var express = require('express');

var router = express.Router();

/\* GET users listing. \*/

router.get('/', function(req, res, next) {

res.send('respond with a resource');

});

module.exports = router;

var express = require('express');

var router = express.Router();

var ejs = require('ejs');

var path = require('path');

var Web3 = require('web3');

var web3 = new Web3(new Web3.providers.HttpProvider('http://localhost:8545'));

web3.setProvider(new Web3.providers.WebsocketProvider('ws://localhost:8545'));

// web3.setProvider(new Web3.providers.HttpProvider('http://localhost:8545'));

var contractAddress = "0xD6FE0a62C403D333f92508F91bbADAA86BF86493";

var mycontract = new web3.eth.Contract([

{

"inputs": [],

"stateMutability": "nonpayable",

"type": "constructor"

},

{

"anonymous": false,

"inputs": [

{

"indexed": false,

"internalType": "bool",

"name": "success",

"type": "bool"

}

],

"name": "registerSuccess",

"type": "event"

},

{

"anonymous": false,

"inputs": [

{

"indexed": false,

"internalType": "bool",

"name": "success",

"type": "bool"

}

],

"name": "verifySuccess",

"type": "event"

},

{

"inputs": [],

"name": "ID",

"outputs": [

{

"internalType": "uint256",

"name": "",

"type": "uint256"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "a",

"type": "string"

},

{

"internalType": "string",

"name": "b",

"type": "string"

}

],

"name": "comString",

"outputs": [

{

"internalType": "bool",

"name": "",

"type": "bool"

}

],

"stateMutability": "pure",

"type": "function"

},

{

"inputs": [],

"name": "getUserInfo",

"outputs": [

{

"internalType": "string",

"name": "",

"type": "string"

},

{

"internalType": "uint256",

"name": "",

"type": "uint256"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "username",

"type": "string"

},

{

"internalType": "string",

"name": "passwordKey",

"type": "string"

}

],

"name": "login",

"outputs": [

{

"internalType": "bool",

"name": "",

"type": "bool"

}

],

"stateMutability": "nonpayable",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "username",

"type": "string"

},

{

"internalType": "string",

"name": "passwordKey",

"type": "string"

}

],

"name": "registerAccount",

"outputs": [

{

"internalType": "bool",

"name": "",

"type": "bool"

}

],

"stateMutability": "nonpayable",

"type": "function"

}

],

contractAddress

);

/\* GET home page. \*/

router.get('/', function (req, res, next) {

res.type('html');

res.render('/index');

});

router.post('/register', function (req, res, next) {

var { username, password, account } = req.body;

console.log(req.body);

console.log(username, password);

mycontract.methods.registerAccount(username, password)

.send({ from: account, gas: '6721975' })

.then(function (receipt) {

console.log(receipt);

res.json();

// // 监听事件

// mycontract.events.registerSuccess({

// filter: {}, // Using an array means OR: e.g. 20 or 23

// fromBlock: 0

// }, function (error, event) { console.log(event); })

// .on("data", function (event) {

// let rs = event.returnValues;

// var msg;

// console.log("注册成功：", rs.success);

// if (rs.success) {

// msg = '注册成功';

// } else {

// msg = '账户已经注册，请勿重复注册';

// }

// return res.status(200).json({ 'msg': msg });

// })

// .on('error', function (error, receipt) {

// console.log(error);

// });

});

});

router.post('/login', function (req, res, next) {

let { username, password, account } = req.body;

console.log(req.body);

console.log(username, password);

mycontract.methods.login(username, password).send({ from: account }).then(function (receipt) {

console.log(receipt);

res.json({});

// 监听事件

// mycontract.events.verifySuccess({

// filter: {}, // Using an array means OR: e.g. 20 or 23

// fromBlock: 0

// }, function (error, event) { console.log(event); })

// .on("data", function (event) {

// let vs = event.returnValues;

// console.log("登录成功：", vs.success);

// var jump;

// if (!vs.success) {

// jump = false;

// } else {

// jump = true;

// }

// return res.json({ 'msg': 'success', 'jump': jump });

// })

// .on('error', function (error, receipt) {

// console.log(error, receipt);

// });

});

});

router.get('/user/:account', function (req, res, next) {

var account = req.params.account;

mycontract.methods.getUserInfo()

.call({ from: account })

.then(function (result) {

console.log(result);

var username = result[0];

var ID = result[1];

console.log(username, ID);

res.type('html');

res.render('user', { username: username, uid: ID });

