JavaScript с нуля за 100 часов

как ваш первый язык программирования

Встреча первая.

Где писать код и как его выполнять (запускать программу)?

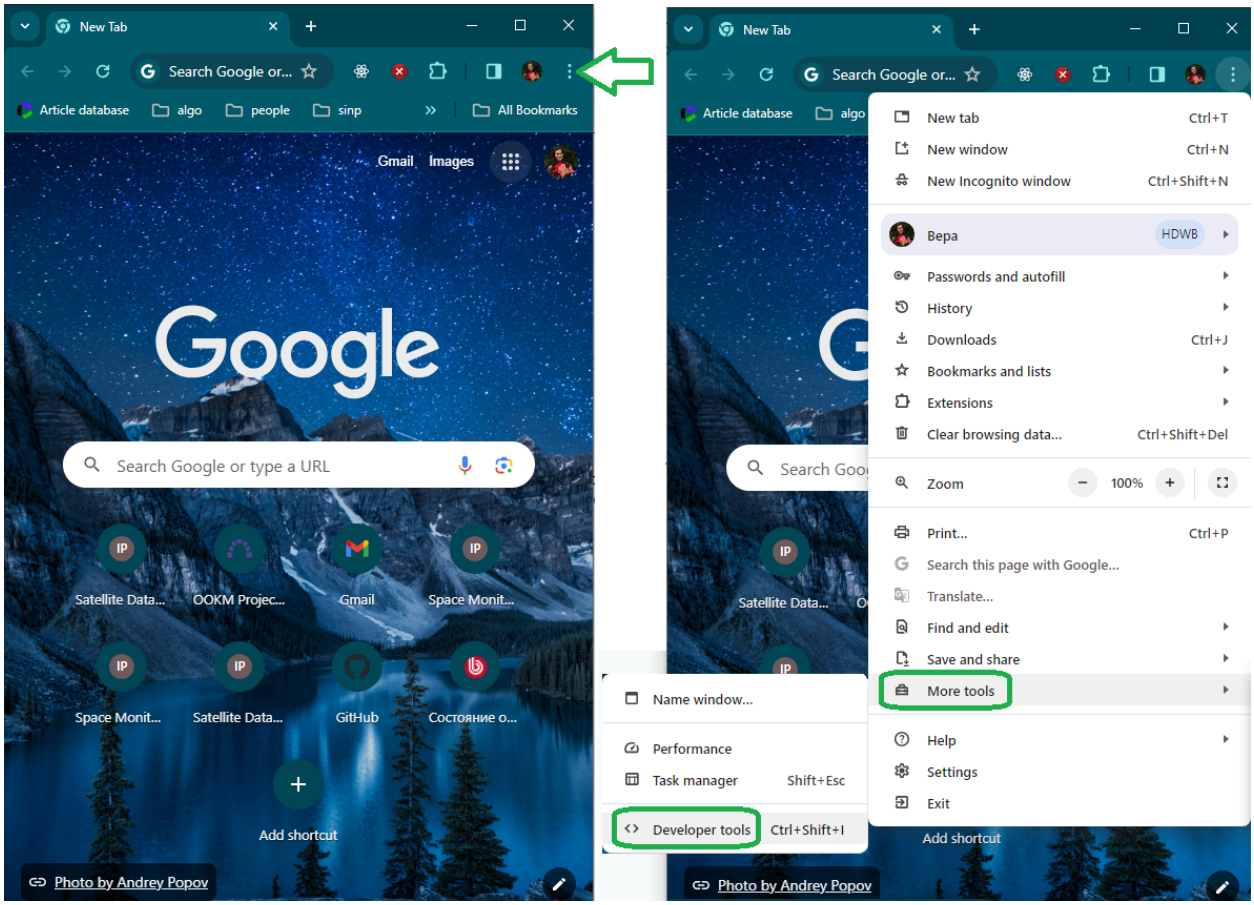
Академия ТОП
Баринова Вера Олеговна
системный программист и преподаватель
стаж с 2005 года

В одной команде

Когда-то программы даже не "писали прямо для компьютера", а выбивали в перфокартах, паяли или надевали перемычки на штырьки. Сейчас можно в удобном кресле на удобной клавиатуре "набирать" текст программы на практически человеческом языке, видя его на большом плоском ярком экране и сразу же оценивая результат, а всю работу по доведению его до физического уровня выполняют наши помощники - контроллеры устройств, операционная система и, наконец, программа-интерпретатор в программе-браузере. Мы с ними в одной команде, они созданы, чтобы упростить нам многие рутинные действия, но за свою работу они требуют уважения и бережного обращения, а иногда излишне нас ограничивают, "заботясь о безопасности" - нашей, своей и своих создателей. Попробуем притереться и поработать вместе?

