

Урок 2. Методичка

Управляем компьютером.
Основные теги.



Содержание

- [Как компьютер понимает человека;](#)
- [Псевдотеги;](#)
- [Структура HTML-страницы;](#)
- [Основные html теги;](#)
- [Сложные вложенные структуры;](#)
- [Вывод изображений;](#)
- [Сохранение кода прямо в Git через Visual Code.](#)

Как компьютер понимает человека

Современные компьютеры не настолько совершенны, чтобы понимать человека с полуслова. Хотя работа над этим идет вовсю, привет Сири и Алисе!

Сейчас, чтобы компьютер сделал то что мы хотим, выполнил программу, отобразил страничку сайта, для него нужно написать программу – текст.

Создавать странички с текстом, в нужной кодировке, и даже размещать их в Интернете, мы научились на предыдущем уроке. Теперь будем учиться управлять компьютером с помощью текста.



Строго говоря, HTML это язык разметки документов, а css язык описания внешнего вида. Однако, язык расширяется, и у него все больше возможностей, вот например игра на чистом html-css:

<https://nisnom.com/veb-razrabotki/igra-stacker-na-chistom-css3-i-html5-bez-ispolzovaniya-js/>

Это не значит что нужно делать именно игры, все нужно использовать по назначению, это значит у нас больше возможностей, который можно использовать для создания по настоящему крутых страниц и сайтов!

Как компьютер понимает человека

Выясним, каким же образом нужно набирать текст, чтобы компьютер **правильно** его понял.


Обычно, мы набираем некоторые блоки текста, так как их правильно набирать?

Попробуйте набрать 2 блока с текстом в новом html-файле и посмотреть, как он его отобразит.



Человек без проблем визуально воспринимает эти 4 блока с текстом, а вот если мы откроем через браузер и посмотрим как отобразятся блоки, то увидим, что нормально браузером читаются только третий и четвертый блок!

1	Первый блок	Второй блок
2	с текстом	с текстом
3		
4		
5	Третий блок	
6	с текстом	
7		
8	Четвертый блок	
9	с текстом	



Первый блок Второй блок с текстом с текстом **Третий блок с текстом Четвертый блок с текстом**

Псевдотег

Получается, что компьютер читает текст строго слева-направо и сверху-вниз, как и мы читаем книгу, поэтому, именно так и нужно набирать код. Однако, как же он будет воспринимать блоки текста? Это мы визуально можем выделить его, а компьютеру надо помочь! Нужно выделить то что нам нужно, специальными символами – **тегами**.

Вы уже знаете из математики, как можно в выражении сделать акцент и выделить что-то, скобками, например $(2 + 3) * 5$

В этом выражении мы выделяем $2+3$ и говорим, что сначала нужно вычислить сумму, а потом уже умножение.

В HTML вместо скобок используются текст, заключенный в такие скобки `<>`, например `<div>` или `<p>`, а для закрывающей скобки используется прямой слеш `</div>` или `</p>`.

Давайте перепишем выражение $(2 + 3) * 5$ по правилам HTML через псевдотег:
`<скобка> 2 + 3 </скобка> * 5`



Обратите внимание, что несмотря на то, что мы поставили множество пробелов и переносов, браузер все отобразил в одну строчку и превратил все пробелы в один. Так удобно оформлять код так, чтобы он легче воспринимался человеком.

Псевдотеги

Как и в математике, скобки можно вкладывать друг в друга:

$((2 + 3) * (4 + 5))$

И перепишем псевдотегом-скобкой:

`<скобка> <скобка> 2 + 3 </скобка> * <скобка> 4 + 5 </скобка> </скобка>`

Здесь, число открывающих скобок должно быть равно закрывающим и очень легко ошибиться набирая такой код. Чтобы было легче визуальнo воспринимать такую конструкцию можно пожертвовать компактностью записи и переписать для удобства чтения.



`<скобка>` не настоящий тег, это псевдотег придуманный нами, чтобы было понятнее читать код по началу.

Псевдотеги

Такая запись с точки зрения компьютера точно такая же как и в одну строчку. Однако согласитесь, визуально человеку гораздо удобнее воспринимать такой код.

$((2 + 3) * (4 + 5))$

`<скобка> <скобка> 2 + 3 </скобка> * <скобка> 4 + 5 </скобка> </скобка>`

```
1 ▼ <скобка>
2 ▼   <скобка>
3       2 + 3
4       </скобка>
5       *
6 ▼   <скобка>
7       4 + 5
8       </скобка>
9   </скобка>
```



Текстовый редактор нам помогает воспринимать такие структуры тем, что позволяет их сворачивать вот такой стрелочкой. Если код не будет помещаться на экране, его можно свернуть и увидеть где там что открывается и закрывается.

Псевдотеги

Попробуем при помощи псевдотегов, сконструировать другую конструкцию – человека. Для этого представим довольно грубо, что вот у нас есть человек, и он у нас будет состоять из головы и тела.



