МИНИСТЕРСТВО НАУКИ и высшего образования РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

филиал «РКТ» МАИ в г. Химки Московской области

**Специальность 09.02.03 «Программирование в компьютерных системах»**

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**ПМ.02«Разработка и администрирование баз данных**»

**Студент**

**Группы МП-32\_\_\_\_\_\_\_\_\_Барабанцев Р.А.\_\_ / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )**

**Руководитель**

**практики от организации \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**Руководитель**

**практики от филиала Шумаев А.Ю. / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )**

**2020г.**

**ПРОГРАММА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

По специальности 09.02.03 «Программирование в компьютерных системах»

|  |  |  |
| --- | --- | --- |
| Дата | Наименование выполняемых работ | Количество дней практики |
| 11.06.20 | Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. | 4 |
| 12.06.20 | Описание структуры сети, в которой может функционировать разработанная база данных. | 4 |
| 13.06.20 | Описание сетевого оборудования необходимого для работы с базой данных. | 4 |
| 15.06.20 | Описание программного обеспечения необходимого для работы с базой данных по сети. | 4 |
| 16.06.20-18.06.20 | Разработка клиентских интерфейсов для клиент-серверных приложений. | 16 |
| 20.06.20 | Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. | 4 |
| 22.06.20-24.06.20 | Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. | 16 |
| 25.06.20-01.07.20 | Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. | 50 |
| 02.07.20-03.07.20 | Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. | 12 |
| 03.07.20-04.07.20 | Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. | 12 |
| 06.07.20 | Составление отчёта по практике | 6 |
| 07.07.20 | Подготовка к защите отчёта по практике. | 6 |
| 08.07.20 | Итоговая аттестация по производственной практике - зачёт | 6 |

Руководитель практики от филиала «РКТ» МАИ преподаватель

Шумаев А.Ю.

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

**АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Барабанцев Роман Александрович \_\_\_\_\_\_\_\_\_\_\_

*ФИО*

обучающийся (аяся) на 3-ем курсе по специальности СПО 09.02.03 «Программирование в компьютерных системах»

успешно прошел(ла) производственную практику по профессиональному модулю   
**ПМ.02 «Разработка и администрирование баз данных»**

в объеме 144 часа с 11.06.2020г. по 08.07.2020г.

в организации филиала «РКТ» МАИ

Виды и качество выполнения работ

|  |  |  |
| --- | --- | --- |
| Виды работ, выполненных обучающимся(ейся) во время практики | Объем работ | Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика |
| Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. | 4 |  |
| Описание структуры сети, в которой может функционировать разработанная база данных. | 4 |  |
| Описание сетевого оборудования необходимого для работы с базой данных. | 4 |  |
| Описание программного обеспечения необходимого для работы с базой данных по сети. | 4 |  |
| Разработка клиентских интерфейсов для клиент-серверных приложений. | 16 |  |
| Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. | 4 |  |
| Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. | 16 |  |
| Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. | 50 |  |
| Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. | 12 |  |
| Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. | 12 |  |
| Составление отчёта по практике | 6 |  |
| Подготовка к защите отчёта по практике. | 6 |  |
| Итоговая аттестация по производственной практике - зачёт | 6 |  |

Руководитель практики от филиала «РКТ» МАИ Шумаев А.Ю.

|  |  |  |
| --- | --- | --- |
| Дата\_\_\_\_\_\_\_\_\_\_\_\_\_ | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *подпись* | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Расшифровка подписи* |

**ДНЕВНИК** **ПРОХОЖДЕНИЯ ПРАКТИКИ**

|  |  |  |
| --- | --- | --- |
| Дата | Наименование выполняемых работ | Подпись  руководителя |
| 11.06.20 | Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. |  |
| 12.06.20 | Описание структуры сети, в которой может функционировать разработанная база данных. |  |
| 13.06.20 | Описание сетевого оборудования необходимого для работы с базой данных. |  |
| 15.06.20 | Описание программного обеспечения необходимого для работы с базой данных по сети. |  |
| 16.06.20-18.06.20 | Разработка клиентских интерфейсов для клиент-серверных приложений. |  |
| 20.06.20 | Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. |  |
| 22.06.20-24.06.20 | Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. |  |
| 25.06.20-01.07.20 | Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. |  |
| 02.07.20-03.07.20 | Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. |  |
| 03.07.20-04.07.20 | Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. |  |
| 06.07.20 | Составление отчёта по практике |  |
| 07.07.20 | Подготовка к защите отчёта по практике. |  |
| 08.07.20 | Итоговая аттестация по производственной практике - зачёт |  |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 7](#_Toc54194706)

[1 Node.js 8](#_Toc54194707)

[1.1 Скорость 8](#_Toc54194708)

[1.2 Простота 8](#_Toc54194709)

[1.3 JavaScript 8](#_Toc54194710)

[1.4 Движок V8 9](#_Toc54194711)

[1.5 Асинхронность 9](#_Toc54194712)

[2 Модули Node.js 11](#_Toc54194713)

[3 Бага-репорт 13](#_Toc54194714)

[ЗАКЛЮЧЕНИЕ 14](#_Toc54194715)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 15](#_Toc54194716)

[Приложение 16](#_Toc54194717)

# ВВЕДЕНИЕ

Практическая выполнена в соответствии с рабочей программой ПМ.02 «Разработка и администрирование баз данных». В работе представлены основные этапы анализа и разработки предметной области «Разработка и администрирование базы данных», навыки работы с Node.js.