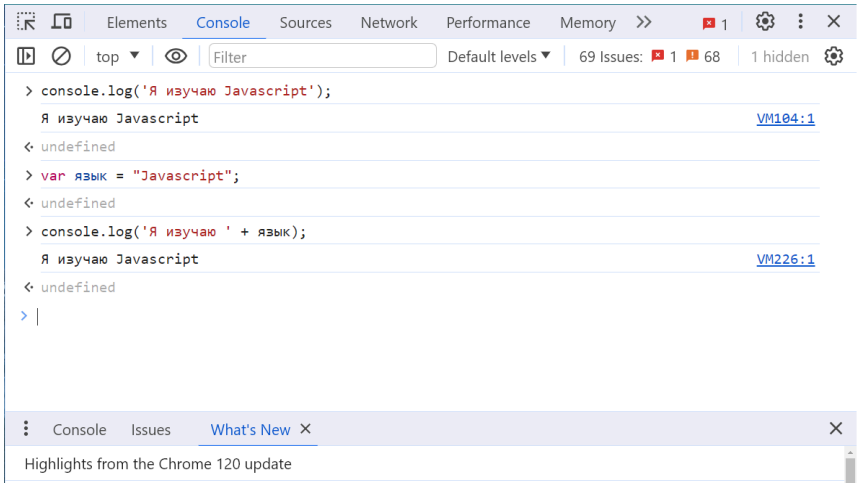
Браузер

JavaScript может исполняться в браузере, либо с помощью системы вроде Node JS. Поработаем с браузером. Для этого вам потребуется ноутбук или компьютер.

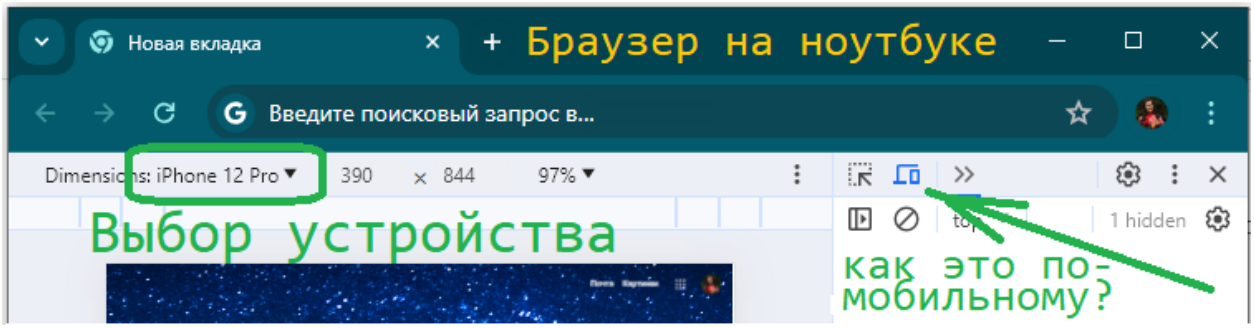


Вы видите консоль. Слева сверху на консоли есть квадратик со стрелочкой, означающий возможность открыть её в отдельном окне, а не как часть вашей страницы,

например, чтобы "не слетала вёрстка".



В телефонах и планшетах "из коробки" в браузерах пока (?) не поддерживается режим разработчика. Для разработки телефонной версии на смартфоне можно установить специализированное программное обеспечение, либо пользоваться online-песочницами. Для отладки телефонной версии вашего сайта на ПК существует специальная кнопка на панели разработчика в десктопной версии браузера.



Кнопка похожая на ноутбук, перед которым стоит телефон, - это кнопка переключения в режим отладки смартфонно-планшетных версий сайта. Она не влияет на интерпретатор JavaScript и нужна для подгона и подбора размещения элементов на странице в мобильной версии.

В разных браузерах интерфейс отличается (а чего уж там говорить про разные операционные системы!), по мере обновлений даже интерфейс разных версий одного браузера может отличаться.

Консоль - это оболочка для выполнения команд. Консолями называют командную строку Windows (cmd, powershell), терминалы Linux и специальное окно браузера, в котором можно выполнять команды JavaScript. На скриншоте (мгновенном изображении экрана, которое можно зафиксировать специальной кнопкой PrtSc на клавиатуре) вы видите несколько строк. Давайте сразу привыкнем к условным обозначениям, чтобы легче было "читать консоль". Уголок-стрелочка вправо (>) означает приглашение к вводу, т.е. место, куда можно вводить команду. Программисты также говорят "вбить", подразумевая немного неуравновешенное состояние души, когда не получается достичь желаемого результата и человек, не контролируя себя, "лупит по клавиатуре", что приводит к её непоправимому разрушению. Особенно ноутбука, где под клавиатурой находится

"математика", т.е. микросхемы и другие устройства. Если не получается, следует отойти от компьютера как можно дальше и выполнить несколько приседаний до успокоения, либо изнеможения.