Наберите вот такую конструкцию и представьте ее для наглядности в виде такой схемы:

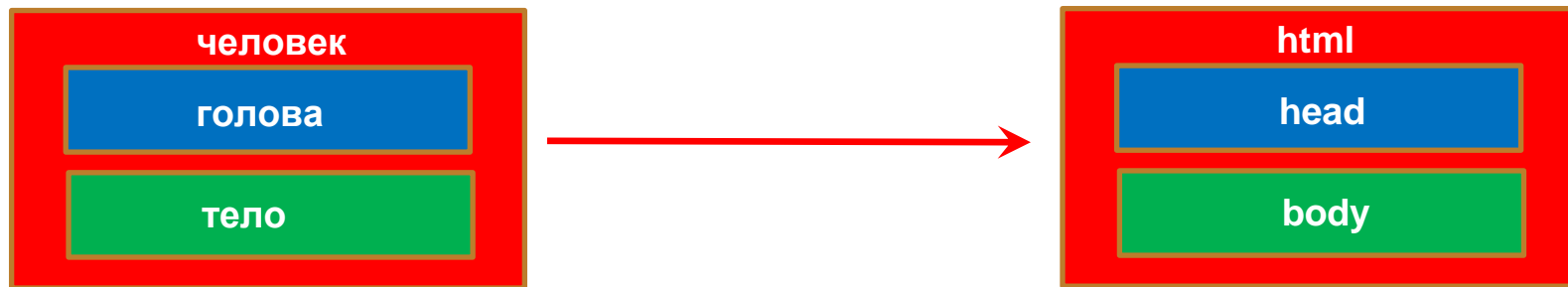


```
1 ▼ <человек>
2
3     <голова>
4     </голова>
5
6     <тело>
7     </тело>
8
9 </человек>
```



Под конструированием понимается процесс перевода каких то вещей из реального или не реального мира в язык, понятный компьютеру.

Структура HTML-страницы



Реальная структура html-страницы очень похожа на человека, посмотрите, у нее тоже есть голова и тело, и это ее обязательные части!

Переведем эту структуру обратно в код и мы получим настоящую минимальную структуру документа.



Под конструированием понимается процесс перевода каких то вещей из реального или не реального мира в язык, понятный компьютеру.

Структура HTML-страницы

```
1  <!DOCTYPE html>
2  <html>
3
4      <head>
5      </head>
6
7      <body>
8      </body>
9
10 </html>
```

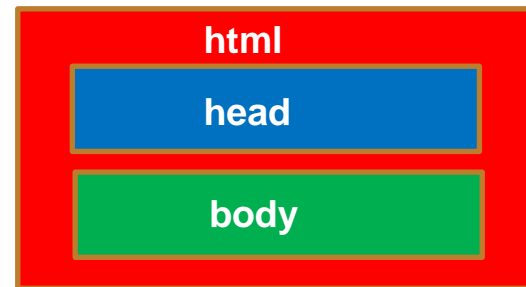
<!DOCTYPE html> - тип документа

<html> - содержание документа (человек)

<head></head> - голова документа

<body>Текст</body> - тело документа

</html>



Итак, любая html-страница должна содержать такую структуру, здесь мы видим строчку `<!DOCTYPE html>` и у нее нет закрывающего тега. Это специальный не парный тег, указывающий для браузера, что тип страницы будет HTML5, это нужно для того, чтобы браузер верно понимал дальнейший код.

Структура HTML-страницы

Вы уже знаете как быстро в редакторе построить такую структуру при помощи плагина emmet.

Нужно набрать ! нажать tab, мы сразу получим заготовку страницы.

Однако мы тут видим еще теги и надписи:

lang = "en" указывает на язык страницы

meta указывает на кодировку текста файла

title содержит текст, выводимый в заголовке браузера

body содержит текст, выводимый собственно в браузере, может содержать другие теги.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6  </head>
7  <body>
8
9  </body>
10 </html>
```



Внутри тега **title** нельзя размещать другие теги, только текст!

Основные html теги

Посмотрим какие еще есть теги для текста, и что они делают:

`<p>`Абзац с текстом`</p>`

Перевод `
` строки (не закрывается)

`<center>`По центру`</ center>` (тег устарел, но пока давайте попользуемся)

``**Жирный текст**``

`<i>`*Курсив*`</i>`

Эти теги создают строчные элементы, они будут располагаться в одной строчке (кроме br)



Обратите внимание, что тег `
` не закрывается! Он не парный. И решает проблему с переносом строк, можно ставить перенос там где нам нужно.

`<p>` абзац автоматически формирует в конце перенос, так что текст после него гарантированно будет с новой строчки.

Основные html теги

Посмотрим на абзац в работе, наберите `p>lorem5` и нажмите таб, увидим один абзац, нажмите `ctrl+D` и строки станет две!