В процессе выполнения проекта должен быть освоен ряд следующих профессиональных компетенций: ознакомление со структурой Node.js, реализация базы данных в конкретной системе управления базами данных. Получение знаний и опыта в области в технологии Jquery.

# 1 Node.js

Node.js — это опенсорсная кроссплатформенная среда выполнения для JavaScript, которая работает на серверах. С момента выпуска этой платформы в 2009 году она стала чрезвычайно популярной и в наши дни играет весьма важную роль в области веб-разработки. Если считать показателем популярности число звёзд, которые собрал некий проект на GitHub, то [Node.js](https://github.com/nodejs/node), у которого более 50000 звёзд, это очень и очень популярный проект.

Платформа Node.js построена на базе JavaScript движка V8 от Google, который используется в браузере Google Chrome. Данная платформа, в основном, используется для создания веб-серверов, однако сфера её применения этим не ограничивается.

## 1.1 Скорость

Одной из основных привлекательных особенностей Node.js является скорость. JavaScript-код, выполняемый в среде Node.js, может быть в два раза быстрее, чем код, написанный на компилируемых языках, вроде C или Java, и на порядки быстрее интерпретируемых языков наподобие Python или Ruby. Причиной подобного является неблокирующая архитектура платформы, а конкретные результаты зависят от используемых тестов производительности, но, в целом, Node.js — это очень быстрая платформа.

## 1.2 Простота

Платформа Node.js проста в освоении и использовании. На самом деле, она прямо-таки очень проста, особенно это заметно в сравнении с некоторыми другими серверными платформами.

## 1.3 JavaScript

В среде Node.js выполняется код, написанный на JavaScript. Это означает, что миллионы фронтенд-разработчиков, которые уже пользуются JavaScript в браузере, могут писать и серверный, и клиентский код на одном и том же языке программирования без необходимости изучать совершенно новый инструмент для

перехода к серверной разработке.

В браузере и на сервере используются одинаковые концепции языка. Кроме того, в Node.js можно оперативно переходить на использование новых стандартов ECMAScript по мере их реализации на платформе. Для этого не нужно ждать до тех пор, пока пользователи обновят браузеры, так как Node.js — это серверная среда, которую полностью контролирует разработчик. В результате новые возможности языка оказываются доступными при установке поддерживающей их версии Node.js.

## 1.4 Движок V8

В основе Node.js, помимо других решений, лежит опенсорсный JavaScript-движок V8 от Google, применяемый в браузере Google Chrome и в других браузерах. Это означает, что Node.js пользуется наработками тысяч инженеров, которые сделали среду выполнения JavaScript Chrome невероятно быстрой и продолжают работать в направлении совершенствования V8.

## 1.5 Асинхронность

В традиционных языках программирования (C, Java, Python, PHP) все инструкции, по умолчанию, являются блокирующими, если только разработчик явным образом не позаботится об асинхронном выполнении кода. В результате если, например, в такой среде, произвести сетевой запрос для загрузки некоего JSON-кода, выполнение потока, из которого сделан запрос, будет приостановлено до тех пор, пока не завершится получение

и обработка ответа.

JavaScript значительно упрощает написание асинхронного и неблокирующего кода с использованием единственного потока, функций обратного вызова (коллбэков) и подхода к разработке, основанной на событиях. Каждый раз, когда нам нужно выполнить тяжёлую операцию, мы передаём соответствующему механизму коллбэк, который будет вызван сразу после завершения этой операции. В результате, для того чтобы программа продолжила работу, ждать результатов выполнения подобных операций не нужно.

Подобный механизм возник в браузерах. Мы не можем позволить себе ждать, скажем, окончания выполнения AJAX-запроса, не имея при этом возможности реагировать на действия пользователя, например, на щелчки по кнопкам. Для того чтобы пользователям было удобно работать с веб-страницами, всё, и загрузка данных из сети, и обработка нажатия на кнопки, должно происходить одновременно, в режиме реального времени.

Если вы создавали когда-нибудь обработчик события нажатия на кнопку, то вы уже пользовались методиками асинхронного программирования.

Асинхронные механизмы позволяют единственному Node.js-серверу одновременно обрабатывать тысячи подключений, не нагружая при этом программиста задачами по управлению потоками и по организации параллельного выполнения кода. Подобные вещи часто являются источниками ошибок.

Node.js предоставляет разработчику неблокирующие базовые механизмы ввода вывода, и, в целом, библиотеки, использующиеся в среде Node.js, написаны с использованием неблокирующих парадигм. Это делает блокирующее поведение кода скорее исключением, чем нормой.

Когда Node.js нужно выполнить операцию ввода-вывода, вроде загрузки данных из сети, доступа к базе данных или к файловой системе, вместо того, чтобы заблокировать ожиданием результатов такой операции главный поток, Node.js инициирует её выполнение и продолжает заниматься другими делами до тех пор, пока результаты выполнения этой операции не будут получены.

# 2 Модули Node.js

Любой проект посложнее «Hello World» состоит из некоторого количества файлов, по которым разносят код. Это дает возможность структурировать проект, вынести независимые части, которые можно будет использовать в других проектах и вообще сделать код нагляднее.

Так вот, в Node.js каждый такой файл и представляет собой модуль, который можно подключить.