Стрелочка влево с точечкой (‹·) означает возвращенное предыдущей командой значение. Пока игнорируйте его.

Команда console.log состоит из двух слов:

1) **console** (консоль) - это запрограммированный заранее разработчиками изучаемого языка объект JavaScript. Он многое умеет (наделён функциями), а что такое объект класса и как создавать свои объекты, вы узнаете на 5-м и 10-м уроках.
2) **log** - это функция "журнал", в данном случае "журналировать, записать". Это одна из суперспособностей (функций) объекта console - выводить на экран текстовые сообщения, числа и другие вещи, которые мы захотим рассмотреть.

Консоль не предназначена для программирования, это не среда программирования и даже не текстовый редактор. В ней удобно писать отдельные команды, но неудобно писать программу: её невозможно сохранить как файл скрипта без сохранения строк-ответов, невозможно "засунуть обратно". Можно лишь сохранить результаты экспериментов в виде файла, содержащего и входные, и выходные строки, нажав на консоль правой кнопкой мыши и выбрав пункт "Сохранить, как..." Зато в консоли будут появляться сообщения об ошибках, если вашим компьютером во время вашего отсутствия завладеет злоумышленник. Вы же не планируете делать ошибки и всегда будете писать только правильные программы? (*истеричный закадровый смех*) Также в консоль удобно выводить сообщения на экран или пробовать отдельные команды, когда хочется "мне только спросить", а не писать целую программу. В примере на рисунке в консоль было введено выражение, заканчивающееся точкой с запятой. Это - **инструкция (команда)**!

```
JavaScript
console.log('Я изучаю JavaScript');
```

После нажатия Enter в консоли появились две строки: ***Я изучаю JavaScript*** и ***undefined***. Первая строка - результат работы метода **log** объекта **console**. Вторая - результат, возвращённый этим методом. Метод записи в консоль ничего не возвращает - только выводит сообщение, поэтому возвращенный результат не определен. Позже мы подробнее познакомимся с понятиями "метод", "объект" и "возвращаемое значение".

Кроме того, обратите внимание, что фраза, которую мы выводим, заключена в кавычки, затем - в скобки. Кавычки обозначают тип данных "строка" (подробнее о типах данных будет рассказано во втором уроке), а скобки означают вызов функции или метода. Разницу между функцией и методом вы узнаете в 10-м уроке.

Пусть требуется выводить похожие сообщения, ***меняя*** название языка. Для этого нам потребуется **переменная** - название для объекта, который может хранить разные **значения**. Переменную можно представлять себе в виде ёмкости, где сама ёмкость остается прежней, а то, что в ней хранится, может меняться.

Заметка для ~~вундеркиндов~~ лиц, уже знакомых с другими языками программирования (ЯП): недостатком такой идеи применительно к языкам, подобным JavaScript, состоит в том, что языки с нестрогой (слабой) типизацией не накладывают ограничения на тип данных, который хранит переменная: он также может меняться по ходу работы программы. Получается "резиновая" ёмкость. Пусть это вас не смущает. Она ОЧЕНЬ резиновая. И липкая. К ней могут "приклеиваться" другие переменные! (уже страшно?)

Раньше переменные должны были содержать только латинские (английские) символы, подчеркивания и цифры, начинаясь непременно с буквы. Теперь мы можем называть переменые по-русски (используя символы кириллицы). В примере создана переменная “язык”. Это делается с помощью одного из ключевых слов **`var`** или **`let`** (любое из них указывает, что далее будет создана новая переменная; разница будет разъяснена позднее) затем следует название для переменной, затем одинарный знак равенства, который обозначает оператор присваивания (запись в переменную), затем - значение. Это - тоже инструкция, значит, она заканчивается точкой с запятой.

```
JavaScript
var язык = "JavaScript";  // теперь можно называть переменные по-русски!
```

Следует сразу отметить, что браузер "слишком добренький" и порой простит вам то, чего не простит другой интерпретатор другого языка, но (тревожная музыка усиливается) вам следует помнить о многочисленных юмористических сценах из фантастики, где результатом неосторожного колдовства были блюда на паучьих лапках с бутербродом, вмороженным в кристалл, либо козы, у которой там, где рога - нога и т.д. - добренький браузер, получив нечёткую или неаккуратно записанную инструкцию, исполнит её по своему усмотрению, которое далеко не всегда легко понять и исправить даже опытному программисту. Например, отсутствие точки с запятой допустимо, но условия, при которых оно допустимо, слишком сложны для первого урока.