```
<body>  
  <p>Lorem ipsum dolor sit amet.</p>  
  <p>Lorem ipsum dolor sit amet.</p>  
</body>
```



Все это делаем внутри `body`, и чтобы увидеть результат откроем в браузере и увидим:

Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet.



`p>lorem5` читается так, создай тег `p` и внутри него размести «рыбный» (случайный) текст из 5 слов. `ctrl+D` работает не всегда, иногда эту комбинацию нужно настроить. Плагин **open in browser** позволяет быстро открыть в браузере по `alt+B`

Основные html теги

Выделим жирным одно слово внутри абзаца, создадим вложенную структуру. Для этого оберните выбранное слово тегом ``, не забудьте его закрыть ``



```
<p>Lorem <b>ipsum</b> dolor sit amet.</p>  
<p>Lorem ipsum dolor sit amet.</p>|
```

И после сохранения мы увидим результат – выделенное слово внутри абзаца.

Lorem **ipsum** dolor sit amet.

Lorem ipsum dolor sit amet.



Можно настроить редактор на быстрое оборачивание текста любым тегом. Для этого нажмите F1 и в поле сверху введите **emmet wrap** чтобы найти нужное свойство (смотри на картинке ниже), потом нажмите на шестеренку справа, нажмите на экране на плюсики и введите нужную быструю комбинацию, например `ctrl+shift+A`, и нажмите enter. Теперь можно быстро оборачивать любым тегом текст, нажав на эту комбинацию и введя нужный тег.

Emmet: Wrap Individual Lines with Abbreviation

Ctrl + Shift + A recently used 

Сложные вложенные структуры

Итак мы создали структуру где есть html-документ, внутри него есть body-тело, внутри тела тег `p`, внутри которого тег `b`! Получилось 4 уровня вложения! Создание любой html странички и является созданием такой вложенной структуры, только гораздо сложнее.

Чтобы научиться безошибочно составлять такие структуры давайте еще потренируемся и сделаем структуру человека с головой и телом на реальных тегах.

Для этого нам понадобится еще один тег **<div>**, он парный и не несет какого-то определенного значения как тег абзаца, хотя и чем-то похож на него. Это просто универсальная обертка любого блока.



Тег `<div>` создает блочный элемент, который будет размещаться друг под другом прямо как абзац, об этом будем еще говорить на следующих уроках. Внутри блочных элементов можно использовать строчные, но не наоборот!

Сложные вложенные структуры

Создаем нашего человека в редакторе, активно пользуясь emmet и делая отступы, чтобы четко визуально видеть что во что вложено.

Однако, если мы сохраним и посмотрим на эту структуру, мы ее не увидим, поэтому, чтобы визуализировать эту структуру, возьмите заготовку из материалов example1 и внутри этого html – файла разместите вашего человека, сохраните и полюбуйте результатом:



```
<body>
  <div>
    Человек
    <div>
      Голова
    </div>
    <div>
      Тело
    </div>
  </div>
</body>
```

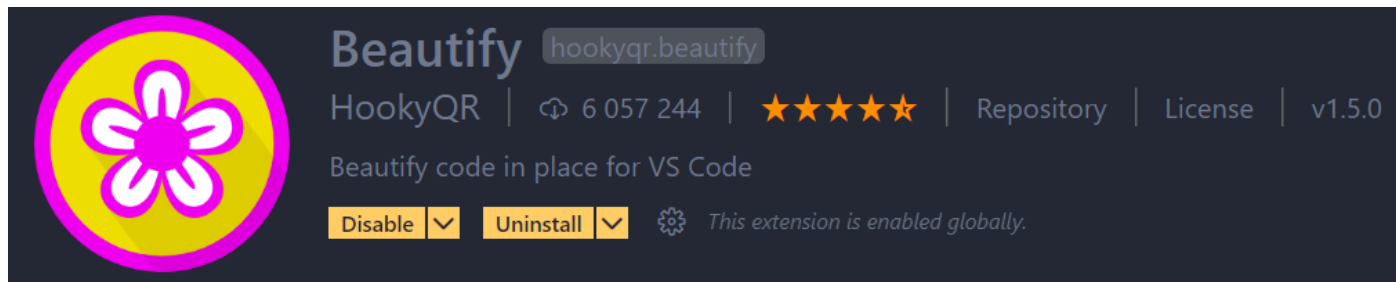


Как это работает мы так же разберемся позже, но при желании можно полюбопытствовать и посмотреть на код, который делает такой вид.

Сложные вложенные структуры

Чтобы еще ускорить набор кода, поставим еще одно расширение для VSC, под названием **Beautify**, которое автоматически будет расставлять отступы для вашей вложенной конструкции.

Найдите в установке плагинов Beautify и установите его, для выравнивания текста нажимайте F1 и в окошке сверху выберите beautify file.



В других редакторах тоже можно найти и установить такой же плагин. Также можно назначить горячую клавишу, чтобы каждый раз не нажимать F1. Есть плагины, которые выравнивание делают автоматически, но такие не рекомендую.

Вывод изображений

Теперь мы можем оформлять страницы с текстом, формировать абзацы, выделять текст, однако, современные страницы это не только текст, а еще и изображения. Изучим тег для вывода изображения и посмотрим на его синтаксис:

```
<img src= "имя файла" alt= "текст">
```

Как мы видим это не парный тег, и содержит **имя файла**-картинки, которую нужно отобразить, и **текст**, который будет выведен, если такого файла не найдется.

Скорей всего вы захотите изменить размер изображения, тогда добавьте ему ширину в пикселях.



