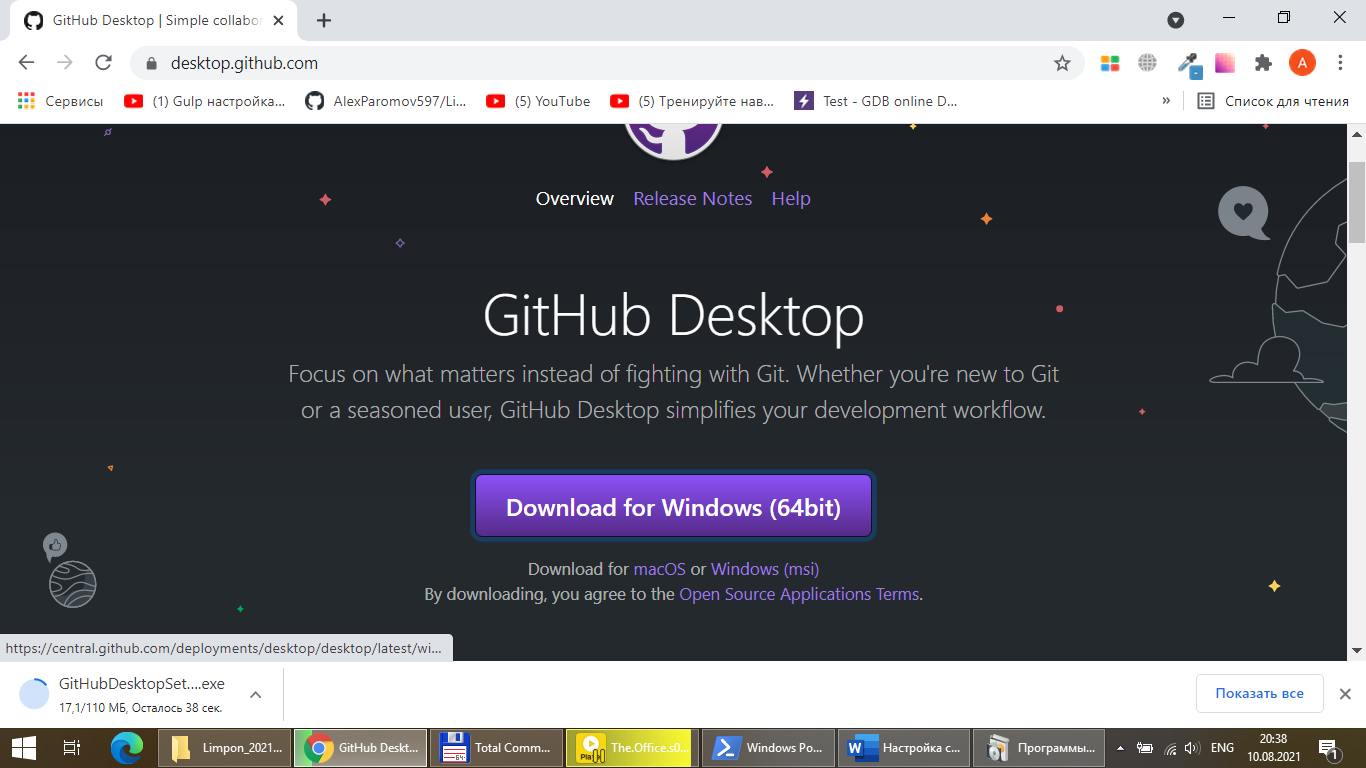
Ч GitHub https://github.com/ 

<https://git-scm.com/>



Ставим десктоп

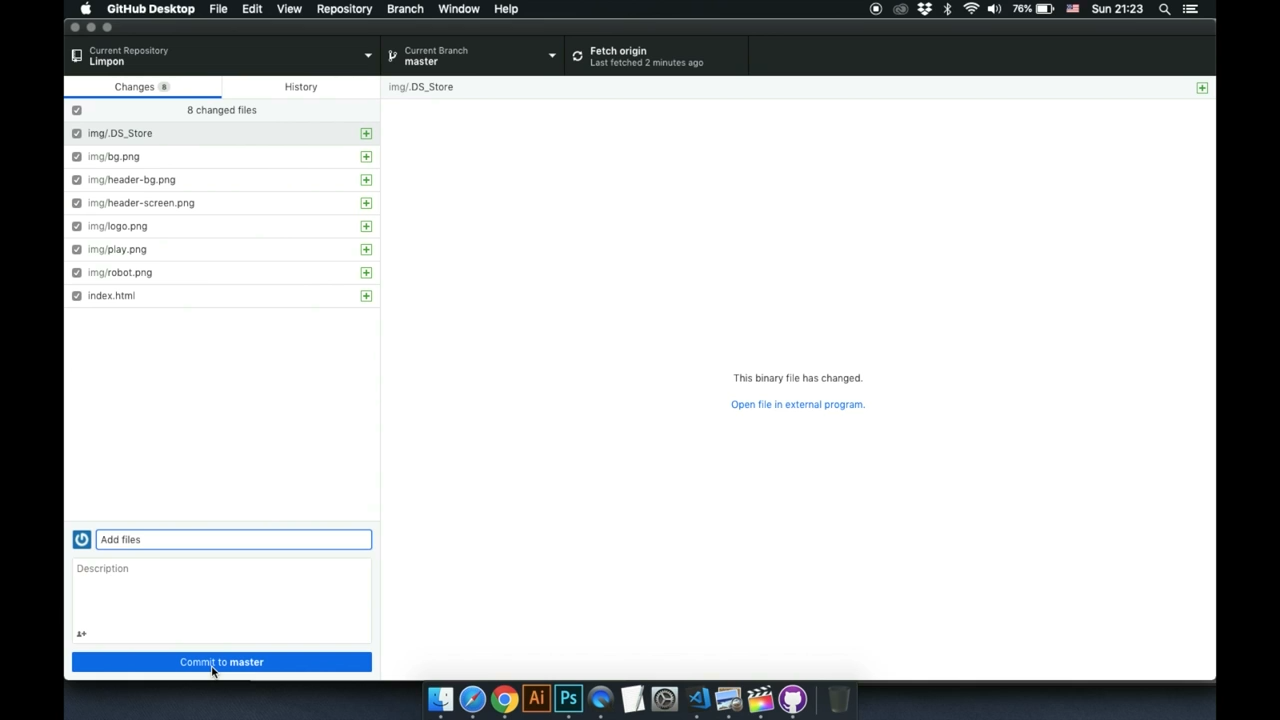
Ч GitHub Desktop <https://desktop.github.com/>



Авторизуемся,

|  |  |  |  |
| --- | --- | --- | --- |
| [alex5170603@mail.ru](mailto:alex5170603@mail.ru) | <https://github.com/login> | AlexParomov597 | a5170603 |

клонируем репозиторий в созданную в корне папку, добавляем в нее файлы, добавляем summary, commit, push



## Git

<https://git-scm.com/>

—сылки с урока:

## Google Fonts https://fonts.google.com

<https://fonts-online.ru/fonts/open-sans>

Gulp на Github (https://github.com/gulpjs/gulp)

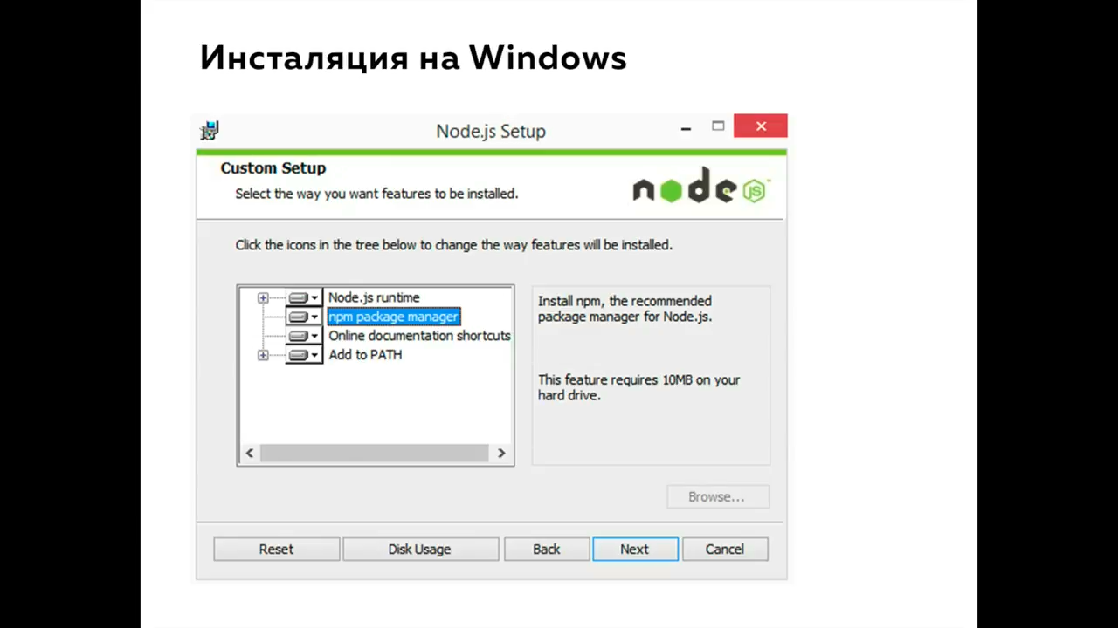
Документация Gulp (https://gulpjs.org/)

Каталог плагинов для Gulp (https://gulpjs.com/plugins/)

Описание установки Gulp (http://tpverstak.ru/gulp/)

Офиц.сайт Gulp (https://gulpjs.com/)

Установка Node (<https://nodejs.org/en/>)



npm install --global yarn

<https://git-scm.com/download/win>

<https://tpverstak.ru/gulp/>

* Установка
* установите [NodeJS](https://nodejs.org/en/) (если требуется)
* ~~npm install -g npm~~
* и
* [Yarn](https://yarnpkg.com/en/docs/install) npm install --global yarn
* скачайте сборку с помощью [Git](https://git-scm.com/downloads):

git clone <https://github.com/andreyalexeich/gulp-scss-starter.git>

git clone https://github.com/dudeonthehorse/kilogram.git

* ~~// установите gulp глобально: yarn global add gulp-cli~~
* ~~установите bem-tools-core глобально: yarn global add bem-tools-core~~