Значение переменной можно изменить только присваиванием, однако, справа от знака присваивания может стоять не только явно записанное число или строка, но и выражение, которое может быть вычислено, либо результат вызова функции - её значение возврата. О возвращаемых значениях мы немного поговорим завтра, а затем - в 7-ом уроке.

Здесь и далее мы будем употреблять JavaScript не только в значении "язык программирования", но и "интерпретатор этого языка". Чтобы JavaScript отличал ключевые слова, названия функций и переменных от "просто текста", просто текст берётся в кавычки. Не важно, двойные, обратные или одинарные - главное, чтобы начало и конец были заключены в одни и те же кавычки и внутри эти кавычки не встречались. Не все символы (если это не буквы и цифры) одинаково полезны для строки. О способе добавить в строку запрещенные символы поговорим на втором уроке, посвященном типам данных. Также, не важно, представляет ли текст собой целый рассказ, одно или несколько предложений, одно слово или один символ - он относится к типу данных string, который мы изучим завтра очень подробно.

Строки можно складывать знаком +. При этом они будут приклеиваться друг к другу подряд. Склеивать можно не только две строки, записанные в кавычках, но и две строки, одна из которых записана в кавычках, а вторая - в переменную.

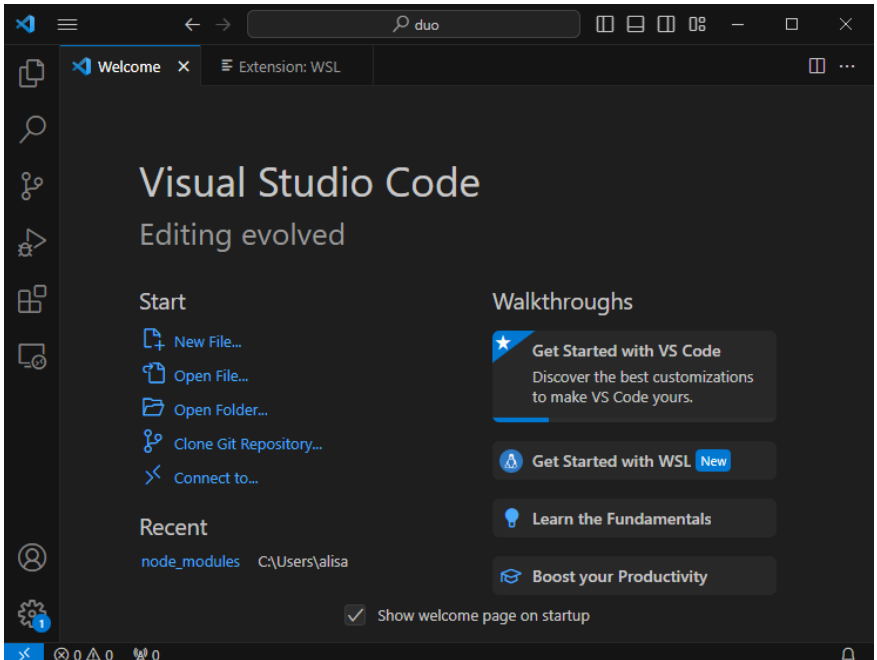
```JavaScript
var язык = "JavaScript";
console.log('Я изучаю ' + язык);
```

В результате работы этих двух строк во второй произойдет подстановка значения в переменную и после вычисления результат сложения строк будет напечатан в консоль. Мы будем использовать термин "на печать" в том числе имея в виду вывод на экран в консоль. Если потребуется вывод на печать на принтер, мы будем подчеркивать, что речь пойдёт именно об этом устройстве.

Настоятельно рекомендую (требую!) пробовать все, что описано в примерах. Чтение учебника по программированию не предполагает пледа и какао с зефиринками, потому что плед производит статическое электричество, а какао, случайно пролитое вашим котом (разумеется, не вами) на ноутбук, может вывести его (ноутбук, а не кота) из строя. Вы же не планируете читать учебник по программированию БЕЗ компьютера?

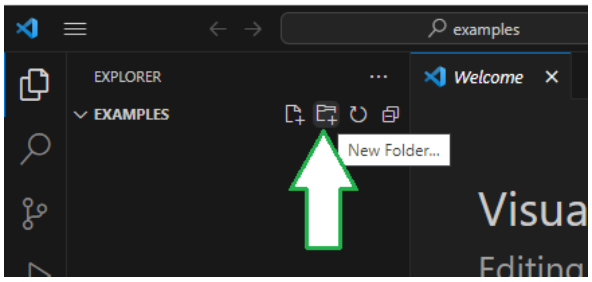
Текстовый редактор. Файлы HTML и JS

В качестве текстового редактора нашего кода мы будем использовать Visual Studio Code. Это бесплатная облегчённая среда разработки, которая позволит вам сформировать базовые навыки работы с кодом и системой контроля версий. Поскольку сред программирования великое множество, описание более чем одной превратило бы учебник по программированию в статью-обзор различных редакторов, что не входит в наши цели в связи с вашим желанием выучить язык как можно быстрее. Создайте директорию для учёбы. Я назвала свою examples, Вы можете назвать ilearnjs