```
<img src= "имя файла" alt= "текст" width="200">
```



Зеленым текстом обозначен собственно текст, **оранжевым** атрибут тега (свойство), а **черным** значение свойства. Об атрибутах поговорим подробнее позже.

Вывод изображений

Выведем изображение котика на нашей страничке, для этого скачайте изображение (любое), и сохраните его в вашу папку с проектом, там где файл html, это важно! Еще переименуйте файл в более удобное название, т.к. скорей всего оно будет очень длинное и не понятное. Вот так:

 cat.jpg	03.07.2020 19:22	Файл "JPG"	55 КБ
 index.html	03.07.2020 19:22	Chrome HTML Do...	1 КБ

```
Мой котик:<br>
<br>
2020
```



Мой котик:



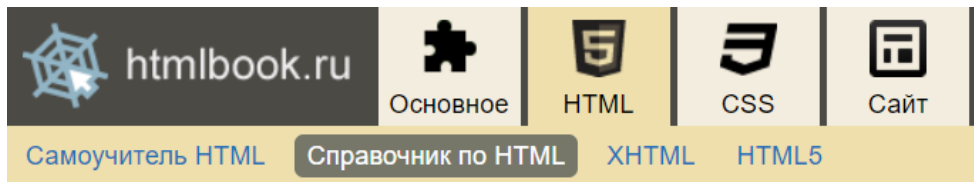
2020



Обратите внимание, если не использовать тег `br`, то и текст и изображение будут выводиться в одной строке, значит `img` тоже строчный элемент. Есть еще одни устаревший парный тег `<center>`, который позволяет выровнять что-то.

Справочник

В Интернете вы можете найти множество сайтов по HTML, попробуйте сначала этот сайт <http://htmlbook.ru/>



Здесь вы найдете описание, синтаксис и примеры использования всех тегов!

Нажмите Справочник по HTML и справа увидите все типы тегов.

Типы тегов

- [HTML5](#)
- [Блочные элементы](#)
- [Строчные элементы](#)
- [Универсальные элементы](#)
- [Нестандартные теги](#)
- [Осуждаемые теги](#)
- [Видео](#)
- [Документ](#)
- [Звук](#)
- [Изображения](#)
- [Объекты](#)
- [Скрипты](#)
- [Списки](#)
- [Ссылки](#)
- [Таблицы](#)
- [Текст](#)
- [Форматирование](#)
- [Формы](#)
- [Фреймы](#)

Сохранение кода прямо в Git через Visual Code.

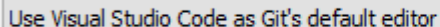
Чтобы выложить сайт на git хостинг, не обязательно каждый раз заходить на сайт и вручную загружать все файлы, это очень не удобно. Можно прямо в редакторе открыть код на git и сохранять прямо на сайт так, что сайт на хостинге будет обновляться сразу же.

Для этого сначала нужно установить git на ваш компьютер, зайдите на сайт <https://git-scm.com/downloads> и выберите ваш вариант и установите его:



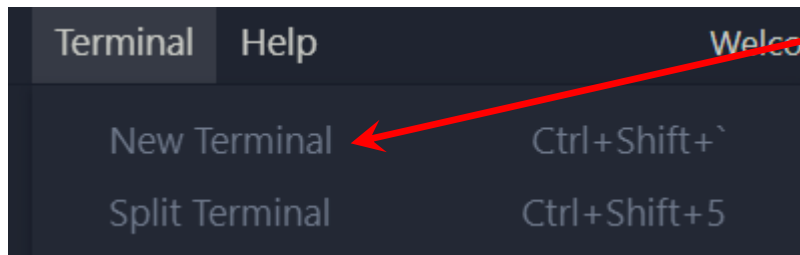
Сохранение кода прямо в Git через Visual Code.

При установке можно не трогать никакие галочки, разве что выбрать редактор VSC если он у вас установлен, но это не обязательно.



Use Visual Studio Code as Git's default editor

Далее нужно открыть консоль или терминал, в VSC это можно сделать через меню.



Как установите git, запускать его не нужно, вся работа будет через консоль или редактор кода.

Сохранение кода прямо в Git через Visual Code.

Проверим, установился ли гит и его версию, для этого в консоли наберите `git --version`, должен быть получен результат как на картинке:

```
PS C:\OpenServer\domains\embersong.github.io> git --version
git version 2.30.0.windows.2
PS C:\OpenServer\domains\embersong.github.io> 
```

Введите еще 2 команды, чтобы настроить имя пользователя и его почту, чтобы гит знал кто делает изменения.