Подключение происходит с помощью вызова функции require, которой нужно передать путь к файлу пример на рисунке 1



Рисунок 1- функции require

Данный код подключает модуль авторизации и делает его доступным через переменную authModule.

В зависимости от того, какой параметр передан в функцию require, будет отличатся алгоритм подключения модуля. Так что давайте посмотрим на принципы подключения модулей в Node.js. Отмечу, что вся эта информация доступна в [документации](http://nodejs.org/api/modules.html).

Вместе с Node.js поставляется несколько встроенных модулей, для подключения которых нужно просто указать название модуля.



Рисунок 2-пример подключения модулей

Нужно отметить, что встроенные модули имеют приоритет над всеми остальными, если в функцию require передано их название. Так как примеру, require('http') всегда вернет встроенный модуль, даже если будет сторонний модуль с таким названием или файл с таким именем. Список всех встроенных модулей и документацию по ним можно найти на [сайте](http://nodejs.org/api/). Исходники этих модулей можно посмотреть в [репозитории проекта](https://github.com/joyent/node/tree/master/lib).

Если передано название модуля и он не является встроенным, тогда идет подключение модуля из папки node\_modules. В данной папке находятся все модули, которые добавлены с помощью [NPM](https://www.npmjs.org/). NPM — это менеджер пакетов для Node.js, который упрощает поиск и подключение сторонних модулей. На момент написания этой статьи в нем находилось уже 89 503 модулей.

Чтобы подключить модуль который находится в node\_modules достаточно указать его название.

Давайте рассмотрим ситуацию, когда такой код был вызван из папки '/var/www/demo', тогда Node.js попытается найти указанные модули в следующих папках:

Node.js рекурсивно, каждый раз переходя в родительскую папку, будет искать папку node\_modules с нужным модулем. NPM также даем возможность установить модуль глобально (npm install -g MODULE), тогда он будет доступен из любого места.

Рассмотрим еще один вариант подключения, если переданный параметр в функцию require начинается с /, ../, или ./, тогда файл для подключения будет происходить по абсолютному пути или относительно текущей папки.

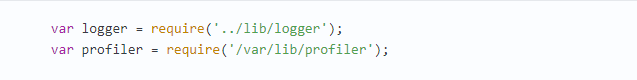


Рисунок 3- пример подключения модулей.

3 Баг-репорт

При подключение приложение если выдает баг EACCES при запуске http.Server.listen() пример на рисунке 4.

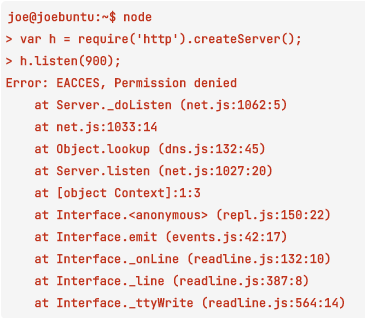


Рисунок 4 Пример ошибки EACCES

Решение этого бага как правило, процессы, запущенные без привилегий root, не может привязать к портам ниже 1024 пример так что попробуйте более высокий порт, или запустить с повышенными привилегиями через судо. Вы можете понизить привилегии после того, как вы привязаны к порту низкого, используя процесс на рисунке 5.



Рисунок 5 решение ошибки EACCES

Если вышеперечисленные инструкции не помогли переустановите и пере подключите приложение к северу

к

# ЗАКЛЮЧЕНИЕ

Мы рассмотрели большинство основных способов настройки клиента и сервера Node.js с тем, чтобы он был максимально соответствовал вашим рабочим процессам и проектам.

Мы узнали о всевозможных настройках, атрибутах файлов и о перехватчиках событий, а также рассмотрели пример настройки сервера с соблюдением политики. Теперь нам должно быть по плечу пользоваться Node.js подстроиться под практически любой тип рабочего процесса, который только можно вообразить.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Node.js: [Электронный ресурс]/ https:// https://tproger.ru/tag/node-js/ - статья в интернете.
2. Руководство Node.js: [Электронный ресурс]/ https://habr.com/ru/company/ruvds/blog/422893/ - статья в интернете.
3. Node.js документация: [Электронный ресурс]/ https://metanit.com/web/nodejs/ - официальная документация.
4. Краткое вступление в Node.js: [Электронный ресурс] https://github.com/nodejs/node - статья в интернете.

# Приложение

ПРИЛОЖЕНИЕ

/\*!

\* jQuery JavaScript Library v3.5.1

-ajax,-ajax/jsonp,

-ajax/load,

-ajax/script,

-ajax/var/location,

-ajax/var/nonce,-ajax/var/rquery

,-ajax/xhr,-manipulation/\_evalUrl,

-deprecated/ajax-event-alias,-effects,-effects/Tween,

-effects/animatedSelector

\* https://jquery.com/

\*

\* Includes Sizzle.js

\* https://sizzlejs.com/

\*

\* Copyright JS Foundation and other contributors

\* Released under the MIT license

\* https://jquery.org/license

\*

\* Date: 2020-05-04T22:49Z

\*/

( function( global, factory ) {

"use strict";

if ( typeof module === "object" && typeof module.exports === "object" ) {

// For CommonJS and CommonJS-like environments where a proper `window`

// is present, execute the factory and get jQuery.