После установки, где вы согласитесь со всеми пунктами в диалоговых окнах, если не знаете, как поступить лучше, и после запуска программы вы увидите окно, подобное этому. Следует выбирать пункт Open Folder... поскольку уже в первом проекте Вы будете иметь

дело не с одним, а с несколькими файлами. Отыщите свою папку, которую только что создали. Кроме того, следует сразу же поддерживать порядок, например, для каждого урока создавать новую директорию, которая называется Дата_тема. Но никаких кнопок нет! Однако если навести мышь на строку с названием папки, появятся 4 картинки-кнопки: Создать файл, Создать директорию, Обновить содержимое, Свернуть все вложения.

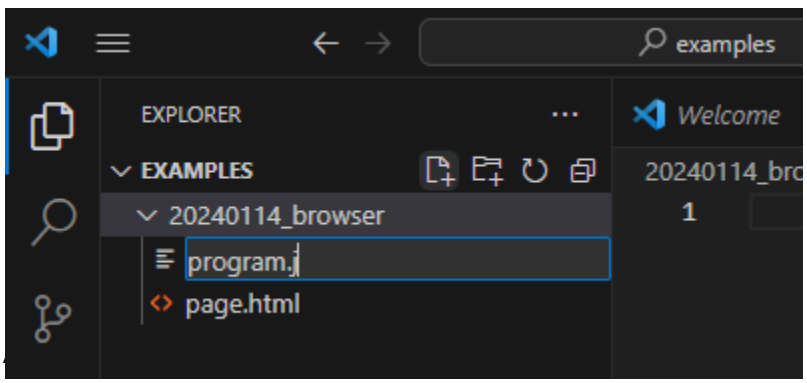


Если вы случайно нажмёте на последнюю кнопку, все файлы не "пропадут", как можно было бы подумать, а будут скрыты. Чтобы увидеть их снова, нажимайте на уголки слева от названий директорий - содержимое будет отображено в виде дерева (“плакучей ивы”) - веток, свисающих вниз. Сегодня можно назвать её 20240114_browser, где 2024 это год, 01 - месяц, 14 - день, а browser - это тема урока, т.к. сегодня мы пытаемся уговорить браузер выполнять наши программы. Папка появится и сразу откроется. Когда вы нажмёте "Создать файл" (New file), курсор будет мигать в поле ввода названия. Введите туда page.html Это будет страница сайта, без которой браузер и не подумает исполнять наш код. Позже (или ранее, в курсе HTML)как это - ранее наполним? мы наполним её интересным содержимым, а пока что она будет стартовой точкой для исполнения кода JavaScript.

Если вы никогда до сегодняшнего дня не работали с HTML, воспользуйтесь примером ниже, вставьте имя файла программы в текст page.html

```
Unset
<meta charset='utf-8'>
<script src='program.js'> </script>
```

и пока больше не думайте о нём. Первая строка поможет браузеру с кодировкой, чтобы русские буквы отобразились верно, а вторая сообщит браузеру, что к этой странице необходимо прочитать и исполнить файл program.js, расположенный в той же папке. Либо изучите предварительно HTML.
Ему посвящены другие учебники и курсы в нашей академии, однако, скорее всего, если вы являетесь нашим студентом, идущим по одной из рекомендованных программ разработки программного обеспечения, вы уже владеете HTML в достаточной степени.