```
git config user.name "Ember" --global
git config user.email "evabay@mail.ru" --global
```



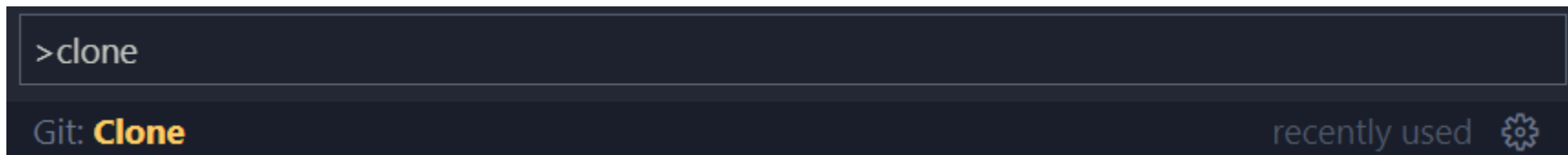
Если при установке git у вас был запущен редактор VSC, то команда git может сразу там не работать, перезапустите редактор кода.

Сохранение кода прямо в Git через Visual Code.

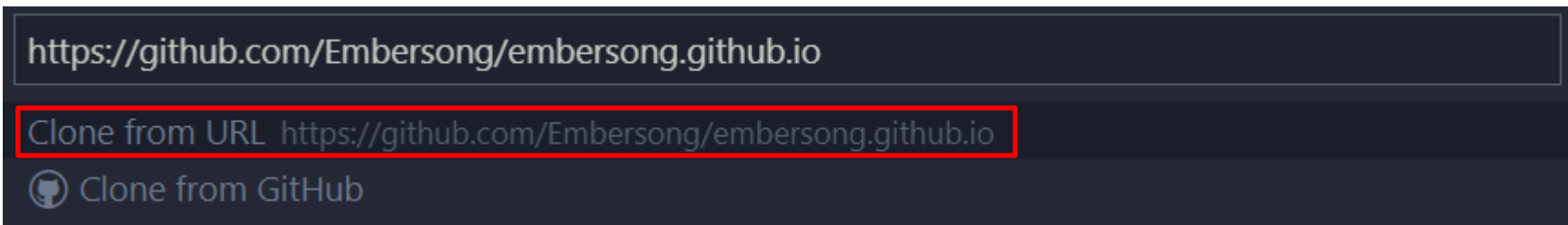
Теперь вы можете открыть ваш проект прямо из гит. Для этого откройте гит и скопируйте ссылку на ваш проект, должно получиться что то типа такого:

<https://github.com/Embersong/embersong.github.io>

Затем в редакторе нажмите F1 и наберите в поле сверху clone, и выберите ниже Git: Clone

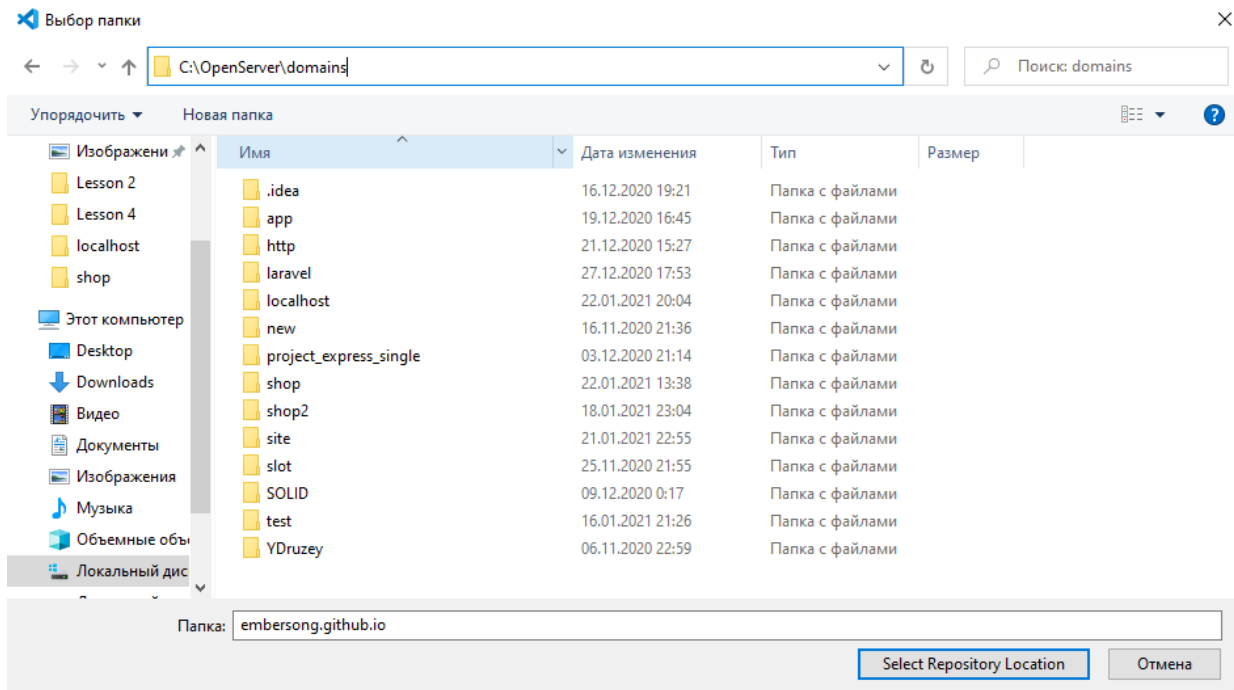


Затем в поле сверху вставьте url вашего git хранилища и нажмите Clone from URL



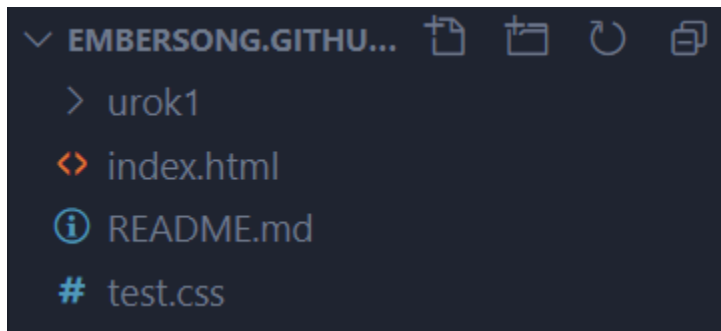
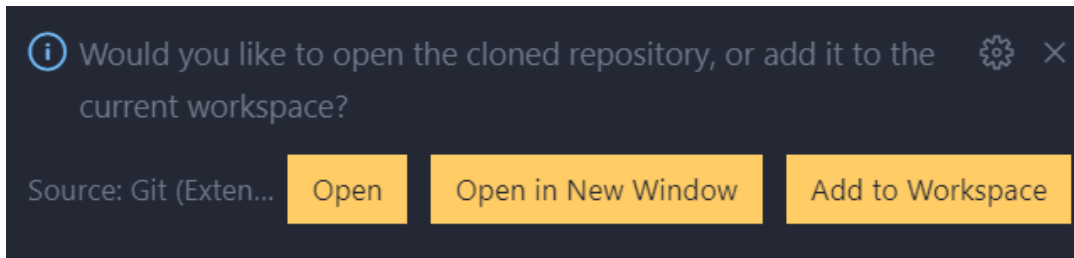
Сохранение кода прямо в Git через Visual Code.

Выберите папку, где будет храниться проект из git.



Сохранение кода прямо в Git через Visual Code.

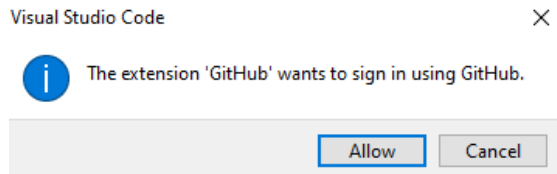
После выбора папки будет предложено открыть проект, выберите Open справа внизу



И ваш проект будет открыт в редакторе прямо из гит.

Сохранение кода прямо в Git через Visual Code.

