Мы ввели адрес сайта — [thecode.media](https://thecode.media/" \t "_blank) — и нажали энтер. Что происходит дальше?

Для начала браузер определит был введен поисковый запрос или URL.

**Из чего состоит URL?**

Название протокола - – способ общения с ресурсом, которому отправляется запрос. Браузер поддерживает следующие протоколы – http, https



**Поиск сервера в интернете**

Каждый сайт в сети физически хранится на каком-то сервере. Как только браузер от нас получил адрес сайта, он должен понять, к какому серверу обратиться за данными. То, что мы ввели в строку поиска - доменное имя. Браузеру необходимо определить по доменному имени ip-адрес, на который отправлять запрос. Чтобы понять, какой именно IP-адрес у сервера с нашим сайтом, браузер делает так:

1. Сначала смотрит, посещали мы этот сайт раньше или нет. Если посещали — возьмёт IP-адрес из истории.
2. Если не посещали — посмотрит в конфигурационных файлах операционной системы.
3. Если в настройках такого нет, браузер смотрит недавние адреса в роутере, через который компьютер подключается к интернету.
4. Если и там нет, то браузер отправляет запрос на DNS-сервер. Там точно всё есть.

DNS-сервер — это такая служба в интернете, которая отвечает всем желающим на вопрос «Какой IP у такого-то домена?». Таких серверов в интернете много, и каждый из них знает про свою часть сети. Если у ближайшего сервера нет записей о нашем домене, то он отвечает «Я не знаю, спроси у DNS-сервера покрупнее, вот его адрес». В итоге браузер найдёт DNS-сервер, который знает то, что нам нужно, и получит IP-адрес сервера с сайтом.

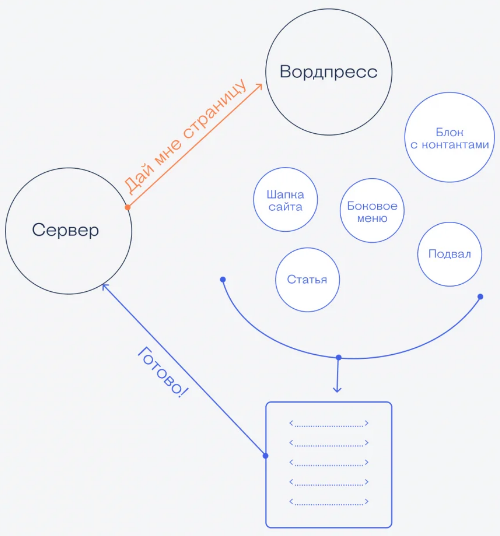
**Отправка запроса**

Браузер нашёл IP-адрес сервера, на котором располагается наш сайт, и отправляет по этому адресу запрос типа «Я знаю, что у тебя есть вот такой домен. Мне нужна вот такая страница с этого домена с такими-то параметрами. Дай, пожалуйста».

Чтобы всё было безопасно и данные никто не перехватил по пути, браузер и сервер договариваются шифровать все сообщения друг другу. Как только все формальности соблюдены, сервер отвечает «Да, конечно, сейчас всё отправлю». Иногда в адресе бывают ошибки, и сервер не может у себя найти нужную страницу. Тогда он отвечает «А у меня нет нужной страницы, ничем не могу помочь», и браузер показывает ошибку.

**Сервер думает**

Когда сервер получает запрос от браузера и с адресом всё в порядке, он начинает готовить данные к отправке. Для этого он смотрит, какие серверные программы отвечают за этот домен, и говорит им «Соберите мне вот эту страницу, чтобы я её отправил в браузер». Например, на сервере может стоять Вордпресс или PHP-обработчик, который на лету собирает страницу из разных фрагментов кода.



**Отправка данных в браузер**

Как только сервер получил от своих внутренних программ всё, что ему нужно, он отправляет результат в браузер.

Для этого он нарезает все данные на мелкие пакеты данных по 8 килобайт, нумерует их и отправляет браузеру. Так делается для того, чтобы одновременно передавать много пакетов — в этом случае загрузка идёт быстрее. Нумерация нужна для того, чтобы браузер потом собрал все пакеты в одно целое и получил исходный документ. Если по пути пакет потерялся, браузер говорит серверу «У меня потерялись такие-то пакеты, отправь их ещё раз». И так до тех пор, пока браузер не соберёт все пакеты.

**Браузер думает**

Когда все пакеты собраны, браузер разбирает документ на составляющие:

* HTML;
* CSS;
* JavaScript;
* прочий код, который браузер может выполнить.

Это нужно для того, чтобы браузер построил DOM-модель страницы. Такая модель содержит:

* все элементы, которые есть на странице;
* связи между ними;
* как они взаимодействуют между собой;
* что умеют и как реагируют на действия пользователя.

На основе DOM-модели браузер в итоге будет рисовать страницу на экране.

**Отрисовка страницы**

Последнее, что нужно сделать браузеру, — взять DOM-модель, найти в ней все видимые элементы и нарисовать их на экране. Если есть JavaScript-код, то он выполняется либо до отрисовки, либо после, смотря как работает скрипт.

Иногда во время отрисовки страницы браузер может снова запросить данные у сервера. В этом случае браузер рисует то, что есть, а остальное — когда придут данные. Пока данных нет, на странице могут быть пустые места — например, браузер отрисовал верхнее меню и статью, но ещё не подгрузил видео с ютуба.

**Всё готово**

Когда страница загрузилась и браузер всё нарисовал, мы видим готовый результат. Но даже сейчас браузер может продолжать работать над страницей:

* выполнять JS-скрипт;
* подгружать в фоне музыку или видео;
* подгружать страницы, на которые переходят с этого сайта чаще всего, чтобы создать эффект моментальной загрузки новых страниц;
* записывать что-то в куки или в локальное хранилище;
* собирать данные о том, что вы делаете на странице;
* и что угодно ещё, что предусмотрели программисты.