// For environments that do not have a `window` with a `document`

// (such as Node.js), expose a factory as module.exports.

// This accentuates the need for the creation of a real `window`.

// e.g. var jQuery = require("jquery")(window);

// See ticket #14549 for more info.

module.exports = global.document ?

factory( global, true ) :

function( w ) {

if ( !w.document ) {

throw new Error( "jQuery requires a window with a document" );

}

return factory( w );

};

} else {

factory( global );

}

// Pass this if window is not defined yet

} )( typeof window !== "undefined" ? window : this, function( window, noGlobal ) {

// Edge <= 12 - 13+, Firefox <=18 - 45+, IE 10 - 11, Safari 5.1 - 9+, iOS 6 - 9.1

// throw exceptions when non-strict code (e.g., ASP.NET 4.5) accesses strict mode

// arguments.callee.caller (trac-13335). But as of jQuery 3.0 (2016), strict mode should be common

// enough that all such attempts are guarded in a try block.

"use strict";

var arr = [];

var getProto = Object.getPrototypeOf;

var slice = arr.slice;

var flat = arr.flat ? function( array ) {

return arr.flat.call( array );

} : function( array ) {

return arr.concat.apply( [], array );

};

var push = arr.push;

var indexOf = arr.indexOf;

var class2type = {};

var toString = class2type.toString;

var hasOwn = class2type.hasOwnProperty;

var fnToString = hasOwn.toString;

var ObjectFunctionString = fnToString.call( Object );

var support = {};

var isFunction = function isFunction( obj ) {

// Support: Chrome <=57, Firefox <=52

// In some browsers, typeof returns "function" for HTML <object> elements

// (i.e., `typeof document.createElement( "object" ) === "function"`).

// We don't want to classify \*any\* DOM node as a function.

return typeof obj === "function" && typeof obj.nodeType !== "number";

};

var isWindow = function isWindow( obj ) {

return obj != null && obj === obj.window;

};

var document = window.document;

var preservedScriptAttributes = {

type: true,

src: true,

nonce: true,

noModule: true

};

function DOMEval( code, node, doc ) {

doc = doc || document;

var i, val,

script = doc.createElement( "script" );

script.text = code;

if ( node ) {

for ( i in preservedScriptAttributes ) {

// Support: Firefox 64+, Edge 18+

// Some browsers don't support the "nonce" property on scripts.

// On the other hand, just using `getAttribute` is not enough as

// the `nonce` attribute is reset to an empty string whenever it

// becomes browsing-context connected.

// See https://github.com/whatwg/html/issues/2369

// See https://html.spec.whatwg.org/#nonce-attributes

// The `node.getAttribute` check was added for the sake of

// `jQuery.globalEval` so that it can fake a nonce-containing node

// via an object.

val = node[ i ] || node.getAttribute && node.getAttribute( i );

if ( val ) {

script.setAttribute( i, val );

}

}

}

doc.head.appendChild( script ).parentNode.removeChild( script );

}

function toType( obj ) {

if ( obj == null ) {

return obj + "";

}

// Support: Android <=2.3 only (functionish RegExp)

return typeof obj === "object" || typeof obj === "function" ?

class2type[ toString.call( obj ) ] || "object" :

typeof obj;

}

/\* global Symbol \*/

// Defining this global in .eslintrc.json would create a danger of using the global

// unguarded in another place, it seems safer to define global only for this module

var

version = "3.5.1 -ajax,-ajax/jsonp,-ajax/load,-ajax/script,-ajax/var/location,-ajax/var/nonce,-ajax/var/rquery,-ajax/xhr,-manipulation/\_evalUrl,-deprecated/ajax-event-alias,-effects,-effects/Tween,-effects/animatedSelector",

// Define a local copy of jQuery

jQuery = function( selector, context ) {

// The jQuery object is actually just the init constructor 'enhanced'

// Need init if jQuery is called (just allow error to be thrown if not included)

return new jQuery.fn.init( selector, context );

};

jQuery.fn = jQuery.prototype = {

// The current version of jQuery being used

jquery: version,

constructor: jQuery,

// The default length of a jQuery object is 0

length: 0,

toArray: function() {

return slice.call( this );

},

// Get the Nth element in the matched element set OR

// Get the whole matched element set as a clean array

get: function( num ) {

// Return all the elements in a clean array

if ( num == null ) {

return slice.call( this );

}

// Return just the one element from the set

return num < 0 ? this[ num + this.length ] : this[ num ];

},

// Take an array of elements and push it onto the stack

// (returning the new matched element set)

pushStack: function( elems ) {

// Build a new jQuery matched element set

var ret = jQuery.merge( this.constructor(), elems );

// Add the old object onto the stack (as a reference)

ret.prevObject = this;

// Return the newly-formed element set

return ret;

},

// Execute a callback for every element in the matched set.

each: function( callback ) {

return jQuery.each( this, callback );

},

map: function( callback ) {

return this.pushStack( jQuery.map( this, function( elem, i ) {

return callback.call( elem, i, elem );

} ) );

},

slice: function() {

return this.pushStack( slice.apply( this, arguments ) );

},

first: function() {

return this.eq( 0 );

},

last: function() {

return this.eq( -1 );

},

even: function() {

return this.pushStack( jQuery.grep( this, function( \_elem, i ) {

return ( i + 1 ) % 2;

} ) );

},

odd: function() {

return this.pushStack( jQuery.grep( this, function( \_elem, i ) {

return i % 2;

} ) );

},

eq: function( i ) {

var len = this.length,

j = +i + ( i < 0 ? len : 0 );

return this.pushStack( j >= 0 && j < len ? [ this[ j ] ] : [] );

},

end: function() {

return this.prevObject || this.constructor();

},

// For internal use only.