В проводнике на папке Code shift+right mouse открыть терминал **GitBash**

* Копируйте папку gulp-scss-starter в свою без readme и .git

cd ..

cd Limpon\_2021

* перейдите в скачанную папку со сборкой: cd gulp-scss-starter
* скачайте необходимые зависимости: yarn
* чтобы начать работу, введите команду: yarn run dev (режим разработки)
* чтобы собрать проект, введите команду yarn run build (режим сборки)

Если вы всё сделали правильно, у вас должен открыться браузер с локальным сервером. Режим сборки предполагает оптимизацию проекта: сжатие изображений, минифицирование CSS и JS-файлов для загрузки на сервер.

http://localhost:4000/

## <http://htmlbook.ru/>

### Cправочник CSS свойств <http://htmlbook.ru/css>

—сылки с урока:

Ч https://cdnjs.com/libraries/normalize

Ч https://code.tutsplus.com/ru/tutorials/the-30-css-selectors-you-must-memorize--net-16048

Ч https://codepen.io/anna\_blok/pen/BGrGJQ

Ч https://tpverstak.ru/seo-html-template/

Ч https://pp.userapi.com/c830109/v830109382/13e819/cfhO9oTl-tI.jpg

### HTML

!<Tab>        => <Структура HTML-документа>

div<Tab>    => <div></div>

span<Tab>   => <span></span>

p<Tab>      => <p></p>

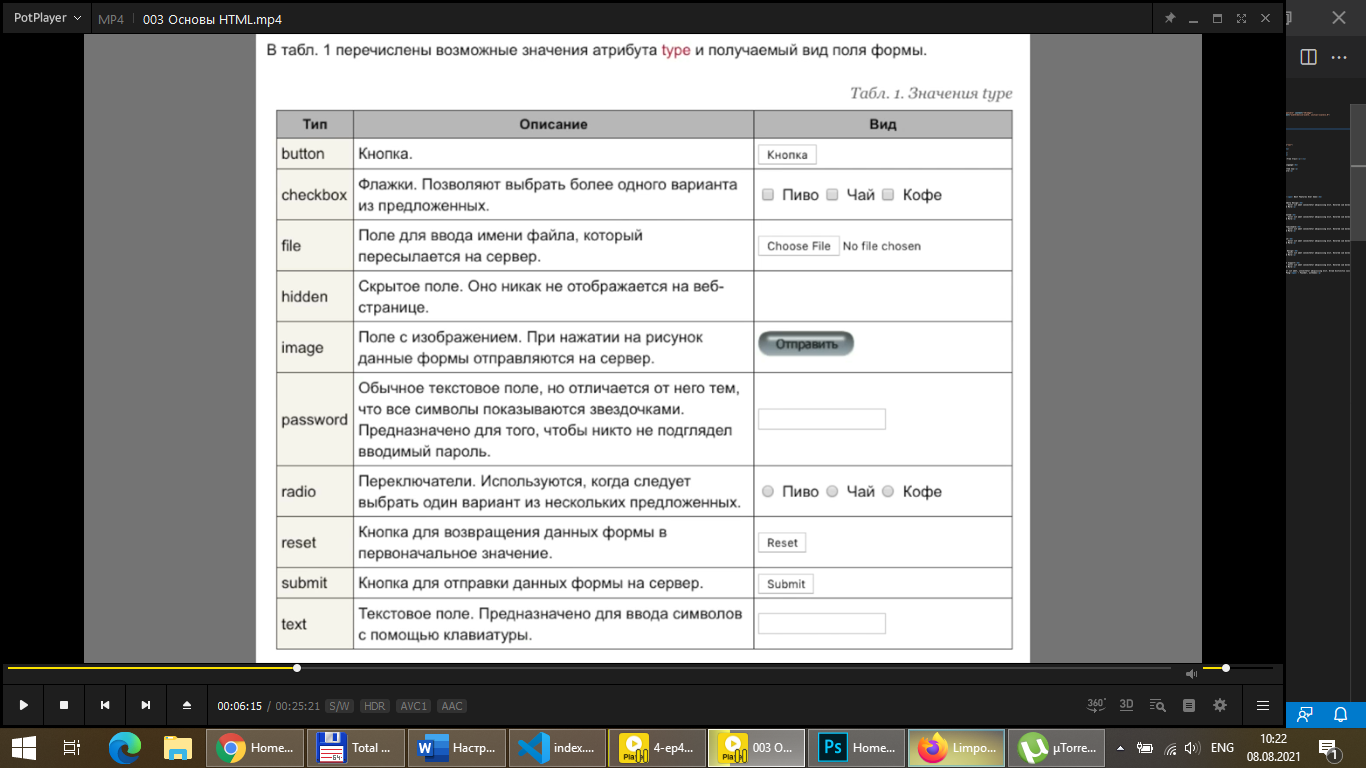
c<Tab>       => <!--  -->

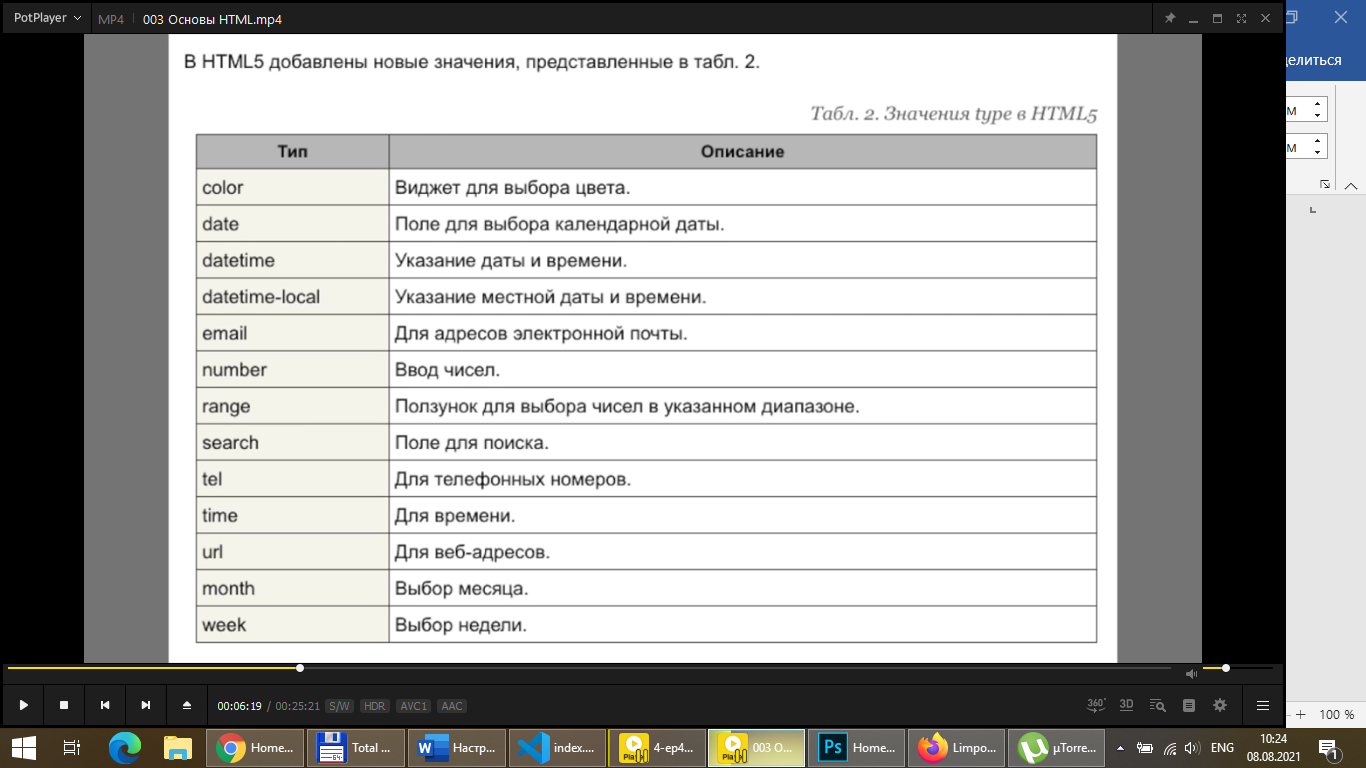
.row<Tab>           => <div class="row"></div>

.row.mb-3<Tab>      => <div class="row mb-3"></div>

#div1<Tab>          => <div id="div1"></div>

#div1.row<Tab>      => <div id="div1" class="row"></div>





—ссылки с урока:

Ч типы для¤ input http://htmlbook.ru/html/input/type

Ч Атрибуты input http://htmlbook.ru/html/input

Ч Атрибуты textarea http://htmlbook.ru/html/textarea

### **Отступы**

Внешние отступы:

m<Tab>      => margin: ;              m:a<Tab>   => margin: auto;

m10<Tab>   => margin: 10px;           m10px<Tab> => margin: 10px;

ml<Tab>     => margin-left: ;         ml:a<Tab>  => margin-left: auto;

mr<Tab>     => margin-right: ;        mr:a<Tab>  => margin-right: auto;

mt<Tab>     => margin-top: ;          mt:a<Tab>  => margin-top: auto;

mb<Tab>     => margin-bottom: ;       mb:a<Tab>  => margin-bottom: auto;

Внутренние отступы:

p<Tab>      => padding: ;

p10<Tab>    => padding: 10px;        p10px<Tab> => padding: 10px;

pl<Tab>     => padding-left: ;        pr<Tab>    => padding-right: ;

pt<Tab>     => padding-top: ;         pb<Tab>    => padding-bottom: ;

### **Ввод числового значения атрибута**

После комбинации символов допускается указание числового значения, которое станет значением атрибута стилей с добавлением (при необходимости) единицы изменения px (для целочисленных значений) или em (для дробных значений):

m10<Tab>   => margin: 10px;

p2.5<Tab>  => padding: 2.5em;

lh1.5<Tab> => line-height: 1.5;

Если нужно указать несколько значений, то они задаются через дефис:

m10-20<Tab> => margin: 10px 20px;

Для ввода отрицательных значений следует указать два дефиса:

p-15--5<Tab> => padding: -15px -5px;

После значения можно указать единицу измерения. В этом случае разделять значения дефисом не нужно:

p10px20px<Tab> => padding: 10px 20px;

Вместо реальной единицы измерения допускается указание псевдонимов: p для %, e для em, x для ex:

p10p20e30x5<Tab> => padding: 10% 20em 30ex 5px;

Для вставки значения !important достаточно ввести восклицательный знак:

m20!<Tab>        => margin: 20px !important;

В качестве значения для цвета можно указать одну, две, три или шесть шестнадцатеричных цифр после символа #. Если цифры не заданы, то получим черный цвет:

c#<Tab>          => color: #000000;

c#f<Tab>         => color: #ffffff;

c#e8<Tab>        => color: #e8e8e8;

c#ec0<Tab>       => color: #eecc00;

c#28a745<Tab>    => color: #28a745;

Interactive Editor Playground (help)

*Alt + Z* весь наш текст перенесется

создать div с классом block для этого я пишу *.block* и нажимаю Tab

создать маркированный список на 10 элементов. Пишем *ul>li\*10* Tab

свои сокращения-сниппеты для быстрого вызова того иного куска кода.  
Переходим в *Управление > Пользовательские фрагменты кода*. Здесь есть выбор:

* Создать сниппеты для конкретного языка;  
  Создать сниппеты для всех языков, но для конкретного проекта;  
  Создать глобальные сниппеты для всех языков и для всех проектов.