При первом входе может понадобится авторизация в гит, вы увидите такое окошко, нажмите Allow



Вас перекинет на сайт git, там нужно нажать зеленую Continue:



Authorize Visual Studio Code to access GitHub

If you initiated this authorization from Visual Studio Code, click 'Continue' to authorize access to GitHub

Continue

[Do not authorize](#)

Сохранение кода прямо в Git через Visual Code.

Далее нажмите Authorize github для разрешения доступа из редактора, и введите пароль от git:



Authorize GitHub for VSCode



GitHub for VSCode by [github](#)
wants to access your **Embersong** account



Repositories
Public and private



Workflow
Update GitHub Action Workflow files.



Cancel

Authorize github

Authorizing will redirect to
<https://vscode-auth.github.com>



Confirm password to continue

Password

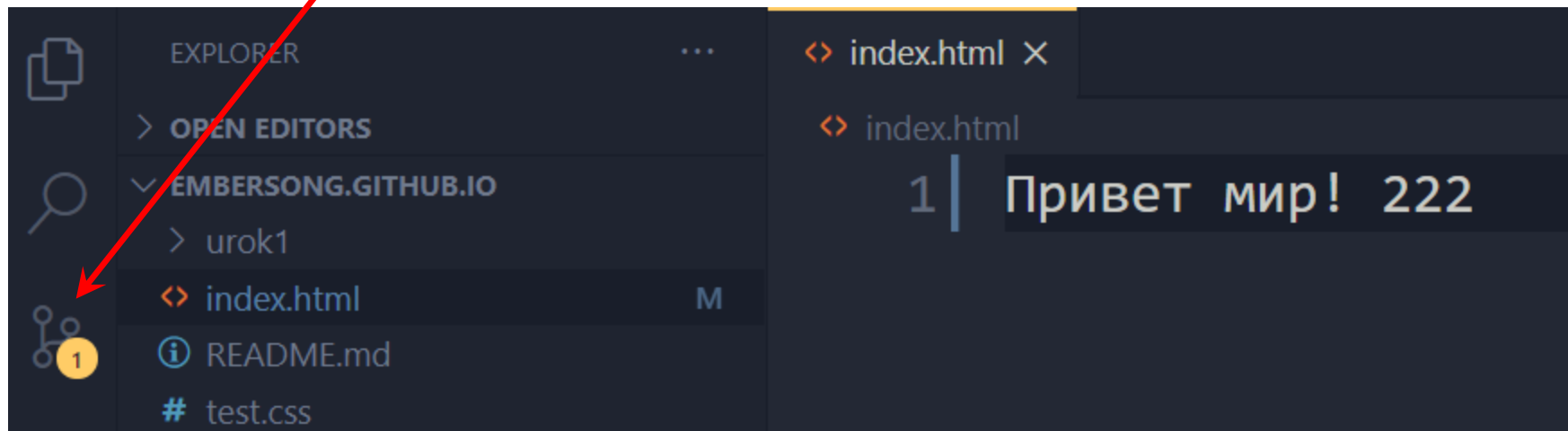
[Forgot password?](#)

Confirm password

Tip: You are entering **sudo mode**. We won't ask for your password again for a few hours.

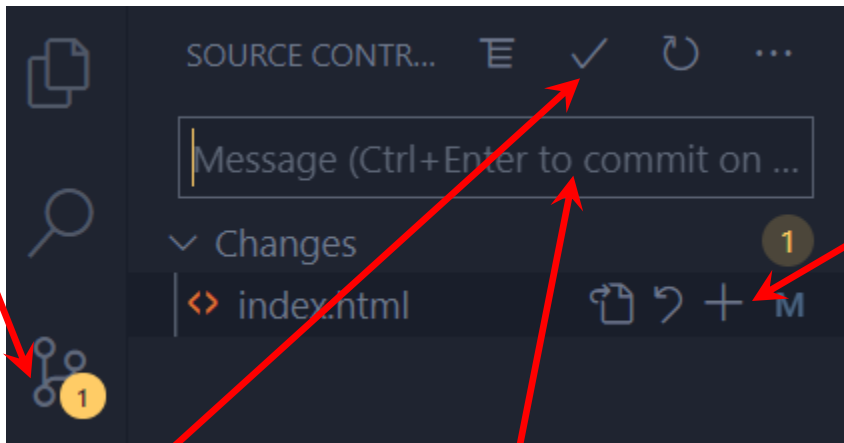
Сохранение кода прямо в Git через Visual Code.

Далее вы работаете со своим проектом, сохраняете файлы на ваш компьютер, и когда что то меняете, ГИТ отслеживает это.



Сохранение кода прямо в Git через Visual Code.

Перед сохранением результатов на сайт, зайдите в гит-вкладку слева в редакторе.