// Behaves like an Array's method, not like a jQuery method.

push: push,

sort: arr.sort,

splice: arr.splice

};

jQuery.extend = jQuery.fn.extend = function() {

var options, name, src, copy, copyIsArray, clone,

target = arguments[ 0 ] || {},

i = 1,

length = arguments.length,

deep = false;

// Handle a deep copy situation

if ( typeof target === "boolean" ) {

deep = target;

// Skip the boolean and the target

target = arguments[ i ] || {};

i++;

}

// Handle case when target is a string or something (possible in deep copy)

if ( typeof target !== "object" && !isFunction( target ) ) {

target = {};

}

// Extend jQuery itself if only one argument is passed

if ( i === length ) {

target = this;

i--;

}

for ( ; i < length; i++ ) {

// Only deal with non-null/undefined values

if ( ( options = arguments[ i ] ) != null ) {

// Extend the base object

for ( name in options ) {

copy = options[ name ];

// Prevent Object.prototype pollution

// Prevent never-ending loop

if ( name === "\_\_proto\_\_" || target === copy ) {

continue;

}

// Recurse if we're merging plain objects or arrays

if ( deep && copy && ( jQuery.isPlainObject( copy ) ||

( copyIsArray = Array.isArray( copy ) ) ) ) {

src = target[ name ];

// Ensure proper type for the source value

if ( copyIsArray && !Array.isArray( src ) ) {

clone = [];

} else if ( !copyIsArray && !jQuery.isPlainObject( src ) ) {

clone = {};

} else {

clone = src;

}

copyIsArray = false;

// Never move original objects, clone them

target[ name ] = jQuery.extend( deep, clone, copy );

// Don't bring in undefined values

} else if ( copy !== undefined ) {

target[ name ] = copy;

}

}

}

}

// Return the modified object

return target;

};

jQuery.extend( {

// Unique for each copy of jQuery on the page

expando: "jQuery" + ( version + Math.random() ).replace( /\D/g, "" ),

// Assume jQuery is ready without the ready module

isReady: true,

error: function( msg ) {

throw new Error( msg );

},

noop: function() {},

isPlainObject: function( obj ) {

var proto, Ctor;

// Detect obvious negatives

// Use toString instead of jQuery.type to catch host objects

if ( !obj || toString.call( obj ) !== "[object Object]" ) {

return false;

}

proto = getProto( obj );

// Objects with no prototype (e.g., `Object.create( null )`) are plain

if ( !proto ) {

return true;

}

// Objects with prototype are plain iff they were constructed by a global Object function

Ctor = hasOwn.call( proto, "constructor" ) && proto.constructor;

return typeof Ctor === "function" && fnToString.call( Ctor ) === ObjectFunctionString;

},

isEmptyObject: function( obj ) {

var name;

for ( name in obj ) {

return false;

}

return true;

},

// Evaluates a script in a provided context; falls back to the global one

// if not specified.

globalEval: function( code, options, doc ) {

DOMEval( code, { nonce: options && options.nonce }, doc );

},

each: function( obj, callback ) {

var length, i = 0;

if ( isArrayLike( obj ) ) {

length = obj.length;

for ( ; i < length; i++ ) {

if ( callback.call( obj[ i ], i, obj[ i ] ) === false ) {

break;

}

}

} else {

for ( i in obj ) {

if ( callback.call( obj[ i ], i, obj[ i ] ) === false ) {

break;

}

}

}

return obj;

},

// results is for internal usage only

makeArray: function( arr, results ) {

var ret = results || [];

if ( arr != null ) {

if ( isArrayLike( Object( arr ) ) ) {

jQuery.merge( ret,

typeof arr === "string" ?

[ arr ] : arr

);

} else {

push.call( ret, arr );

}

}

return ret;

},

inArray: function( elem, arr, i ) {

return arr == null ? -1 : indexOf.call( arr, elem, i );

},

// Support: Android <=4.0 only, PhantomJS 1 only

// push.apply(\_, arraylike) throws on ancient WebKit

merge: function( first, second ) {

var len = +second.length,

j = 0,

i = first.length;

for ( ; j < len; j++ ) {

first[ i++ ] = second[ j ];

}

first.length = i;

return first;

},

grep: function( elems, callback, invert ) {

var callbackInverse,

matches = [],

i = 0,

length = elems.length,

callbackExpect = !invert;

// Go through the array, only saving the items

// that pass the validator function

for ( ; i < length; i++ ) {

callbackInverse = !callback( elems[ i ], i );

if ( callbackInverse !== callbackExpect ) {

matches.push( elems[ i ] );

}

}

return matches;

},

// arg is for internal usage only

map: function( elems, callback, arg ) {

var length, value,

i = 0,

ret = [];

// Go through the array, translating each of the items to their new values

if ( isArrayLike( elems ) ) {

length = elems.length;

for ( ; i < length; i++ ) {

value = callback( elems[ i ], i, arg );

if ( value != null ) {

ret.push( value );