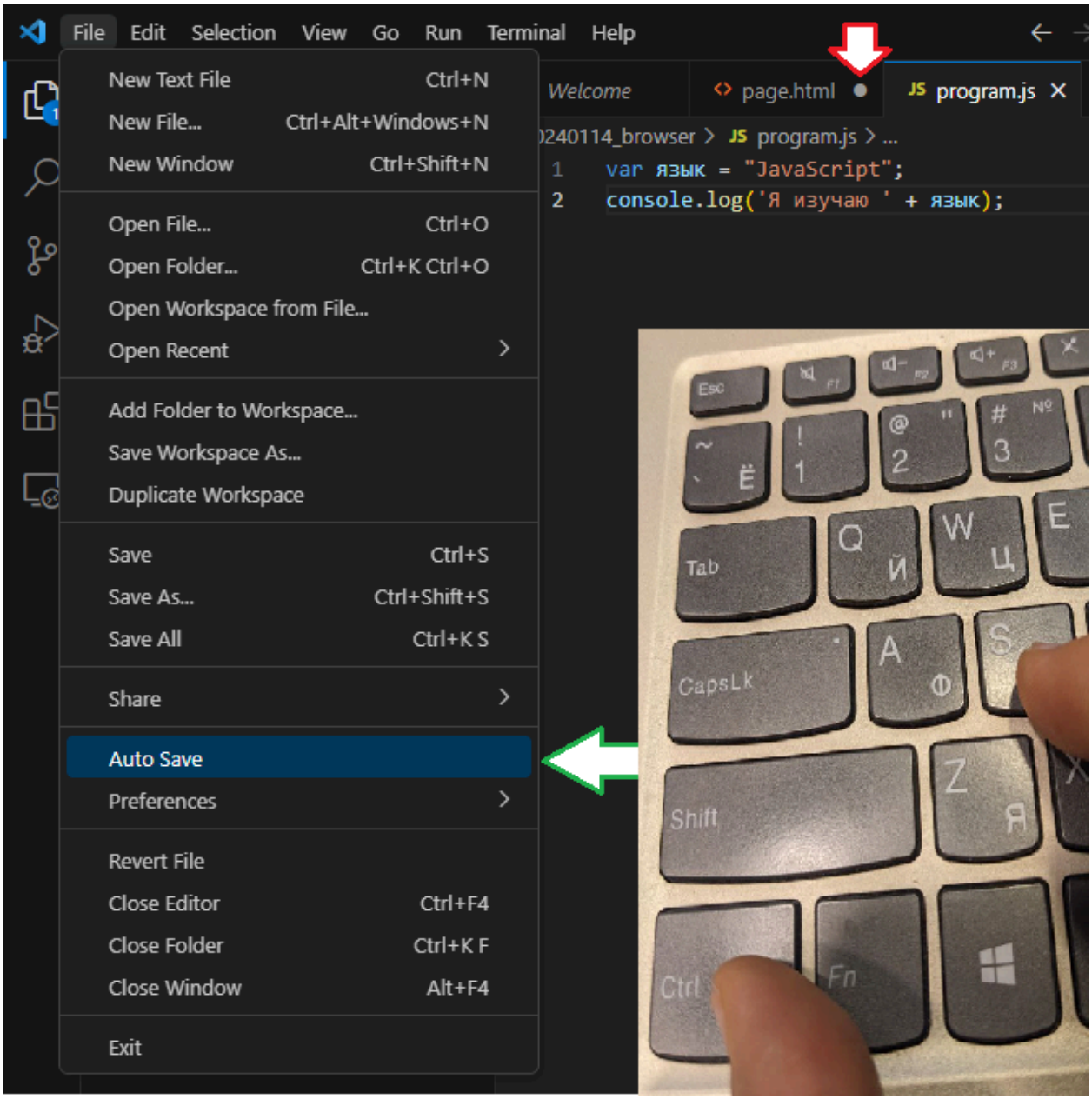


Повторите эти же действия самостоятельно для создания файла program.js
Обратите внимание, как только вы введете js, вместо полосок перед файлом появится значок, а справа открывается новая вкладка с

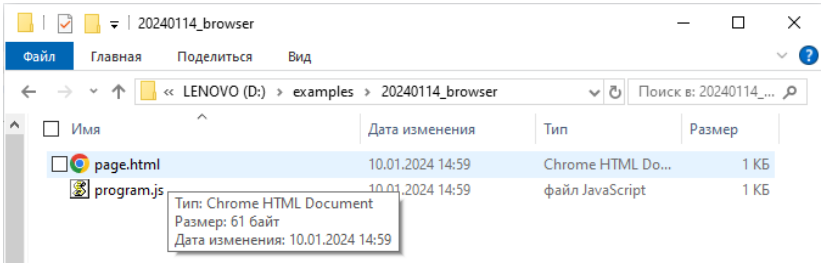
пустым файлом! Это будет файл, содержащий код скрипта нашей программы, который нам еще предстоит написать. Он сразу же откроется справа. Напишите:

```javascript
var язык = "JavaScript";
console.log('Я изучаю ' + язык);
```