## Учебник по Node.js в Visual Studio Code

<https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>

Node.js - это платформа для создания быстрых и масштабируемых серверных приложений с использованием JavaScript. Node.js - это среда выполнения, а npm - менеджер пакетов для модулей Node.js.

Visual Studio Code имеет поддержку языков JavaScript и TypeScript из коробки, а также отладку Node.js. Однако, **чтобы запустить приложение Node.js, вам необходимо установить среду выполнения Node.js на вашу машину**.

Чтобы начать это руководство, установите Node.js для вашей платформы. Менеджер пакетов Node включен в дистрибутив Node.js. Вам нужно будет открыть новый терминал (командную строку), чтобы инструменты командной строки node и npm были в вашем PATH.

Чтобы проверить, правильно ли установлен Node.js на вашем компьютере, откройте новый терминал и введите node --version, и вы увидите текущую установленную версию Node.js.

## Emmet: ускорение набора HTML-кода

Расширение Emmet позволяет значительно ускорить набор кода, путем ввода специальной комбинации символов и нажатия клавиши <Tab>. Вместо этой комбинации будет вставлен фрагмент кода. Расширение не нуждается в отдельной установке, т. к. входит в состав компонентов редактора VS Code.

*Примечание*

Комбинации, заканчивающиеся на + (например, ul+), в редакторе VS Code не работают.

### Вставка структуры HTML-документа

Для вставки структуры HTML-документа достаточно ввести восклицательный знак и нажать клавишу <Tab>. В результате будет вставлен код:

<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>Document</title>

</head>

<body>

</body>

</html>

Внутри шаблона помечены точки останова, перемещаться между которыми можно с помощью клавиши <Tab>:

!<Tab>        => <Структура HTML-документа>

html:5<Tab>   => <Структура HTML-документа>

doc<Tab>      => <Структура HTML-документа без тега doctype>

html:xml<Tab> => <html xmlns="http://www.w3.org/1999/xhtml"></html>

Три восклицательных знака позволяют вставить тег <!doctype>:

!!!<Tab>      => <!DOCTYPE html>

В структуре HTML-документа в параметре lang тега <html> по умолчанию указывается код английского языка. Чтобы выводился код русского языка нужно в настройках редактора выполнить поиск параметра Emmet: Variables и указать значение ru для переменной lang или просто добавить следующий фрагмент кода в конец файла settings.json:

"emmet.variables": {

    "lang": "ru",

    "charset": "utf-8"

}

Результат после изменения значений переменных при вводе комбинации !<Tab>:

<!DOCTYPE html>

<html lang="ru">

<head>

   <meta charset="utf-8">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>Document</title>

</head>

<body>

</body>

</html>

### **Вставка HTML-комментария**

Буква c позволяет вставить HTML-комментарий:

c<Tab>       => <!--  -->

cc:ie<Tab>   => <!--[if IE]><![endif]-->

cc:noie<Tab> => <!--[if !IE]><!--><!--<![endif]-->

Чтобы закомментировать блок кода, нужно предварительно его выделить, а затем в меню Правка выбрать пункт **Переключить комментарий строки** или нажать комбинацию клавиш <Ctrl>+</>. Повторное действие удаляет комментарий. Можно также выбрать пункт **Переключить комментарий блока** или нажать комбинацию клавиш <Shift>+<Alt>+<A>.

### **Теги из раздела HEAD**

Комбинации для тега <meta>:

meta<Tab>        => <meta>

meta:desc<Tab>   => <meta name="description" content="">

meta:kw<Tab>     => <meta name="keywords" content="">

meta:compat<Tab> => <meta http-equiv="X-UA-Compatible" content="IE=7">

meta:edge<Tab>   => <meta http-equiv="X-UA-Compatible" content="ie=edge">

meta:redirect<Tab> =>

   <meta http-equiv="refresh" content="0; url=http://example.com">

meta:utf<Tab>      =>

   <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">

meta:vp<Tab>       =>

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

Комбинации для тега <link>:

link<Tab>          => <link rel="stylesheet" href="">

link:css<Tab>      => <link rel="stylesheet" href="style.css">

link:favicon<Tab>  =>

   <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">

link:im<Tab>       => <link rel="import" href="component.html">

link:import<Tab>   => <link rel="import" href="component.html">

link:manifest<Tab> => <link rel="manifest" href="manifest.json">

link:mf<Tab>       => <link rel="manifest" href="manifest.json">

link:print<Tab> => <link rel="stylesheet" href="print.css" media="print">

link:rss<Tab>   =>

   <link rel="alternate" href="rss.xml" type="application/rss+xml"

         title="RSS">

link:touch<Tab> => <link rel="apple-touch-icon" href="favicon.png">

link:atom<Tab>  =>

  <link rel="alternate" href="atom.xml" type="application/atom+xml"

         title="Atom">

Комбинации для тега <script>:

script<Tab>     => <script></script>

script:src<Tab> => <script src=""></script>

Комбинация для тега <base>:

base<Tab>       => <base href="">

### **Основные теги из раздела BODY**

При вводе названий тегов, не имеющих отдельных комбинаций, получим открывающий и закрывающий тег:

div<Tab>    => <div></div>

span<Tab>   => <span></span>

p<Tab>      => <p></p>

Для некоторых тегов генерируются параметры, а также учитывается, является ли тег парным или одинарным:

abbr<Tab>   => <abbr title=""></abbr>

area<Tab>   => <area shape="" coords="" href="" alt="">

area:c<Tab> => <area shape="circle" coords="" href="" alt="">

area:d<Tab> => <area shape="default" coords="" href="" alt="">

area:p<Tab> => <area shape="poly" coords="" href="" alt="">

area:r<Tab> => <area shape="rect" coords="" href="" alt="">

audio<Tab>  => <audio src=""></audio>

br<Tab>     => <br>

map<Tab>    => <map name=""></map>

marquee<Tab> => <marquee behavior="" direction=""></marquee>

video<Tab>   => <video src=""></video>

Комбинации для тега <a>:

a<Tab>       => <a href=""></a>

a:link<Tab>  => <a href="http://"></a>

a:mail<Tab>  => <a href="mailto:"></a>

a:tel<Tab>   => <a href="tel:+"></a>

a:blank<Tab> =>

   <a href="http://" target="\_blank" rel="noopener noreferrer"></a>

Комбинации для вставки изображений:

img<Tab>        => <img src="" alt="">

img:s<Tab>      => <img src="" alt="" srcset="">

img:srcset<Tab> => <img src="" alt="" srcset="">

ri:d<Tab>       => <img src="" alt="" srcset="">

ri:dpr<Tab>     => <img src="" alt="" srcset="">

img:sizes<Tab>   => <img src="" alt="" sizes="" srcset="">

img:z<Tab>       => <img src="" alt="" sizes="" srcset="">

ri:v<Tab>        => <img src="" alt="" sizes="" srcset="">

ri:viewport<Tab> => <img src="" alt="" sizes="" srcset="">

pic<Tab>         => <picture></picture>

src<Tab>         => <source>

source<Tab>      => <source>

src:m<Tab>        => <source media="(min-width: )" srcset="">

source:media<Tab> => <source media="(min-width: )" srcset="">

src:mz<Tab>       =>

   <source media="(min-width: )" srcset="" sizes="">

source:media:sizes<Tab> =>

   <source media="(min-width: )" srcset="" sizes="">

src:mt<Tab>            =>

   <source media="(min-width: )" srcset="" type="image/">

source:media:type<Tab> =>

   <source media="(min-width: )" srcset="" type="image/">

src:z<Tab>             => <source sizes="" srcset="">

source:sizes<Tab>      => <source sizes="" srcset="">

src:zt<Tab>            => <source sizes="" srcset="" type="image/">

source:sizes:type<Tab> => <source sizes="" srcset="" type="image/">

src:sc<Tab>            => <source src="" type="">

source:src<Tab>        => <source src="" type="">

src:s<Tab>             => <source srcset="">

source:srcset<Tab>     => <source srcset="">

src:t<Tab>             => <source srcset="" type="image/">

source:type<Tab>       => <source srcset="" type="image/">

ri:a<Tab> или ri:art<Tab> =>

   <picture>

      <source media="(min-width: )" srcset="">

      <img src="" alt="">

   </picture>

ri:t<Tab> или ri:type<Tab> =>

   <picture>

      <source srcset="" type="image/">

      <img src="" alt="">

   </picture>

В следующих случаях можно использовать сокращенные названия тегов:

mn<Tab>         => <main></main>

hdr<Tab>        => <header></header>

sect<Tab>       => <section></section>

art<Tab>        => <article></article>

ftr<Tab>        => <footer></footer>

adr<Tab>        => <address></address>

bq<Tab>         => <blockquote></blockquote>

emb<Tab>        => <embed src="" type="">

fig<Tab>        => <figure></figure>

figc<Tab>       => <figcaption></figcaption>

ifr<Tab>        => <iframe src="" frameborder="0"></iframe>

iframe<Tab>     => <iframe src="" frameborder="0"></iframe>

str<Tab>        => <strong></strong>

det<Tab>           => <details></details>

det>summary+p<Tab> =>

   <details>

      <summary></summary>

      <p></p>

   </details>

### **Добавление текста**

После вставки шаблона тега без параметров текстовый курсор будет расположен перед закрывающим тегом, что позволяет сразу начать набор текста:

p<Tab>        => <p>{Курсор}</p>

Текст можно также добавить в составе комбинации внутри фигурных скобок:

p{Текст}<Tab> => <p>Текст</p>

p{Текст1 }>a{ссылка}+{ текст2}<Tab> =>

   <p>Текст1 <a href="">ссылка</a> текст2</p>

Если в составе текста указать символ $, то вместо него будет подставлен порядковый номер элемента. Создадим три абзаца и пронумеруем их:

p\*3>{Абзац $}<Tab> =>

   <p>Абзац 1</p>

   <p>Абзац 2</p>

   <p>Абзац 3</p>

Если указать несколько символов $ подряд, то перед номером добавятся нули:

p\*3>{Абзац $$$}<Tab> =>

   <p>Абзац 001</p>

   <p>Абзац 002</p>

   <p>Абзац 003</p>

По умолчанию нумерация элементов начинается с 1. После символа @ можно указать другой начальный номер, например, 3:

p\*3>{Абзац $@3}<Tab> =>

   <p>Абзац 3</p>

   <p>Абзац 4</p>

   <p>Абзац 5</p>

Для вставки текста-заполнителя следует ввести слово lorem после которого можно указать количество генерируемых слов (задавать количество слов необязательно). Создадим три абзаца и добавим в них по пять слов:

p\*3>lorem5<Tab> =>

   <p>Lorem ipsum dolor sit amet.</p>

   <p>Eligendi commodi sit temporibus. Doloribus?</p>

   <p>Rem dignissimos eum consectetur molestias.</p>

После слова lorem можно указать код языка. Сгенерируем текст на русском языке:

p>loremru20<Tab> =>

   <p>Далеко-далеко за словесными горами в стране гласных и согласных

   живут рыбные тексты. Ручеек безопасную, встретил маленький раз гор

   lorem назад. Свою, которой?</p>

Не пытайтесь найти смысл во вставленном тексте. Основная цель lorem показать заказчику, как будет выглядеть дизайн сайта с произвольным текстом.

### **Добавление параметров тегов**

Если указать только CSS-селектор стилевого класса или идентификатора, то будет создан тег <div> с переданным стилевым классом или идентификатором:

.row<Tab>           => <div class="row"></div>

.row.mb-3<Tab>      => <div class="row mb-3"></div>

#div1<Tab>          => <div id="div1"></div>

#div1.row<Tab>      => <div id="div1" class="row"></div>

Чтобы добавить эти параметры в другой тег, предварительно указываем его название:

p.text-center<Tab>  => <p class="text-center"></p>

Если эти параметры вкладываются в блочный тег, то они добавляются к тегу <span>:

p>.text-center<Tab> => <p><span class="text-center"></span></p>

или к другим вложенным тегам в зависимости от родительского элемента:

ul>.item<Tab>       =>

   <ul>

      <li class="item"></li>

   </ul>

table>.row>.col<Tab> =>

   <table>

      <tr class="row">

         <td class="col"></td>

      </tr>

   </table>

Для добавления произвольных параметров, их следует указывать внутри квадратных скобок. Значение параметра задается после символа = внутри одинарных или двойных кавычек. Если значение не содержит пробела, то кавычки можно не указывать. Если параметр не содержит значения, то достаточно вставить только название параметра:

p[title="Подсказка" class=p$]\*2<Tab> =>

   <p title="Подсказка" class="p1"></p>

   <p title="Подсказка" class="p2"></p>

p[title]<Tab> => <p title=""></p>

### **Формы и элементы управления**

Для вставки формы и элементов управления используются следующие комбинации символов:

form<Tab>           => <form action=""></form>

form:get<Tab>       => <form action="" method="get"></form>

form:post<Tab>      => <form action="" method="post"></form>

label<Tab>          => <label for=""></label>

input:b<Tab>        => <input type="button" value="">

input:button<Tab>   => <input type="button" value="">

input:reset<Tab>    => <input type="reset" value="">

input:s<Tab>        => <input type="submit" value="">

input:submit<Tab>   => <input type="submit" value="">

btn<Tab>            => <button></button>

button<Tab>         => <button></button>

btn:r<Tab>          => <button type="reset"></button>

button:r<Tab>       => <button type="reset"></button>

button:reset<Tab>   => <button type="reset"></button>

btn:s<Tab>          => <button type="submit"></button>

button:s<Tab>       => <button type="submit"></button>

button:submit<Tab>  => <button type="submit"></button>

btn:d<Tab>           => <button disabled="disabled"></button>

button:d<Tab>        => <button disabled="disabled"></button>

button:disabled<Tab> => <button disabled="disabled"></button>

tarea<Tab>    => <textarea name="" id="" cols="30" rows="10"></textarea>

textarea<Tab> => <textarea name="" id="" cols="30" rows="10"></textarea>

input<Tab>          => <input type="text">

inp<Tab>            => <input type="text" name="" id="">

input:t<Tab>       => <input type="text" name="" id="">

input:text<Tab>     => <input type="text" name="" id="">

input:p<Tab>        => <input type="password" name="" id="">

input:password<Tab> => <input type="password" name="" id="">

input:url<Tab>      => <input type="url" name="" id="">

input:email<Tab>    => <input type="email" name="" id="">

input:tel<Tab>      => <input type="tel" name="" id="">

input:number<Tab>   => <input type="number" name="" id="">

input:search<Tab>   => <input type="search" name="" id="">

input:date<Tab>     => <input type="date" name="" id="">

input:time<Tab>     => <input type="time" name="" id="">

input:datetime<Tab> => <input type="datetime" name="" id="">

input:datetime-local<Tab> => <input type="datetime-local" name="" id="">

input:month<Tab>    => <input type="month" name="" id="">

input:week<Tab>     => <input type="week" name="" id="">

input:h<Tab>        => <input type="hidden" name="">

input:hidden<Tab>   => <input type="hidden" name="">

datal<Tab>          => <datalist></datalist>

select<Tab>   => <select name="" id=""></select>

select:d<Tab> => <select name="" id="" disabled="disabled"></select>

select:disabled<Tab> =>

   <select name="" id="" disabled="disabled"></select>

opt<Tab>      => <option value=""></option>

option<Tab>   => <option value=""></option>

optg<Tab>     => <optgroup></optgroup>

optgroup<Tab> => <optgroup></optgroup>

select>option\*3<Tab> =>

   <select name="" id="">

      <option value=""></option>

      <option value=""></option>

      <option value=""></option>

  </select>

input:c<Tab>        => <input type="checkbox" name="" id="">

input:checkbox<Tab> => <input type="checkbox" name="" id="">

input:r<Tab>        => <input type="radio" name="" id="">

input:radio<Tab>    => <input type="radio" name="" id="">

input:f<Tab>        => <input type="file" name="" id="">

input:file<Tab>     => <input type="file" name="" id="">

input:i<Tab>        => <input type="image" src="" alt="">

input:image<Tab>    => <input type="image" src="" alt="">

input:range<Tab>    => <input type="range" name="" id="">

input:color<Tab>    => <input type="color" name="" id="">

prog<Tab>           => <progress></progress>

fst<Tab>            => <fieldset></fieldset>

fset<Tab>           => <fieldset></fieldset>

fst:d<Tab>          => <fieldset disabled="disabled"></fieldset>

fset:d<Tab>         => <fieldset disabled="disabled"></fieldset>

leg<Tab>            => <legend></legend>

legend<Tab>         => <legend></legend>

### **Таблицы**

Для быстрой вставки таблиц, строк и ячеек можно использовать такие комбинации:

table<Tab>           => <table></table>

cap<Tab>             => <caption></caption>

tr<Tab>              => <tr></tr>

th<Tab>              => <th></th>

td<Tab>              => <td></td>

thead<Tab>           => <thead></thead>

tbody<Tab>           => <tbody></tbody>

tfoot<Tab>           => <tfoot></tfoot>

colg<Tab>            => <colgroup></colgroup>

col<Tab>             => <col>

table>tr\*2>td\*2<Tab> =>

   <table>

      <tr>

         <td></td>

         <td></td>

      </tr>

      <tr>

         <td></td>

         <td></td>

      </tr>

   </table>

thead>tr>th\*2<Tab> =>

   <thead>

      <tr>

         <th></th>

         <th></th>

      </tr>

   </thead>

### **Списки**

Для быстрой вставки списков можно использовать следующие комбинации:

ul<Tab>      => <ul></ul>

ol<Tab>      => <ol></ol>

li<Tab>      => <li></li>

ul>li\*3<Tab> =>

   <ul>

      <li></li>

      <li></li>

      <li></li>

   </ul>

ul>li\*3>{Пункт $}<Tab> =>

   <ul>

      <li>Пункт 1</li>

      <li>Пункт 2</li>

      <li>Пункт 3</li>

   </ul>

ol>.item$\*3<Tab> =>

   <ol>

      <li class="item1"></li>

      <li class="item2"></li>

      <li class="item3"></li>

   </ol>

dl<Tab>           => <dl></dl>

dt<Tab>           => <dt></dt>

dd<Tab>           => <dd></dd>

dl>(dt+dd)\*2<Tab> =>

   <dl>

      <dt></dt>

      <dd></dd>

      <dt></dt>

      <dd></dd>

   </dl>

### **Вставка сразу нескольких тегов**

Чтобы вставить несколько одинаковых тегов друг за другом, следует добавить символ \*, после которого указать количество повторов. Вставим три абзаца и пронумеруем их:

p\*3>{Абзац $}<Tab> =>

   <p>Абзац 1</p>

   <p>Абзац 2</p>

   <p>Абзац 3</p>

Символ > означает, что нужно выполнить вложение внутрь тега. В предыдущем примере мы вкладывали текст. Вложим пункт внутрь списка:

ul>li<Tab> =>

   <ul>

      <li></li>

   </ul>

Символ + добавляет элемент после текущего элемента:

div+p+ul>li\*3<Tab> =>

   <div></div>

   <p></p>

   <ul>

      <li></li>

      <li></li>

      <li></li>

   </ul>

Как видно из примера, можно комбинировать символы \*, > и +, создавая сложные структуры. В последнем подвыражении мы вставляли пункты внутрь списка. Если попробовать в конец выражения вставить символ + и тег, то это тег будет вставлен после тега <li>. Как же нам вставить тег после списка? Чтобы переместиться выше по дереву на один уровень нужно указать символ ^:

ul>li\*3^div<Tab> =>

   <ul>

      <li></li>

      <li></li>

      <li></li>

   </ul>

   <div></div>

Чтобы переместиться на несколько уровней выше, следует символ ^ указать несколько раз:

div>p>span+i^^div<Tab> =>

   <div>

      <p><span></span><i></i></p>

   </div>

   <div></div>

Можно также ограничить вложение, добавив группу внутрь круглых скобок:

(div>p>span+i)+div<Tab> =>

   <div>

      <p><span></span><i></i></p>

   </div>

   <div></div>

## Emmet: ускорение набора CSS-кода

Специальные комбинации символов, рассмотренные в предыдущих разделах, работают только в файлах с расширениями html, php, jsx и некоторых других. В CSS-файлах (и в файлах CSS-препроцессоров) нужно использовать другие комбинации, которые мы рассмотрим в этом разделе.