}

}

// Go through every key on the object,

} else {

for ( i in elems ) {

value = callback( elems[ i ], i, arg );

if ( value != null ) {

ret.push( value );

}

}

}

// Flatten any nested arrays

return flat( ret );

},

// A global GUID counter for objects

guid: 1,

// jQuery.support is not used in Core but other projects attach their

// properties to it so it needs to exist.

support: support

} );

if ( typeof Symbol === "function" ) {

jQuery.fn[ Symbol.iterator ] = arr[ Symbol.iterator ];

}

// Populate the class2type map

jQuery.each( "Boolean Number String Function Array Date RegExp Object Error Symbol".split( " " ),

function( \_i, name ) {

class2type[ "[object " + name + "]" ] = name.toLowerCase();

} );

function isArrayLike( obj ) {

// Support: real iOS 8.2 only (not reproducible in simulator)

// `in` check used to prevent JIT error (gh-2145)

// hasOwn isn't used here due to false negatives

// regarding Nodelist length in IE

var length = !!obj && "length" in obj && obj.length,

type = toType( obj );

if ( isFunction( obj ) || isWindow( obj ) ) {

return false;

}

return type === "array" || length === 0 ||

typeof length === "number" && length > 0 && ( length - 1 ) in obj;

}

var Sizzle =

/\*!

\* Sizzle CSS Selector Engine v2.3.5

\* https://sizzlejs.com/

\*

\* Copyright JS Foundation and other contributors

\* Released under the MIT license

\* https://js.foundation/

\*

\* Date: 2020-03-14

\*/

( function( window ) {

var i,

support,

Expr,

getText,

isXML,

tokenize,

compile,

select,

outermostContext,

sortInput,

hasDuplicate,

// Local document vars

setDocument,

document,

docElem,

documentIsHTML,

rbuggyQSA,

rbuggyMatches,

matches,

contains,

// Instance-specific data

expando = "sizzle" + 1 \* new Date(),

preferredDoc = window.document,

dirruns = 0,

done = 0,

classCache = createCache(),

tokenCache = createCache(),

compilerCache = createCache(),

nonnativeSelectorCache = createCache(),

sortOrder = function( a, b ) {

if ( a === b ) {

hasDuplicate = true;

}

return 0;

},

// Instance methods

hasOwn = ( {} ).hasOwnProperty,

arr = [],

pop = arr.pop,

pushNative = arr.push,

push = arr.push,

slice = arr.slice,

// Use a stripped-down indexOf as it's faster than native

// https://jsperf.com/thor-indexof-vs-for/5

indexOf = function( list, elem ) {

var i = 0,

len = list.length;

for ( ; i < len; i++ ) {

if ( list[ i ] === elem ) {

return i;

}

}

return -1;

},

booleans = "checked|selected|async|autofocus|autoplay|controls|defer|disabled|hidden|" +

"ismap|loop|multiple|open|readonly|required|scoped",

// Regular expressions

// http://www.w3.org/TR/css3-selectors/#whitespace

whitespace = "[\\x20\\t\\r\\n\\f]",

// https://www.w3.org/TR/css-syntax-3/#ident-token-diagram

identifier = "(?:\\\\[\\da-fA-F]{1,6}" + whitespace +

"?|\\\\[^\\r\\n\\f]|[\\w-]|[^\0-\\x7f])+",

// Attribute selectors: http://www.w3.org/TR/selectors/#attribute-selectors

attributes = "\\[" + whitespace + "\*(" + identifier + ")(?:" + whitespace +

// Operator (capture 2)

"\*([\*^$|!~]?=)" + whitespace +

// "Attribute values must be CSS identifiers [capture 5]

// or strings [capture 3 or capture 4]"

"\*(?:'((?:\\\\.|[^\\\\'])\*)'|\"((?:\\\\.|[^\\\\\"])\*)\"|(" + identifier + "))|)" +

whitespace + "\*\\]",

pseudos = ":(" + identifier + ")(?:\\((" +

// To reduce the number of selectors needing tokenize in the preFilter, prefer arguments:

// 1. quoted (capture 3; capture 4 or capture 5)

"('((?:\\\\.|[^\\\\'])\*)'|\"((?:\\\\.|[^\\\\\"])\*)\")|" +

// 2. simple (capture 6)

"((?:\\\\.|[^\\\\()[\\]]|" + attributes + ")\*)|" +

// 3. anything else (capture 2)

".\*" +

")\\)|)",

// Leading and non-escaped trailing whitespace, capturing some non-whitespace characters preceding the latter

rwhitespace = new RegExp( whitespace + "+", "g" ),

rtrim = new RegExp( "^" + whitespace + "+|((?:^|[^\\\\])(?:\\\\.)\*)" +

whitespace + "+$", "g" ),

rcomma = new RegExp( "^" + whitespace + "\*," + whitespace + "\*" ),

rcombinators = new RegExp( "^" + whitespace + "\*([>+~]|" + whitespace + ")" + whitespace +