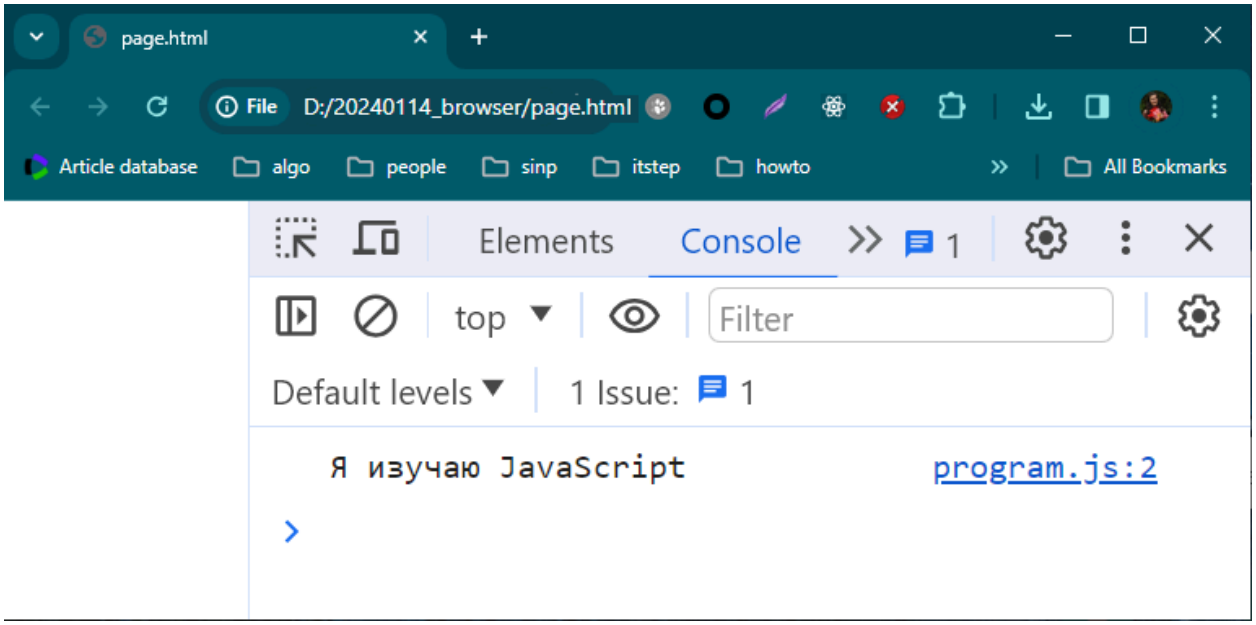
Файлы, которые вы создаете, необходимо постоянно сохранять. Это самое главное правило. Если файл открыт, редактируется и не сохранен, при внезапном отключении питания или сбое в работе программы вы можете потерять не две-три, а гораздо больше драгоценных строк.



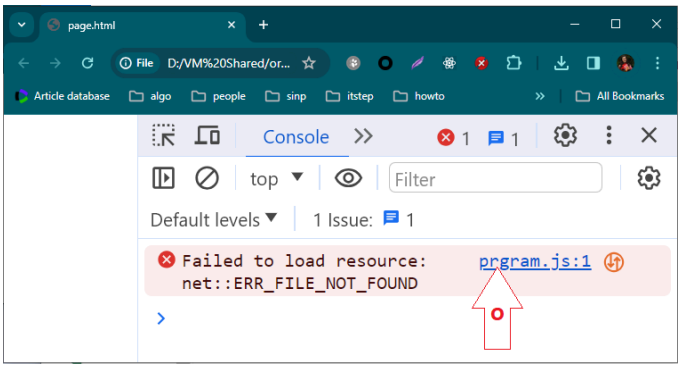
Кружочек на вкладке рядом с именем файла сигнализирует о том, что файл не сохранен. Можно настроить Auto Save в редакторе, но лучше приучить себя нажимать **ctrl+s** каждый раз, как только вы задумались над тем, чтобы ввести следующие несколько символов.



Теперь найдите эту директорию в проводнике (или как он там называется в вашей операционной системе). Кликните по файлу page.html дважды, он откроет страницу в браузере. Откройте консоль.

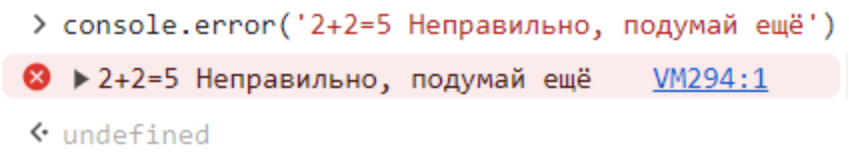


Поздравляем! Это ваша первая страница с работающим Js в ней! Возможно, вместо надписи выше вы видите что-то вроде этого:



Так в консоль выводятся ошибки. Это означает, что Вы опечатались и указали имя файла, которого реально нет (например, я пропустила букву о). Но браузер не будет просматривать вашу директорию и угадывать, какой файл вы имели в виду. Красную стрелочку с буковкой я нарисовала в MS Paint, а красный фон и красный кружок с крестиком - браузер.