### **Форматирование шрифта**

Одновременное указание характеристик шрифта:

f<Tab>     => font: 1em sans-serif;

Имя шрифта:

ff<Tab>    => font-family: serif;    ff:s<Tab>  => font-family: serif;

ff:ss<Tab> => font-family: sans-serif;

ff:c<Tab>  => font-family: cursive;

ff:f<Tab>  => font-family: fantasy;

ff:m<Tab>  => font-family: monospace;

ff:a<Tab>  => font-family: auto;

ff:u<Tab>  => font-family: unset;

ff:i<Tab>  => font-family: inherit;

fft<Tab>   =>

   font-family: "Times New Roman", Times, Baskerville, Georgia, serif;

ffa<Tab>   =>

   font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;

ffv<Tab>   => font-family: Verdana, Geneva, sans-serif;

Стиль шрифта:

fs<Tab>    => font-style: italic;    fs:i<Tab>  => font-style: italic;

fs:n<Tab>  => font-style: normal;

fs:o<Tab>  => font-style: oblique;

Размер шрифта:

fsz<Tab>   => font-size: ;

fsza<Tab>  => font-size-adjust: ;

Цвет текста:

c<Tab>     => color: #000;

Жирность шрифта:

fw<Tab>    => font-weight: normal;   fw:n<Tab>  => font-weight: normal;

fw:b<Tab>  => font-weight: bold;

fw:lr<Tab> => font-weight: lighter;

fw:br<Tab> => font-weight: bolder;

Вид строчных букв:

fv<Tab>    => font-variant: normal;

fv:n<Tab>  => font-variant: normal;

fv:sc<Tab> => font-variant: small-caps;

Загружаемые шрифты:

@f<Tab>    =>

   @font-face {

      font-family: ;

      src: url();

   }

@ff<Tab> =>

   @font-face {

      font-family: 'FontName';

      src: url('FileName.eot');

      src: url('FileName.eot?#iefix') format('embedded-opentype'),

          url('FileName.woff') format('woff'),

          url('FileName.ttf') format('truetype'),

          url('FileName.svg#FontName') format('svg');

      font-style: normal;

      font-weight: normal;

   }

Прочие атрибуты:

fd<Tab>     => font-display: auto;

fd:a<Tab>   => font-display: auto;

fd:b<Tab>   => font-display: block;

fd:s<Tab>   => font-display: swap;

fd:f<Tab>   => font-display: fallback;

fd:o<Tab>   => font-display: optional;

fef<Tab>    => font-effect: none;

fef:n<Tab>  => font-effect: none;

fef:eg<Tab> => font-effect: engrave;

fef:eb<Tab> => font-effect: emboss;

fef:o<Tab>  => font-effect: outline;

fem<Tab>    => font-emphasize: ;

femp<Tab>   => font-emphasize-position: before;

femp:b<Tab> => font-emphasize-position: before;

femp:a<Tab> => font-emphasize-position: after;

fems<Tab>   => font-emphasize-style: none;

fems:n<Tab> => font-emphasize-style: none;

fems:ac<Tab> => font-emphasize-style: accent;

fems:dt<Tab> => font-emphasize-style: dot;

fems:ds<Tab> => font-emphasize-style: disc;

fems:c<Tab> => font-emphasize-style: circle;

fsm<Tab>    => font-smoothing: antialiased;

fsm:a<Tab>  => font-smoothing: antialiased;

fsm:s<Tab>  => font-smoothing: subpixel-antialiased;

fsm:n<Tab>  => font-smoothing: none;

fst<Tab>    => font-stretch: normal;

fst:n<Tab>  => font-stretch: normal;

fst:uc<Tab> => font-stretch: ultra-condensed;

fst:ue<Tab> => font-stretch: ultra-expanded;

fst:c<Tab>  => font-stretch: condensed;

fst:ec<Tab> => font-stretch: extra-condensed;

fst:ee<Tab> => font-stretch: extra-expanded;

fst:e<Tab>  => font-stretch: expanded;

fst:sc<Tab> => font-stretch: semi-condensed;

fst:se<Tab> => font-stretch: semi-expanded;

fvs<Tab>    => font-variation-settings: normal;

fvs:n<Tab>  => font-variation-settings: normal;

fvs:u<Tab>  => font-variation-settings: unset;

fvs:ih<Tab> => font-variation-settings: inherit;

fvs:il<Tab> => font-variation-settings: initial;

### **Форматирование текста**

Расстояние между символами в словах:

lts<Tab>    => letter-spacing: normal;

lts:n<Tab>  => letter-spacing: normal;

Расстояние между словами:

wos<Tab>    => word-spacing: ;

Отступ первой строки:

ti<Tab>     => text-indent: ;

Вертикальное расстояние между строками:

lh<Tab>     => line-height: ;

Горизонтальное выравнивание текста:

ta<Tab>     => text-align: left;

ta:l<Tab>   => text-align: left;

ta:r<Tab>   => text-align: right;

ta:c<Tab>   => text-align: center;

ta:j<Tab>   => text-align: justify;

tal<Tab>    => text-align-last: left;

tal:l<Tab>  => text-align-last: left;

tal:r<Tab>  => text-align-last: right;

tal:c<Tab>  => text-align-last: center;

tal:a<Tab>  => text-align-last: auto;

Вертикальное выравнивание текста:

va<Tab>     => vertical-align: top;

va:t<Tab>   => vertical-align: top;

va:bl<Tab>  => vertical-align: baseline;

va:m<Tab>   => vertical-align: middle;

va:b<Tab>   => vertical-align: bottom;

va:sub<Tab> => vertical-align: sub;

va:sup<Tab> => vertical-align: super;

va:tt<Tab>  => vertical-align: text-top;

va:tb<Tab>  => vertical-align: text-bottom;

Подчеркивание, надчеркивание и зачеркивание текста:

td<Tab>     => text-decoration: none;

td:n<Tab>   => text-decoration: none;

td:u<Tab>   => text-decoration: underline;

td:o<Tab>   => text-decoration: overline;

td:l<Tab>   => text-decoration: line-through;

Изменение регистра символов:

tt<Tab>     => text-transform: uppercase;

tt:u<Tab>   => text-transform: uppercase;

tt:l<Tab>   => text-transform: lowercase;

tt:c<Tab>   => text-transform: capitalize;

tt:n<Tab>   => text-transform: none;

Обработка пробелов между словами:

whs<Tab>    => white-space: nowrap;  whs:nw<Tab> => white-space: nowrap;

whs:n<Tab>  => white-space: normal;

whs:p<Tab>  => white-space: pre;

whs:pl<Tab> => white-space: pre-line;

whs:pw<Tab> => white-space: pre-wrap;

Перенос слов:

wow<Tab>    => word-wrap: none;      wow:n<Tab>  => word-wrap: none;

wow:nm<Tab> => word-wrap: normal;

wow:b<Tab>  => word-wrap: break-word;

wow:u<Tab>  => word-wrap: unrestricted;

wow:s<Tab>  => word-wrap: suppress;

wob<Tab>    => word-break: normal;

wob:n<Tab>  => word-break: normal;

wob:b<Tab>  => word-break: break-all;

wob:k<Tab>  => word-break: keep-all;

tw<Tab>     => text-wrap: none;      tw:n<Tab>   => text-wrap: none;

tw:nl<Tab>  => text-wrap: normal;

tw:u<Tab>   => text-wrap: unrestricted;

tw:s<Tab>   => text-wrap: suppress;

Атрибут content:

cnt<Tab>    => content: '';

ct<Tab>     => content: '';

cnt:n<Tab>  => content: normal;

ct:n<Tab>   => content: normal;

cnt:a<Tab>  => content: attr();

ct:a<Tab>   => content: attr();

cnt:oq<Tab> => content: open-quote;

ct:oq<Tab>  => content: open-quote;

cnt:cq<Tab> => content: close-quote;

ct:cq<Tab>  => content: close-quote;

cnt:noq<Tab> => content: no-open-quote;

ct:noq<Tab>  => content: no-open-quote;

cnt:ncq<Tab> => content: no-close-quote;

ct:ncq<Tab>  => content: no-close-quote;

cnt:c<Tab>   => content: counter();

ct:c<Tab>    => content: counter();

cnt:cns<Tab> => content: counters();

ct:cns<Tab>  => content: counters();

q<Tab>       => quotes: ;

coi<Tab>     => counter-increment: ;

cor<Tab>     => counter-reset: ;

Прочие атрибуты:

te<Tab>     => text-emphasis: none;

te:n<Tab>   => text-emphasis: none;

te:ac<Tab>  => text-emphasis: accent;

te:dt<Tab>  => text-emphasis: dot;

te:c<Tab>   => text-emphasis: circle;

te:ds<Tab>  => text-emphasis: disc;

te:b<Tab>   => text-emphasis: before;

te:a<Tab>   => text-emphasis: after;

th<Tab>     => text-height: auto;

th:a<Tab>   => text-height: auto;

th:f<Tab>   => text-height: font-size;

th:t<Tab>   => text-height: text-size;

th:m<Tab>   => text-height: max-size;

tj<Tab>     => text-justify: auto;

tj:a<Tab>   => text-justify: auto;

tj:iw<Tab>  => text-justify: inter-word;

tj:ii<Tab>  => text-justify: inter-ideograph;

tj:ic<Tab>  => text-justify: inter-cluster;

tj:d<Tab>   => text-justify: distribute;

tj:k<Tab>   => text-justify: kashida;

tj:t<Tab>   => text-justify: tibetan;

to<Tab>     => text-outline: 0 0 #000;

tr<Tab>     => text-replace: ;

tov<Tab>    => text-overflow: ellipsis;

tov:e<Tab>  => text-overflow: ellipsis;

tov:c<Tab>  => text-overflow: clip;