});

});

module.exports = router;

{

"name": "dapp",

"version": "0.0.0",

"private": true,

"scripts": {

"start": "node ./bin/www"

},

"dependencies": {

"cookie-parser": "~1.4.4",

"debug": "~2.6.9",

"ejs": "~2.6.1",

"express": "~4.16.1",

"http-errors": "~1.6.3",

"morgan": "~1.9.1"

}

}

var createError = require('http-errors');

var express = require('express');

var path = require('path');

var cookieParser = require('cookie-parser');

var logger = require('morgan');

var indexRouter = require('./routes/index');

var usersRouter = require('./routes/users');

var app = express();

app.use(express.json());

app.use(express.urlencoded({ extended: false }));

// view engine setup

app.set('views', path.join(\_\_dirname, 'views'));

app.set('view engine', 'ejs');

app.use(logger('dev'));

app.use(express.json());

app.use(express.urlencoded({ extended: false }));

app.use(cookieParser());

app.use(express.static(path.join(\_\_dirname, 'public')));

app.use('/', indexRouter);

app.use('/users', usersRouter);

// catch 404 and forward to error handler

app.use(function (req, res, next) {

next(createError(404));

});

// error handler

app.use(function (err, req, res, next) {

// set locals, only providing error in development

res.locals.message = err.message;

res.locals.error = req.app.get('env') === 'development' ? err : {};

// render the error page

res.status(err.status || 500);

res.render('error');

});

module.exports = app;

#!/usr/bin/env node

/\*\*

\* Module dependencies.

\*/

var app = require('../app');

var debug = require('debug')('dapp:server');

var http = require('http');

/\*\*

\* Get port from environment and store in Express.

\*/

var port = normalizePort(process.env.PORT || '3000');

app.set('port', port);

/\*\*

\* Create HTTP server.

\*/

var server = http.createServer(app);

/\*\*

\* Listen on provided port, on all network interfaces.

\*/

server.listen(port);

server.on('error', onError);

server.on('listening', onListening);

/\*\*

\* Normalize a port into a number, string, or false.

\*/

function normalizePort(val) {

var port = parseInt(val, 10);

if (isNaN(port)) {

// named pipe

return val;

}

if (port >= 0) {

// port number

return port;

}

return false;

}

/\*\*

\* Event listener for HTTP server "error" event.

\*/

function onError(error) {

if (error.syscall !== 'listen') {

throw error;

}

var bind = typeof port === 'string'

? 'Pipe ' + port

: 'Port ' + port;

// handle specific listen errors with friendly messages

switch (error.code) {

case 'EACCES':

console.error(bind + ' requires elevated privileges');

process.exit(1);

break;

case 'EADDRINUSE':

console.error(bind + ' is already in use');

process.exit(1);

break;

default:

throw error;

}

}

/\*\*

\* Event listener for HTTP server "listening" event.

\*/

function onListening() {

var addr = server.address();

var bind = typeof addr === 'string'

? 'pipe ' + addr

: 'port ' + addr.port;

debug('Listening on ' + bind);

}

var Web3 = require('web3');

var web3 = new Web3(new Web3.providers.HttpProvider('http://localhost:8545'));

var addressDeploy = "0x28Be2D380Ff7525915E5c50655bC696460084C14";

web3.eth.getAccounts().then(console.log);

console.log(addressDeploy);

var mycounterContract = new web3.eth.Contract([{ "inputs": [], "stateMutability": "nonpayable", "type": "constructor" }, { "inputs": [], "name": "getCounter", "outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }], "stateMutability": "view", "type": "function" }, { "inputs": [{ "internalType": "uint256", "name": "step", "type": "uint256" }], "name": "stepCounter", "outputs": [], "stateMutability": "nonpayable", "type": "function" }]);

var mycounter = mycounterContract.deploy({

data: '',

arguments: [

]

}).send({

from: addressDeploy, // 部署合约的账户地址

gas: '6721975'

}, function (e, contract) {

console.log(e, contract);

if (typeof contract.address !== 'undefined') {

console.log('Contract mined! address: ' + contract.address + ' transactionHash: ' + contract.transactionHash);

}

}).then(function (contract) {

console.log("Contract Address: ", contract.options.address);

// var fromAccount = addressDeploy

// var toAccount = contract.options.address;

// var amount = 50;