Вы тоже можете выводить в консоль сообщения об ошибках. Например,



В приведённом примере никакой проверки в действительности не происходит. Всё, заключенное в кавычки является текстом и выводится на красном фоне с кружочком с крестиком только потому, что вместо console.log была использована функция console.error Полный список того, что можно делать в консоли Google Chrome, доступен по ссылке, правда, на английском языке, и постепенно пополняется, а для своего любимого браузера Вы можете найти такой список самостоятельно. Чуть позже мы познакомимся с ещё несколькими удобными выводами в консоль, например, в виде таблицы.

А теперь подведём итоги:

Вы узнали, как создавать скрипт и где его запускать. Встретились с первой ошибкой. Познакомились со средой разработки Visual Studio Code. Решили, что будете выучивать новые слова на английском языке.

Вопросы по теме:

1. Как создать собственную страницу в браузере?
2. Что такое консоль и как в неё войти?
3. Можно ли на ноутбуке отладить страницу для смартфона?
4. Где хранится код скрипта JS? Как его можно редактировать (укажите не менее двух способов)?
5. Как “сообщить” странице браузера предназначенный для неё код JS?
6. Может ли несколько файлов кода js привязано к одной странице браузера? А один файл кода js - к нескольким разным страницам?

Задания для самостоятельной работы

1. Выведите в консоли строку «Я читаю первую главу»
2. Используя переменные, изменяйте в выводимой в консоли строке «Я читаю первую главу» номер главы на «вторую», «третью», «пятую». Выведите все 4 строки.
3. Создайте текст программы, выводящей на экран код для вывода строки «Я изучаю JavaScript» с комментариями к каждой строке.
4. Создайте текст программы, выводящей в столбик список дней недели. Сохраните файл под именем «Week.js» и запустите его при помощи браузера. Какой ещё файл вам потребовалось создать?

5. Сравните 2 любых редактора кода и опишите 5 любых инструментов, имеющихся в обоих редакторах, с точки зрения удобства пользования.
6. Напишите подробную пошаговую инструкцию для написания и запуска из браузера программы, выводящей в консоль строку «Здравствуй, мир!», начиная с включения компьютера и заканчивая сохранением наработок и выключением. Можете изобразить результат с помощью блок-схемы

Советы по выполнению домашнего задания:

1. Измеряйте время, потребовавшееся для выполнения
2. Записывайте источники, которые вам помогли.
3. Если вы изучаете предмет в группе, не забывайте делиться результатами и неудачами с одногруппниками. Это позволит быстрее справиться с проблемами и подтолкнёт других к началу выполнения дз.
4. Начинать всегда сложно. Запишите задачу, которую собираетесь решить, в тот же файл, в котором будет решение, и начало выполнения. Текст нужно оформить как комментарий. Это позволит не отвлекаться, переключаясь между окнами, и поможет не забыть, что требовалось сделать.
5. Храните порядок в рабочем каталоге. Когда вам потребуется что-то вспомнить, всё нужное будет под рукой.

Словарь урока

Поскольку вы никак не ожидали, что в учебнике по JS будет словарь, поясню, что приведенные здесь переводы даются именно в контексте программирования, иногда в виде статьи, как в толковом словаре. Слова могут иметь жаргонный (слэнговый) характер, использоваться в переносном смысле, но важно, что именно этот смысл вкладывают в него разработчики.

script - скрипт - сценарий. Программа на языке программирования, предназначенная для исполнения другой специальной программой - интерпретатором.
run - ран - ("бег, бежать") запуск, начало выполнения.
site - сайт - ("участок")
browse - брауз - ("пастись", "просматривать"), browser - программа для доступа к сайтам в интернете
more - мо - больше
new - нью -новый. Часто используется как глагол "создать", хотя им не является
tab - тэб - вкладка, листок. Ещё - специальный форматирующий отступ. Устаревающий.
develop - дивелоуп - разрабатывать, программировать. Developer - это Вы! программист, разработчик
tool - тул -инструмент
console - консоул - консоль, она же shell ("оболочка") - командная строка

log - лог - журнал. console.log("надпись") - записать в консоль "надпись".
enter - вход, ввод. Клавиша на клавиатуре, переводящая курсор на новую строку, начинающая новый абзац, запускающая одну команду в командной строке или отправляющая сообщение в системах обмена сообщениями (мессенджерах).
visual - в**и**жуал -("визуальный") - созданный для передачи и восприятия зрительной информации
define - дифайн - определить. **undefined** - неопределенный
variable - в**э**риэбл - переменная. **var** - сокращение для variable
value - вэлью - значение
let - лэт - разрешить, допустить
string - стрин (но по-русски все говорят стрин**г**) - строка (как текст, а не как одна строка текста до переноса)
open - оупен - открыть
fold - фолд - заворачивать. **folder** - ф**о**лдэ - свёрток, директория (папка с файлами и другими директориями)

Тест по уроку
Тест по английскому языку

Вперед, к типам данных и принятию компьютером решения!