### **Отступы**

Внешние отступы:

m<Tab>      => margin: ;              m:a<Tab>   => margin: auto;

m10<Tab>   => margin: 10px;           m10px<Tab> => margin: 10px;

ml<Tab>     => margin-left: ;         ml:a<Tab>  => margin-left: auto;

mr<Tab>     => margin-right: ;        mr:a<Tab>  => margin-right: auto;

mt<Tab>     => margin-top: ;          mt:a<Tab>  => margin-top: auto;

mb<Tab>     => margin-bottom: ;       mb:a<Tab>  => margin-bottom: auto;

Внутренние отступы:

p<Tab>      => padding: ;

p10<Tab>    => padding: 10px;        p10px<Tab> => padding: 10px;

pl<Tab>     => padding-left: ;        pr<Tab>    => padding-right: ;

pt<Tab>     => padding-top: ;         pb<Tab>    => padding-bottom: ;

### **Рамки**

Одновременное задание характеристик рамки:

bd<Tab>     => border: 1px solid #000;

bd:n<Tab>   => border: none;

bdl<Tab>    => border-left: 1px solid #000;

bdr<Tab>    => border-right: 1px solid #000;

bdt<Tab>    => border-top: 1px solid #000;

bdb<Tab>    => border-bottom: 1px solid #000;

Стиль линии рамки:

bds<Tab>    => border-style: none;

bds:n<Tab>  => border-style: none;

bds:h<Tab>  => border-style: hidden;

bds:s<Tab>  => border-style: solid;

bds:dt<Tab> => border-style: dotted;

bds:ds<Tab> => border-style: dashed;

bds:db<Tab> => border-style: double;

bds:g<Tab>  => border-style: groove;

bds:r<Tab>  => border-style: ridge;

bds:i<Tab>  => border-style: inset;

bds:o<Tab>  => border-style: outset;

bds:w<Tab>  => border-style: wave;

bds:dtds<Tab>   => border-style: dot-dash;

bds:dtdtds<Tab> => border-style: dot-dot-dash;

bdls<Tab>       => border-left-style: ;

bdrst<Tab>      => border-right-style: ;

bdts<Tab>       => border-top-style: ;

bdbs<Tab>       => border-bottom-style: ;

Толщина линии рамки:

bdw<Tab>     => border-width: ;

bdlw<Tab>    => border-left-width: ;

bdrw<Tab>    => border-right-width: ;

bdtw<Tab>    => border-top-width: ;

bdbw<Tab>    => border-bottom-width: ;

Цвет линии рамки:

bdc<Tab>     => border-color: #000;

bdlc<Tab>    => border-left-color: #000;

bdrc<Tab>    => border-right-color: #000;

bdtc<Tab>    => border-top-color: #000;

bdbc<Tab>    => border-bottom-color: #000;

Рамки со скругленными углами:

bdrs<Tab>    => border-radius: ;

bdtlrs<Tab>  => border-top-left-radius: ;

bdtrrs<Tab>  => border-top-right-radius: ;

bdblrs<Tab>  => border-bottom-left-radius: ;

bdbrrs<Tab>  => border-bottom-right-radius: ;

Атрибут border-image:

bdi<Tab>     => border-image: url();

bdli<Tab>    => border-left-image: url();

bdri<Tab>    => border-right-image: url();

bdti<Tab>    => border-top-image: url();

bdbi<Tab>    => border-bottom-image: url();

bdtli<Tab>   => border-top-left-image: url();

bdtli:c<Tab> => border-top-left-image: continue;

bdtri<Tab>   => border-top-right-image: url();

bdtri:c<Tab> => border-top-right-image: continue;

bdbli<Tab>   => border-bottom-left-image: url();

bdbli:c<Tab> => border-bottom-left-image: continue;

bdbri<Tab>   => border-bottom-right-image: url();

bdbri:c<Tab> => border-bottom-right-image: continue;

Внешняя рамка:

ol<Tab>      => outline: ;

ols<Tab>     => outline-style: none;

ols:n<Tab>   => outline-style: none;

ols:s<Tab>   => outline-style: solid;

ols:dt<Tab>  => outline-style: dotted;

ols:ds<Tab>  => outline-style: dashed;

ols:db<Tab>  => outline-style: double;

ols:g<Tab>   => outline-style: groove;

ols:r<Tab>   => outline-style: ridge;

ols:i<Tab>   => outline-style: inset;

ols:o<Tab>   => outline-style: outset;

olc<Tab>     => outline-color: #000;

olc:i<Tab>   => outline-color: invert;

olw<Tab>     => outline-width|thin|medium|thick

olo<Tab>     => outline-offset: ;

Прочие атрибуты:

bdbk<Tab>    => border-break: close;

bdbk:c<Tab>  => border-break: close;

bdci<Tab>    => border-corner-image: url();

bdci:c<Tab>  => border-corner-image: continue;

bdf<Tab>     => border-fit: repeat;

bdf:r<Tab>   => border-fit: repeat;

bdf:c<Tab>   => border-fit: clip;

bdf:sc<Tab>  => border-fit: scale;

bdf:st<Tab>  => border-fit: stretch;

bdf:ow<Tab>  => border-fit: overwrite;

bdf:of<Tab>  => border-fit: overflow;

bdf:sp<Tab>  => border-fit: space;

bdlen<Tab>   => border-length: ;

bdlen:a<Tab> => border-length: auto;

### **Фон элемента**

Одновременное задание характеристик фона:

bg<Tab>      => background: #000;

Цвет фона:

bgc<Tab>     => background-color: #fff;

Фоновый рисунок:

bgi<Tab>     => background-image: url();

lg<Tab>      => background-image: linear-gradient();

Режим повтора фонового рисунка:

bgr<Tab>     => background-repeat: no-repeat;

bgr:n<Tab>   => background-repeat: no-repeat;

bgr:rx<Tab>  => background-repeat: repeat-x;

bgr:ry<Tab>  => background-repeat: repeat-y;

bgr:rd<Tab>  => background-repeat: round;

bgr:sp<Tab>  => background-repeat: space;

Прокрутка фонового рисунка:

bga<Tab>     => background-attachment: fixed;

bga:f<Tab>   => background-attachment: fixed;

bga:s<Tab>   => background-attachment: scroll;

Положение фонового рисунка:

bgp<Tab>     => background-position: 0 0;

bgpx<Tab>    => background-position-x: ;

bgpy<Tab>    => background-position-y: ;

Размеры фонового изображения:

bgsz<Tab>    => background-size: contain;

bgsz:ct<Tab> => background-size: contain;

bgsz:a<Tab>  => background-size: auto;

bgsz:cv<Tab> => background-size: cover;

Режим позиционирования фонового изображения:

bgo<Tab>     => background-origin: padding-box;

bgo:pb<Tab>  => background-origin: padding-box;

bgo:bb<Tab>  => background-origin: border-box;

bgo:cb<Tab>  => background-origin: content-box;

Режим заполнения для фона:

bgcp<Tab>    => background-clip: padding-box;

bgcp:pb<Tab> => background-clip: padding-box;

bgcp:cb<Tab> => background-clip: content-box;

bgcp:bb<Tab> => background-clip: border-box;

bgcp:nc<Tab> => background-clip: no-clip;

Атрибут background-break:

bgbk<Tab>    => background-break: bounding-box;

bgbk:bb<Tab> => background-break: bounding-box;

bgbk:eb<Tab> => background-break: each-box;

bgbk:c<Tab>  => background-break: continuous;

### **Списки**

Одновременное указание характеристик списка:

lis<Tab>      => list-style: ;

Вид маркера списка:

list<Tab>     => list-style-type: disc;

list:d<Tab>   => list-style-type: disc;

list:c<Tab>   => list-style-type: circle;

list:s<Tab>   => list-style-type: square;

list:dl<Tab>  => list-style-type: decimal;

list:dlz<Tab> => list-style-type: decimal-leading-zero;

list:lr<Tab>  => list-style-type: lower-roman;

list:ur<Tab>  => list-style-type: upper-roman;

Изображение в качестве маркера списка:

lisi<Tab>     => list-style-image: ;

Компактное отображение списка:

lisp<Tab>     => list-style-position: inside;

lisp:i<Tab>   => list-style-position: inside;

lisp:o<Tab>   => list-style-position: outside;

### **Таблицы**

Рамки таблицы и ячеек:

bdsp<Tab>   => border-spacing: ;

bdcl<Tab>   => border-collapse: collapse;

bdcl:c<Tab> => border-collapse: collapse;

bdcl:s<Tab> => border-collapse: separate;

Размеры таблицы:

tbl<Tab>    => table-layout: fixed;

tbl:f<Tab>  => table-layout: fixed;

tbl:a<Tab>  => table-layout: auto;

Местоположение заголовка:

cps<Tab>    => caption-side: top;    cps:t<Tab>  => caption-side: top;

cps:b<Tab>  => caption-side: bottom;

Указание характеристик ячеек:

ec<Tab>     => empty-cells: show;    ec:s<Tab>   => empty-cells: show;

ec:h<Tab>   => empty-cells: hide;

### **Вид курсора**

Вид курсора:

cur<Tab>    => cursor: pointer;      cur:p<Tab>  => cursor: pointer;

cur:a<Tab>  => cursor: auto;

cur:c<Tab>  => cursor: crosshair;

cur:d<Tab>  => cursor: default;

cur:m<Tab>  => cursor: move;

cur:t<Tab>  => cursor: text;

cur:he<Tab> => cursor: help;

cur:ha<Tab> => cursor: hand;

### **Форматирование блоков**

Управление отображением:

v<Tab>     => visibility: hidden;    v:h<Tab>   => visibility: hidden;

v:v<Tab>   => visibility: visible;

v:c<Tab>   => visibility: collapse;