// // 对输入的数字做一个检查

// if (web3.utils.isAddress(fromAccount) &&

// web3.utils.isAddress(toAccount) &&

// amount != null && amount.length > 0) {

// var message = { from: fromAccount, to: toAccount, value: web3.utils.toWei(amount, 'ether') };

// web3.eth.sendTransaction(message);

// }

})

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>基于区块链的身份认证系统</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<meta http-equiv="Access-Control-Allow-Origin" content="\*">

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">

<meta name="apple-mobile-web-app-status-bar-style" content="black">

<meta name="apple-mobile-web-app-capable" content="yes">

<meta name="format-detection" content="telephone=no">

<link rel="stylesheet" href="/stylesheets/bootstrap.min.css" type="text/css">

<script src="/javascripts/jquery-3.5.1.js"></script>

<script>

$(function () {

// alert(SHA256("username"));

linkWallet();

getAccount();

// 点击注册按钮

$('#register').click(function () {

var username = $('#username').val();

var password = $('#password').val();

if (username && password) {

$('#password').val(SHA256(password));

$.post('/register', $('#userInfo').serialize(), function (data) {

// alert(data.msg);

});

} else {

alert('用户名和密码不能为空！');

}

});

$('#login').click(function () {

var username = $('#username').val();

var password = $('#password').val();

if (username && password) {

$('#password').val(SHA256(password));

$.post('/login', $('#userInfo').serialize(), function (data) {

// alert(data.msg, data.jump);

// if (data.jump) {

// location.href = '/user:' + $('#account').val();

// }

});

} else {

alert('用户名和密码不能为空！');

}

})

});

</script>

<style>

</style>

</head>

<body>

<div class="container">

<div class="row">

<div class="col-lg-offset-3 col-lg-6">

<ol class="breadcrumb">

<li><a href="#">登录</a></li>

<li><a href="#">身份认证</a></li>

</ol>

</div>

</div>

<div class="row">

<div class="col-lg-offset-4 col-lg-4">

<form id="userInfo">

<div class="form-group">

<label for="username">用户名</label>

<input type="text" name="username" class="form-control" id="username" placeholder="密码">

</div>

<div class="form-group">

<label for="password">密码</label>

<input type="password" name="password" class="form-control" id="password" placeholder="密码">

</div>

<input id="account" type="hidden" name="account" />

<div class="form-inline">

<button type="button" class="btn btn-primary" id="register">注册</button>

<button type="button" class="btn btn-primary" id="login">登录</button>

</div>

</form>

</div>

</div>

</div>

</body>

<script src="https://cdn.jsdelivr.net/npm/web3@latest/dist/web3.min.js" type="text/javascript" charset="utf-8">

</script>

<script src="javascripts/sha-256.js"></script>

<script type="text/javascript">

if (typeof web3 !== 'undefined') {

web3 = new Web3(web3.currentProvider);

} else {

// set the provider you want from Web3.providers

web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

}

var web3Provider;

var web3js;

function linkWallet() {

if (ethereum) {

web3Provider = ethereum;

// 新版需要请求用户授权

try {

ethereum.enable();

} catch (error) {

alert("用户取消授权");

return;

}

} else if (web3) {

// MetaMask Legacy dapp browsers...

web3Provider = web3.currentProvider;

console.log("web3.currentProvider:");

console.log(web3.currentProvider);

} else {

web3Provider = new Web3.providers.HttpProvider('http://localhost:8545');

console.log("https://http-testnet.hecochain.com");

}

web3js = new Web3(web3Provider);

//document.getElementById("btn-wallet").innerText = "重新连接";

}

function getAccount() {

if (!web3js) {

document.getElementById("account").value = "请先连接 wallet";

return;

}

web3js.eth.getAccounts(function (error, result) {

if (!error) {

document.getElementById("account").value = result;

} else {

document.getElementById("account").value = "获取地址失败";

}

});

}

var contractAddress = '0xD6FE0a62C403D333f92508F91bbADAA86BF86493';

var infoContract = MyContract = new web3.eth.Contract([

{

"inputs": [],

"stateMutability": "nonpayable",

"type": "constructor"

},

{

"anonymous": false,

"inputs": [

{

"indexed": false,

"internalType": "bool",

"name": "success",

"type": "bool"

}

],

"name": "registerSuccess",

"type": "event"

},

{

"anonymous": false,

"inputs": [

{

"indexed": false,

"internalType": "bool",

"name": "success",

"type": "bool"