"\*" ),

rdescend = new RegExp( whitespace + "|>" ),

rpseudo = new RegExp( pseudos ),

ridentifier = new RegExp( "^" + identifier + "$" ),

matchExpr = {

"ID": new RegExp( "^#(" + identifier + ")" ),

"CLASS": new RegExp( "^\\.(" + identifier + ")" ),

"TAG": new RegExp( "^(" + identifier + "|[\*])" ),

"ATTR": new RegExp( "^" + attributes ),

"PSEUDO": new RegExp( "^" + pseudos ),

"CHILD": new RegExp( "^:(only|first|last|nth|nth-last)-(child|of-type)(?:\\(" +

whitespace + "\*(even|odd|(([+-]|)(\\d\*)n|)" + whitespace + "\*(?:([+-]|)" +

whitespace + "\*(\\d+)|))" + whitespace + "\*\\)|)", "i" ),

"bool": new RegExp( "^(?:" + booleans + ")$", "i" ),

// For use in libraries implementing .is()

// We use this for POS matching in `select`

"needsContext": new RegExp( "^" + whitespace +

"\*[>+~]|:(even|odd|eq|gt|lt|nth|first|last)(?:\\(" + whitespace +

"\*((?:-\\d)?\\d\*)" + whitespace + "\*\\)|)(?=[^-]|$)", "i" )

},

rhtml = /HTML$/i,

rinputs = /^(?:input|select|textarea|button)$/i,

rheader = /^h\d$/i,

rnative = /^[^{]+\{\s\*\[native \w/,

// Easily-parseable/retrievable ID or TAG or CLASS selectors

rquickExpr = /^(?:#([\w-]+)|(\w+)|\.([\w-]+))$/,

rsibling = /[+~]/,

// CSS escapes

// http://www.w3.org/TR/CSS21/syndata.html#escaped-characters

runescape = new RegExp( "\\\\[\\da-fA-F]{1,6}" + whitespace + "?|\\\\([^\\r\\n\\f])", "g" ),

funescape = function( escape, nonHex ) {

var high = "0x" + escape.slice( 1 ) - 0x10000;

return nonHex ?

// Strip the backslash prefix from a non-hex escape sequence

nonHex :

// Replace a hexadecimal escape sequence with the encoded Unicode code point

// Support: IE <=11+

// For values outside the Basic Multilingual Plane (BMP), manually construct a

// surrogate pair

high < 0 ?

String.fromCharCode( high + 0x10000 ) :

String.fromCharCode( high >> 10 | 0xD800, high & 0x3FF | 0xDC00 );

},

// CSS string/identifier serialization

// https://drafts.csswg.org/cssom/#common-serializing-idioms

rcssescape = /([\0-\x1f\x7f]|^-?\d)|^-$|[^\0-\x1f\x7f-\uFFFF\w-]/g,

fcssescape = function( ch, asCodePoint ) {

if ( asCodePoint ) {

// U+0000 NULL becomes U+FFFD REPLACEMENT CHARACTER

if ( ch === "\0" ) {

return "\uFFFD";

}

// Control characters and (dependent upon position) numbers get escaped as code points

return ch.slice( 0, -1 ) + "\\" +

ch.charCodeAt( ch.length - 1 ).toString( 16 ) + " ";

}

// Other potentially-special ASCII characters get backslash-escaped

return "\\" + ch;

},

// Used for iframes

// See setDocument()

// Removing the function wrapper causes a "Permission Denied"

// error in IE

unloadHandler = function() {

setDocument();

},

inDisabledFieldset = addCombinator(

function( elem ) {

return elem.disabled === true && elem.nodeName.toLowerCase() === "fieldset";

},

{ dir: "parentNode", next: "legend" }

);

// Optimize for push.apply( \_, NodeList )

try {

push.apply(

( arr = slice.call( preferredDoc.childNodes ) ),

preferredDoc.childNodes

);

// Support: Android<4.0

// Detect silently failing push.apply

// eslint-disable-next-line no-unused-expressions

arr[ preferredDoc.childNodes.length ].nodeType;

} catch ( e ) {

push = { apply: arr.length ?

// Leverage slice if possible

function( target, els ) {

pushNative.apply( target, slice.call( els ) );

} :

// Support: IE<9

// Otherwise append directly

function( target, els ) {

var j = target.length,

i = 0;

// Can't trust NodeList.length

while ( ( target[ j++ ] = els[ i++ ] ) ) {}

target.length = j - 1;

}

};

}

function Sizzle( selector, context, results, seed ) {

var m, i, elem, nid, match, groups, newSelector,

newContext = context && context.ownerDocument,

// nodeType defaults to 9, since context defaults to document

nodeType = context ? context.nodeType : 9;

results = results || [];

// Return early from calls with invalid selector or context

if ( typeof selector !== "string" || !selector ||

nodeType !== 1 && nodeType !== 9 && nodeType !== 11 ) {

return results;

}

// Try to shortcut find operations (as opposed to filters) in HTML documents

if ( !seed ) {

setDocument( context );

context = context || document;

if ( documentIsHTML ) {

// If the selector is sufficiently simple, try using a "get\*By\*" DOM method

// (excepting DocumentFragment context, where the methods don't exist)

if ( nodeType !== 11 && ( match = rquickExpr.exec( selector ) ) ) {

// ID selector

if ( ( m = match[ 1 ] ) ) {

// Document context

if ( nodeType === 9 ) {

if ( ( elem = context.getElementById( m ) ) ) {

// Support: IE, Opera, Webkit

// TODO: identify versions

// getElementById can match elements by name instead of ID

if ( elem.id === m ) {

results.push( elem );

return results;

}

} else {

return results;