Указание типа блока:

d<Tab>     => display: grid;

d:n<Tab>   => display: none;

d:b<Tab>   => display: block;

d:i<Tab>   => display: inline;

d:ib<Tab>  => display: inline-block;

d:li<Tab>  => display: list-item;

d:tb<Tab>  => display: table;

d:itb<Tab> => display: inline-table;

d:tbcp<Tab> => display: table-caption;

d:tbhg<Tab> => display: table-header-group;

d:tbrg<Tab> => display: table-row-group;

d:tbfg<Tab> => display: table-footer-group;

d:tbr<Tab>  => display: table-row;

d:tbcol<Tab> => display: table-column;

d:tbclg<Tab> => display: table-column-group;

d:tbc<Tab>  => display: table-cell;

d:f<Tab>    => display: flex;

d:if<Tab>   => display: inline-flex;

d:g<Tab>    => display: grid;

d:ig<Tab>   => display: inline-grid;

d:s<Tab>    => display: subgrid;

d:ri<Tab>   => display: run-in;

d:cp<Tab>   => display: compact;

d:rb<Tab>   => display: ruby;

d:rbt<Tab>  => display: ruby-text;

d:rbtg<Tab> => display: ruby-text-group;

Указание размеров:

bxsz<Tab>    => box-sizing: border-box;

bxsz:bb<Tab> => box-sizing: border-box;

bxsz:cb<Tab> => box-sizing: content-box;

w<Tab>       => width: ;             w:a<Tab>   => width: auto;

h<Tab>       => height: ;            h:a<Tab>   => height: auto;

miw<Tab>     => min-width: ;         mih<Tab>   => min-height: ;

maw<Tab>    => max-width: ;         mah<Tab>   => max-height: ;

Атрибут overflow:

ov<Tab>    => overflow: hidden;      ov:h<Tab>  => overflow: hidden;

ov:v<Tab>  => overflow: visible;

ov:s<Tab>  => overflow: scroll;

ov:a<Tab>  => overflow: auto;

ovx<Tab>   => overflow-x: hidden;    ovx:h<Tab> => overflow-x: hidden;

ovx:v<Tab> => overflow-x: visible;

ovx:s<Tab> => overflow-x: scroll;

ovx:a<Tab> => overflow-x: auto;

ovy<Tab>   => overflow-y: hidden;    ovy:h<Tab> => overflow-y: hidden;

ovy:v<Tab> => overflow-y: visible;

ovy:s<Tab> => overflow-y: scroll;

ovy:a<Tab> => overflow-y: auto;

ovs<Tab>   => overflow-style: scrollbar;

ovs:s<Tab> => overflow-style: scrollbar;

ovs:a<Tab> => overflow-style: auto;

ovs:p<Tab> => overflow-style: panner;

ovs:m<Tab> => overflow-style: move;

ovs:mq<Tab> => overflow-style: marquee;

Управление обтеканием:

fl<Tab>    => float: left;           fl:l<Tab>  => float: left;

fl:r<Tab>  => float: right;

fl:n<Tab>  => float: none;

cl<Tab>    => clear: both;           cl:b<Tab>  => clear: both;

cl:l<Tab>  => clear: left;

cl:r<Tab>  => clear: right;

cl:n<Tab>  => clear: none;

Позиционирование блока:

pos<Tab>   => position: relative;    pos:r<Tab> => position: relative;

pos:a<Tab> => position: absolute;

pos:f<Tab> => position: fixed;

pos:s<Tab> => position: static;

l<Tab>     => left: ;                l:a<Tab>   => left: auto;

t<Tab>     => top: ;                 t:a<Tab>   => top: auto;

r<Tab>     => right: ;               r:a<Tab>   => right: auto;

b<Tab>     => bottom: ;              b:a<Tab>   => bottom: auto;

Последовательность отображения слоев:

z<Tab>     => z-index: ;             z:a<Tab>   => z-index: auto;

### **Flex-контейнеры**

Указание типа блока:

d:f<Tab>    => display: flex;

d:if<Tab>   => display: inline-flex;

Направление выравнивания элементов внутри контейнера:

fxd<Tab>    => flex-direction: row;

fxd:r<Tab>  => flex-direction: row;

fxd:rr<Tab> => flex-direction: row-reverse;

fxd:c<Tab>  => flex-direction: column;

fxd:cr<Tab> => flex-direction: column-reverse;

Перенос на новую строку:

fxw<Tab>    => flex-wrap: nowrap;

fxw:n<Tab>  => flex-wrap: nowrap;

fxw:w<Tab>  => flex-wrap: wrap;

fxw:wv<Tab> => flex-wrap: wrap-reverse;

fxw:we<Tab> => flex-wrap: wrap-reverse;

Одновременное указание характеристик flex-контейнера:

fxf<Tab>    => flex-flow: ;

Размеры элемента:

fxb<Tab>     => flex-basis: fill;

fxb:f<Tab>   => flex-basis: fill;

fxb:a<Tab>   => flex-basis: auto;

fxb:mic<Tab> => flex-basis: min-content;

fxb:mac<Tab> => flex-basis: max-content;

fxb:fc<Tab>  => flex-basis: fit-content;

fxb:c<Tab>   => flex-basis: content;

Растяжение элементов:

fxg<Tab>    => flex-grow: ;

Сжатие элементов:

fxsh<Tab>   => flex-shrink: ;

Одновременное указание характеристик элементов:

fx<Tab>     => flex: ;

Выравнивание элементов внутри контейнера:

jc<Tab>     => justify-content: ;

jc:fs<Tab>  => justify-content: flex-start;

jc:fe<Tab>  => justify-content: flex-end;

jc:c<Tab>   => justify-content: center;

jc:sb<Tab>  => justify-content: space-between;

jc:sd<Tab>  => justify-content: space-around;

jc:sy<Tab>  => justify-content: space-evenly;

ac<Tab>     => align-content: ;

ac:sh<Tab>  => align-content: stretch;

ac:fs<Tab>  => align-content: flex-start;

ac:fe<Tab>  => align-content: flex-end;

ac:c<Tab>   => align-content: center;

ac:sb<Tab>  => align-content: space-between;

ac:sd<Tab>  => align-content: space-around;

ac:sy<Tab>  => align-content: space-evenly;

ai<Tab>     => align-items: ;

ai:sh<Tab>  => align-items: stretch;

ai:fs<Tab>  => align-items: flex-start;

ai:fe<Tab>  => align-items: flex-end;

ai:c<Tab>   => align-items: center;

ai:b<Tab>   => align-items: baseline;

as<Tab>     => align-self: ;

as:a<Tab>   => align-self: auto;

as:sh<Tab>  => align-self: stretch;

as:fs<Tab>  => align-self: flex-start;

as:fe<Tab>  => align-self: flex-end;

as:c<Tab>   => align-self: center;

as:b<Tab>   => align-self: baseline;

Порядок следования элементов внутри контейнера:

ord<Tab>    => order: ;

### **CSS Grid**

Указание типа блока:

d<Tab>     => display: grid;         d:g<Tab>   => display: grid;

d:ig<Tab>  => display: inline-grid;

d:s<Tab>   => display: subgrid;

Описание строк и столбцов:

gt<Tab>    => grid-template: ;

gtc<Tab>   => grid-template-columns: repeat();

gtc:r<Tab> => grid-template-columns: repeat();

gtc:m<Tab> => grid-template-columns: minmax();

gtc:a<Tab> => grid-template-columns: auto;

gtr<Tab>   => grid-template-rows: repeat();

gtr:m<Tab> => grid-template-rows: minmax();

gtr:a<Tab> => grid-template-rows: auto;

Автоматическое размещение элементов внутри сетки:

gaf<Tab>   => grid-auto-flow: row;

gaf:r<Tab> => grid-auto-flow: row;

gaf:c<Tab> => grid-auto-flow: column;

gaf:d<Tab> => grid-auto-flow: dense;

gaf:i<Tab> => grid-auto-flow: initial;

gaf:ih<Tab> => grid-auto-flow: inherit;

gaf:u<Tab> => grid-auto-flow: unset;

Добавление элементов в ячейки сетки:

gcs<Tab>   => grid-column-start: ;

gce<Tab>   => grid-column-end: ;

gc<Tab>    => grid-column: ;

grs<Tab>   => grid-row-start: ;

gre<Tab>   => grid-row-end: ;

gr<Tab>    => grid-row: ;

ga<Tab>    => grid-area: ;

Размеры неявных ячеек:

gac<Tab>   => grid-auto-columns: auto;

gac:m<Tab> => grid-auto-columns: minmax();

gar<Tab>   => grid-auto-rows: auto;

gar:m<Tab> => grid-auto-rows: minmax();

Расстояние между ячейками:

gcg<Tab>   => grid-column-gap: ;

grg<Tab>   => grid-row-gap: ;

gg<Tab>    => grid-gap: ;

Имена элементов:

ga<Tab>    => grid-area: ;

gta<Tab>   => grid-template-areas: ;

gt<Tab>    => grid-template: ;

Одновременное указание характеристик контейнера:

gd<Tab>    => grid: ;

Выравнивание сетки внутри контейнера:

jc<Tab>     => justify-content: start;

jc:s<Tab>   => justify-content: start;

jc:e<Tab>   => justify-content: end;

jc:c<Tab>   => justify-content: center;

jc:sb<Tab>  => justify-content: space-between;

jc:sd<Tab>  => justify-content: space-around;

jc:sy<Tab>  => justify-content: space-evenly;

ac<Tab>     => align-content: start;

ac:s<Tab>   => align-content: start;

ac:e<Tab>   => align-content: end;

ac:c<Tab>   => align-content: center;

ac:sb<Tab>  => align-content: space-between;

ac:sd<Tab>  => align-content: space-around;

ac:sy<Tab>  => align-content: space-evenly;