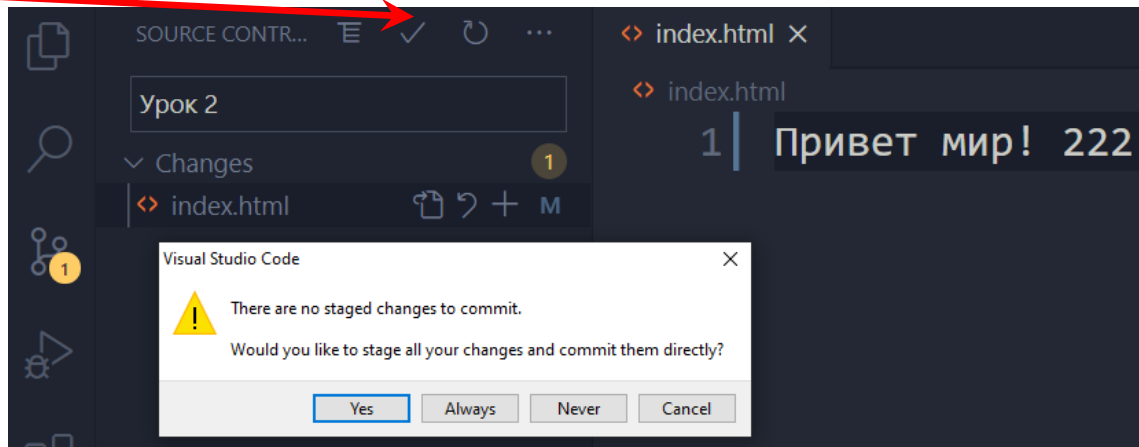
А этой кнопкой фиксируете
все изменения

Здесь напишите что было сделано, например
«решение урока 2»

Этой кнопкой вы
добавляете новый файл в
ГИТ.

Сохранение кода прямо в Git через Visual Code.

Теперь все готово для фиксации изменений, нажмите галочку и выберите yes.



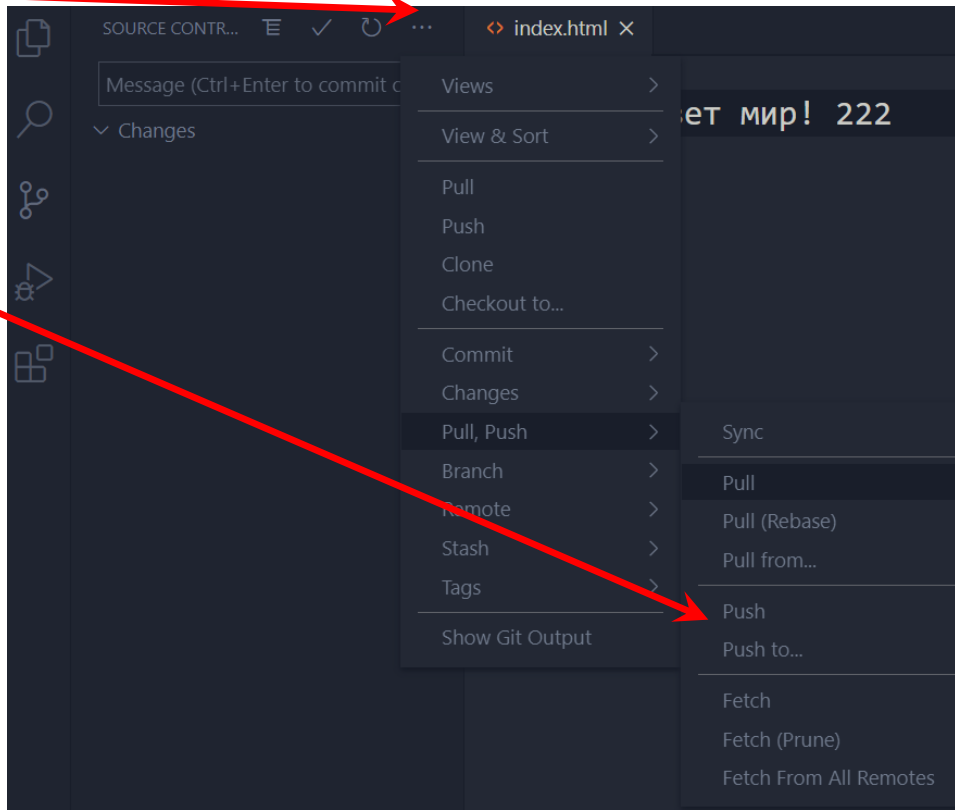
Но данные еще на сервер загружены не будут! Для этого из надо выгрузить в git, сделать Push.

Сохранение кода прямо в Git через Visual Code.

Чтобы сделать Push нажмите многоточие и выберите Push



По адресу хостинга, например <https://embersong.github.io/> Изменения появятся не сразу, может полминуты нужно будет подождать.

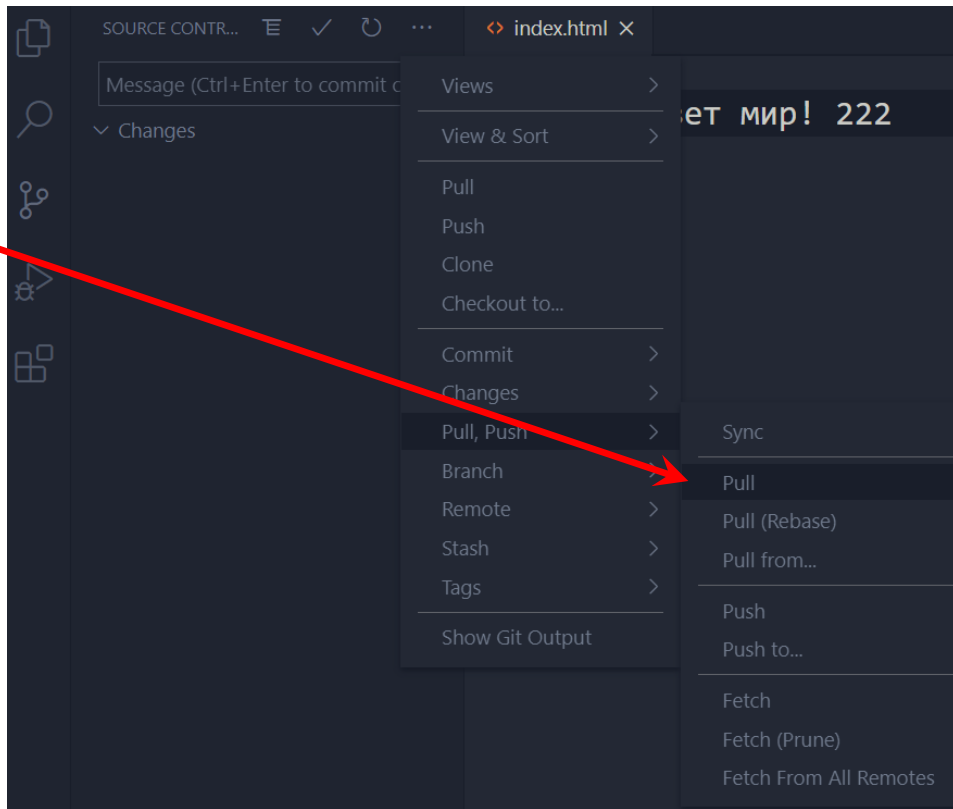


Сохранение кода прямо в Git через Visual Code.

Если нужно наоборот, загрузить изменения из git, выберите Pull.



Это если вы вручную на сайте git что-то поменяли, лучше работать только через редактор.



Квест на урок 2

1. С помощью тегов `div` смоделируйте следующую структуру:

Автомобиль состоит из кузова и шасси, в кузове есть окна, двери и двигатель, на шасси есть 4 колеса и коробка передач, двигатель состоит из корпуса и четырех поршней.

Наберите код и визуализируйте с помощью `example1` полученную структуру.