}

// Element context

} else {

// Support: IE, Opera, Webkit

// TODO: identify versions

// getElementById can match elements by name instead of ID

if ( newContext && ( elem = newContext.getElementById( m ) ) &&

contains( context, elem ) &&

elem.id === m ) {

results.push( elem );

return results;

}

}

// Type selector

} else if ( match[ 2 ] ) {

push.apply( results, context.getElementsByTagName( selector ) );

return results;

// Class selector

} else if ( ( m = match[ 3 ] ) && support.getElementsByClassName &&

context.getElementsByClassName ) {

push.apply( results, context.getElementsByClassName( m ) );

return results;

}

}

// Take advantage of querySelectorAll

if ( support.qsa &&

!nonnativeSelectorCache[ selector + " " ] &&

( !rbuggyQSA || !rbuggyQSA.test( selector ) ) &&

// Support: IE 8 only

// Exclude object elements

( nodeType !== 1 || context.nodeName.toLowerCase() !== "object" ) ) {

newSelector = selector;

newContext = context;

// qSA considers elements outside a scoping root when evaluating child or

// descendant combinators, which is not what we want.

// In such cases, we work around the behavior by prefixing every selector in the

// list with an ID selector referencing the scope context.

// The technique has to be used as well when a leading combinator is used

// as such selectors are not recognized by querySelectorAll.

// Thanks to Andrew Dupont for this technique.

if ( nodeType === 1 &&

( rdescend.test( selector ) || rcombinators.test( selector ) ) ) {

// Expand context for sibling selectors

newContext = rsibling.test( selector ) && testContext( context.parentNode ) ||

context;

// We can use :scope instead of the ID hack if the browser

// supports it & if we're not changing the context.

if ( newContext !== context || !support.scope ) {

// Capture the context ID, setting it first if necessary

if ( ( nid = context.getAttribute( "id" ) ) ) {

nid = nid.replace( rcssescape, fcssescape );

} else {

context.setAttribute( "id", ( nid = expando ) );

}

}

// Prefix every selector in the list

groups = tokenize( selector );

i = groups.length;

while ( i-- ) {

groups[ i ] = ( nid ? "#" + nid : ":scope" ) + " " +

toSelector( groups[ i ] );

}

newSelector = groups.join( "," );

}

try {

push.apply( results,

newContext.querySelectorAll( newSelector )

);

return results;

} catch ( qsaError ) {

nonnativeSelectorCache( selector, true );

} finally {

if ( nid === expando ) {

context.removeAttribute( "id" );

}

}

}

}

}

// All others

return select( selector.replace( rtrim, "$1" ), context, results, seed );

}

/\*\*

\* Create key-value caches of limited size

\* @returns {function(string, object)} Returns the Object data after storing it on itself with

\* property name the (space-suffixed) string and (if the cache is larger than Expr.cacheLength)

\* deleting the oldest entry

\*/

function createCache() {

var keys = [];

function cache( key, value ) {

// Use (key + " ") to avoid collision with native prototype properties (see Issue #157)

if ( keys.push( key + " " ) > Expr.cacheLength ) {

// Only keep the most recent entries

delete cache[ keys.shift() ];

}

return ( cache[ key + " " ] = value );

}

return cache;

}

/\*\*

\* Mark a function for special use by Sizzle

\* @param {Function} fn The function to mark

\*/

function markFunction( fn ) {

fn[ expando ] = true;

return fn;

}

/\*\*

\* Support testing using an element

\* @param {Function} fn Passed the created element and returns a boolean result

\*/

function assert( fn ) {

var el = document.createElement( "fieldset" );

try {

return !!fn( el );

} catch ( e ) {

return false;

} finally {

// Remove from its parent by default

if ( el.parentNode ) {

el.parentNode.removeChild( el );

}

// release memory in IE

el = null;

}

}

/\*\*

\* Adds the same handler for all of the specified attrs

\* @param {String} attrs Pipe-separated list of attributes

\* @param {Function} handler The method that will be applied

\*/

function addHandle( attrs, handler ) {

var arr = attrs.split( "|" ),

i = arr.length;

while ( i-- ) {

Expr.attrHandle[ arr[ i ] ] = handler;

}

}

/\*\*

\* Checks document order of two siblings

\* @param {Element} a

\* @param {Element} b

\* @returns {Number} Returns less than 0 if a precedes b, greater than 0 if a follows b

\*/

function siblingCheck( a, b ) {

var cur = b && a,

diff = cur && a.nodeType === 1 && b.nodeType === 1 &&

a.sourceIndex - b.sourceIndex;

// Use IE sourceIndex if available on both nodes

if ( diff ) {

return diff;

}

// Check if b follows a

if ( cur ) {

while ( ( cur = cur.nextSibling ) ) {

if ( cur === b ) {

return -1;

}

}

}

return a ? 1 : -1;

}

/\*\*

\* Returns a function to use in pseudos for input types

\* @param {String} type

\*/

function createInputPseudo( type ) {

return function( elem ) {

var name = elem.nodeName.toLowerCase();

return name === "input" && elem.type === type;

};

}

/\*\*

\* Returns a function to use in pseudos for buttons

\* @param {String} type

\*/

function createButtonPseudo( type ) {

return function( elem ) {

var name = elem.nodeName.toLowerCase();

return ( name === "input" || name === "button" ) && elem.type === type;

};

}