Выравнивание элемента внутри ячейки:

js<Tab>     => justify-self: start;

js:s<Tab>   => justify-self: start;

js:e<Tab>   => justify-self: end;

js:c<Tab>   => justify-self: center;

js:sh<Tab>  => justify-self: stretch;

ji<Tab>     => justify-items: start;

ji:s<Tab>   => justify-items: start;

ji:e<Tab>   => justify-items: end;

ji:c<Tab>   => justify-items: center;

ji:sh<Tab>  => justify-items: stretch;

as<Tab>     => align-self: start;

as:s<Tab>   => align-self: start;

as:e<Tab>   => align-self: end;

as:sh<Tab>  => align-self: stretch;

as:c<Tab>   => align-self: center;

as:a<Tab>   => align-self: auto;

ai<Tab>     => align-items: start;

ai:s<Tab>   => align-items: start;

ai:e<Tab>   => align-items: end;

ai:sh<Tab>  => align-items: stretch;

ai:c<Tab>   => align-items: center;

### **Многоколоночный текст**

Количество колонок:

colm<Tab>    => columns: ;

colmc<Tab>   => column-count: ;

colmc:a<Tab> => column-count: auto;

Размеры колонок:

colmw<Tab>   => column-width: ;

colmw:a<Tab> => column-width: auto;

colmf<Tab>   => column-fill: ;

colmf:a<Tab> => column-fill: auto;

Расстояние между колонками:

colmg<Tab>   => column-gap: ;

Линии между колонками:

colmr<Tab>   => column-rule: ;

colmrs<Tab>  => column-rule-style: ;

colmrc<Tab>  => column-rule-color: ;

colmrw<Tab>  => column-rule-width: ;

colms<Tab>   => column-span: ;

### **Эффекты**

Изменение прозрачности:

op<Tab>     => opacity: ;

Создание тени для текста:

tsh<Tab>    => text-shadow: hoff voff blur #000;

Создание тени для блока:

bxsh<Tab>   => box-shadow: inset hoff voff blur #000;

bxsh:n<Tab> => box-shadow: none;

### **Анимация с двумя состояниями**

Одновременное задание всех параметров анимации:

trs<Tab>    => transition: prop time;

Продолжительность анимации:

trsdu<Tab>  => transition-duration: time;

Задержка перед началом анимации:

trsde<Tab>  => transition-delay: time;

Задание анимируемых атрибутов:

trsp<Tab>   => transition-property: prop;

Закон анимации:

trstf<Tab>  => transition-timing-function: fn;

### **Анимация с несколькими состояниями**

Шкала времени:

@kf<Tab>    =>

   @keyframes identifier {

   }

Одновременное задание всех параметров анимации:

anim<Tab>        =>

   animation: name duration timing-function delay iteration-count

   direction fill-mode;

Указание названия шкалы времени:

animn<Tab>       => animation-name: ;

Продолжительность анимации:

animdur<Tab>     => animation-duration: 0s;

Задержка перед началом анимации:

animdel<Tab>     => animation-delay: time;

Закон анимации:

animtf<Tab>      => animation-timing-function: linear;

animtf:l<Tab>    => animation-timing-function: linear;

animtf:e<Tab>    => animation-timing-function: ease;

animtf:ei<Tab>   => animation-timing-function: ease-in;

animtf:eo<Tab>   => animation-timing-function: ease-out;

animtf:eio<Tab>  => animation-timing-function: ease-in-out;

animtf:cb<Tab>   =>

   animation-timing-function: cubic-bezier(0.1, 0.7, 1.0, 1.0);

Количество повторений анимации:

animic<Tab>      => animation-iteration-count: 1;

animic:i<Tab>    => animation-iteration-count: infinite;

Направление анимации:

animdir<Tab>     => animation-direction: normal;

animdir:n<Tab>   => animation-direction: normal;

animdir:r<Tab>   => animation-direction: reverse;

animdir:a<Tab>   => animation-direction: alternate;

animdir:arv<Tab> => animation-direction: alternate-reverse;

Текущее состояние анимации:

animps<Tab>      => animation-play-state: running;

animps:r<Tab>    => animation-play-state: running;

animps:p<Tab>    => animation-play-state: paused;

Состояние элемента до начала анимации и после ее завершения:

animfm<Tab>      => animation-fill-mode: both;

animfm:bh<Tab>   => animation-fill-mode: both;

animfm:bt<Tab>   => animation-fill-mode: both;

animfm:bs<Tab>   => animation-fill-mode: backwards;

animfm:f<Tab>    => animation-fill-mode: forwards;

### **Двумерные трансформации**

Атрибут transform:

trf<Tab>         => transform: ;

Смещение:

trf:tx<Tab>      => transform: translateX(x);

trf:ty<Tab>      => transform: translateY(y);

trf:t<Tab>       => transform: translate(x, y);

Изменение масштаба:

trf:scx<Tab>     => transform: scaleX(x);

trf:scy<Tab>     => transform: scaleY(y);

trf:sc<Tab>      => transform: scale(x, y);

Наклон:

trf:skx<Tab>     => transform: skewX(angle);

trf:sky<Tab>     => transform: skewY(angle);

Вращение:

trf:r<Tab>       => transform: rotate(angle);

Позиционирование точки начала координат для двумерных трансформаций:

trfo<Tab>        => transform-origin: ;

### **Трехмерные трансформации**

Выполнение трехмерных трансформаций:

trf:tz<Tab>      => transform: translateZ(z);

trf:td<Tab>      => transform: translate3d(tx, ty, tz);

trf:scz<Tab>     => transform: scaleZ(z);

trf:scd<Tab>     => transform: scale3d(x, y, z);

trf:rx<Tab>      => transform: rotateX(angle);

trf:ry<Tab>      => transform: rotateY(angle);

trf:rz<Tab>      => transform: rotateZ(angle);

Скрытие обратной стороны элемента:

bfv<Tab>         => backface-visibility: hidden;

bfv:h<Tab>       => backface-visibility: hidden;

bfv:v<Tab>       => backface-visibility: visible;

Режим проецирования элементов на контейнер:

trfs<Tab>        => transform-style: preserve-3d;

Позиционирование точки начала координат для трехмерных трансформаций:

trfo<Tab>        => transform-origin: ;

### **Прочие атрибуты и правила**

Правила:

@i<Tab>      => @import url();

@import<Tab> => @import url();

@m<Tab>      =>

   @media screen {

   }

@media<Tab>  =>

   @media screen {

   }

Прочие атрибуты:

ap<Tab>      => appearance: none;

cp<Tab>      => clip: auto;              cp:a<Tab>    => clip: auto;

cp:r<Tab>    => clip: rect(top right bottom left);

mar<Tab>     => max-resolution: ;

mir<Tab>     => min-resolution: ;

ori<Tab>     => orientation: landscape;

ori:l<Tab>   => orientation: landscape;

ori:p<Tab>   => orientation: portrait;

orp<Tab>     => orphans: ;

pgbb<Tab>    => page-break-before: auto;

pgbb:au<Tab> => page-break-before: auto;

pgbb:al<Tab> => page-break-before: always;

pgbb:l<Tab>  => page-break-before: left;

pgbb:r<Tab>  => page-break-before: right;

pgba<Tab>    => page-break-after: auto;

pgba:au<Tab> => page-break-after: auto;

pgba:al<Tab> => page-break-after: always;

pgba:l<Tab>  => page-break-after: left;

pgba:r<Tab>  => page-break-after: right;

pgbi<Tab>    => page-break-inside: auto;

pgbi:au<Tab> => page-break-inside: auto;

pgbi:av<Tab> => page-break-inside: avoid;

rsz<Tab>     => resize: none;            rsz:n<Tab>   => resize: none;

rsz:b<Tab>   => resize: both;

rsz:h<Tab>   => resize: horizontal;

rsz:v<Tab>   => resize: vertical;

whsc<Tab>    => white-space-collapse: normal;

whsc:n<Tab>  => white-space-collapse: normal;

whsc:k<Tab>  => white-space-collapse: keep-all;

whsc:l<Tab>  => white-space-collapse: loose;

whsc:bs<Tab> => white-space-collapse: break-strict;

whsc:ba<Tab> => white-space-collapse: break-all;

wm<Tab>      => writing-mode: lr-tb;

wm:btl<Tab>  => writing-mode: bt-lr;

wm:btr<Tab>  => writing-mode: bt-rl;

wm:lrb<Tab>  => writing-mode: lr-bt;

wm:lrt<Tab>  => writing-mode: lr-tb;

wm:rlb<Tab>  => writing-mode: rl-bt;

wm:rlt<Tab>  => writing-mode: rl-tb;

wm:tbl<Tab>  => writing-mode: tb-lr;

wm:tbr<Tab>  => writing-mode: tb-rl;

us<Tab>      => user-select: none;

zom<Tab>     => zoom: 1;

### **Ввод числового значения атрибута**

После комбинации символов допускается указание числового значения, которое станет значением атрибута стилей с добавлением (при необходимости) единицы изменения px (для целочисленных значений) или em (для дробных значений):

m10<Tab>   => margin: 10px;

p2.5<Tab>  => padding: 2.5em;

lh1.5<Tab> => line-height: 1.5;

Если нужно указать несколько значений, то они задаются через дефис:

m10-20<Tab> => margin: 10px 20px;

Для ввода отрицательных значений следует указать два дефиса:

p-15--5<Tab> => padding: -15px -5px;

После значения можно указать единицу измерения. В этом случае разделять значения дефисом не нужно:

p10px20px<Tab> => padding: 10px 20px;

Вместо реальной единицы измерения допускается указание псевдонимов: p для %, e для em, x для ex:

p10p20e30x5<Tab> => padding: 10% 20em 30ex 5px;

Для вставки значения !important достаточно ввести восклицательный знак:

m20!<Tab>        => margin: 20px !important;

В качестве значения для цвета можно указать одну, две, три или шесть шестнадцатеричных цифр после символа #. Если цифры не заданы, то получим черный цвет:

c#<Tab>          => color: #000000;

c#f<Tab>         => color: #ffffff;

c#e8<Tab>        => color: #e8e8e8;

c#ec0<Tab>       => color: #eecc00;

c#28a745<Tab>    => color: #28a745;

### **Добавление вендорных префиксов**

Если перед комбинацией символов указать дефис, то будут сгенерированы названия атрибутов с вендорными префиксами:

-bdrs<Tab> =>

   -webkit-border-radius: ;

   -moz-border-radius: ;

   border-radius: ;

*На заметку*

Пакет Autoprefixer позволяет автоматически добавить вендорные префиксы в зависимости от выбранных Web-браузеров. Поэтому его использование является более правильным решением, чем генерация всех префиксов.

## Плагины