2. С помощью тегов `div` смоделируйте реальную структуру сайта, например – в оболочке сайта находится заголовок, тело и подвал, тело состоит из меню и содержимого. Заголовок состоит из картинки и названия. Меню состоит из элементов (элемент1, элемент2). Тело состоит из заголовка, галереи и статьи. Галерея состоит из секций, каждая секция состоит из заголовка и изображения. Подвал состоит из копирайта и телефона.

3*. Создайте любую свою структуру, чем больше уровней вложенности получится тем лучше. Это может быть как реальная вещь, так и сайт, можно и то и другое.

Квест на урок 2

4. Сделайте страничку как на примере (можно не точно и больше сделать), можно использовать свою тему и свои изображения, должно быть использовано выравнивание по центру, код должен быть выровнен через Beautify и выложен на ваш хостинг. Обязательно выделите структуру страницы и обозначьте ее тегами div. Пришлите файлы и ссылку на сайт.

Сохраните ваш сайт в папочку urok2 в git и пришлите ссылку на сайт в git, например так <https://embersong.github.io/urok2/>

- 5*. Изучите сайт <http://htmlbook.ru/> и найдите еще какие ни будь строчные теги, прочитайте описание и примените их в п.4.

Сайт о кошках и собаках

Собаки



Кошки



(c) 2020