}

],

"name": "verifySuccess",

"type": "event"

},

{

"inputs": [],

"name": "ID",

"outputs": [

{

"internalType": "uint256",

"name": "",

"type": "uint256"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "a",

"type": "string"

},

{

"internalType": "string",

"name": "b",

"type": "string"

}

],

"name": "comString",

"outputs": [

{

"internalType": "bool",

"name": "",

"type": "bool"

}

],

"stateMutability": "pure",

"type": "function"

},

{

"inputs": [],

"name": "getUserInfo",

"outputs": [

{

"internalType": "string",

"name": "",

"type": "string"

},

{

"internalType": "uint256",

"name": "",

"type": "uint256"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "username",

"type": "string"

},

{

"internalType": "string",

"name": "passwordKey",

"type": "string"

}

],

"name": "login",

"outputs": [

{

"internalType": "bool",

"name": "",

"type": "bool"

}

],

"stateMutability": "nonpayable",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "username",

"type": "string"

},

{

"internalType": "string",

"name": "passwordKey",

"type": "string"

}

],

"name": "registerAccount",

"outputs": [

{

"internalType": "bool",

"name": "",

"type": "bool"

}

],

"stateMutability": "nonpayable",

"type": "function"

}

], contractAddress);

// var info = infoContract.at('0x0eBF143bBfA48b98a8749c6A945E632d8EAD4Ff1');

// var registerEvent =

infoContract.events.registerSuccess(function (error, result) {

if (!error) {

if (result.returnValues.success) {

alert('注册成功');

} else {

alert('账号已注册，请勿重复注册');

}

location.href = '/';

} else {

console.log(error);

}

});

// var loginEvent =

infoContract.events.verifySuccess(function (error, result) {

if (!error) {

console.log(result.returnValues.success);

if (result.returnValues.success) {

alert('登录成功');

location.href = '/user/' + $('#account').val();

} else {

alert('用户名或密码错误');

location.href = '/';

}

} else {

console.log(error);

}

});

</script>

</html>

/\*\*

\*

\* Secure Hash Algorithm (SHA256)

\* http://www.webtoolkit.info/

\*

\* Original code by Angel Marin, Paul Johnston.

\*

\*\*/

function SHA256(s) {

var chrsz = 8;

var hexcase = 0;

function safe\_add(x, y) {

var lsw = (x & 0xFFFF) + (y & 0xFFFF);

var msw = (x >> 16) + (y >> 16) + (lsw >> 16);

return (msw << 16) | (lsw & 0xFFFF);

}

function S(X, n) { return (X >>> n) | (X << (32 - n)); }

function R(X, n) { return (X >>> n); }

function Ch(x, y, z) { return ((x & y) ^ ((~x) & z)); }

function Maj(x, y, z) { return ((x & y) ^ (x & z) ^ (y & z)); }

function Sigma0256(x) { return (S(x, 2) ^ S(x, 13) ^ S(x, 22)); }

function Sigma1256(x) { return (S(x, 6) ^ S(x, 11) ^ S(x, 25)); }

function Gamma0256(x) { return (S(x, 7) ^ S(x, 18) ^ R(x, 3)); }

function Gamma1256(x) { return (S(x, 17) ^ S(x, 19) ^ R(x, 10)); }

function core\_sha256(m, l) {

var K = new Array(0x428A2F98, 0x71374491, 0xB5C0FBCF, 0xE9B5DBA5, 0x3956C25B, 0x59F111F1, 0x923F82A4, 0xAB1C5ED5, 0xD807AA98, 0x12835B01, 0x243185BE, 0x550C7DC3, 0x72BE5D74, 0x80DEB1FE, 0x9BDC06A7, 0xC19BF174, 0xE49B69C1, 0xEFBE4786, 0xFC19DC6, 0x240CA1CC, 0x2DE92C6F, 0x4A7484AA, 0x5CB0A9DC, 0x76F988DA, 0x983E5152, 0xA831C66D, 0xB00327C8, 0xBF597FC7, 0xC6E00BF3, 0xD5A79147, 0x6CA6351, 0x14292967, 0x27B70A85, 0x2E1B2138, 0x4D2C6DFC, 0x53380D13, 0x650A7354, 0x766A0ABB, 0x81C2C92E, 0x92722C85, 0xA2BFE8A1, 0xA81A664B, 0xC24B8B70, 0xC76C51A3, 0xD192E819, 0xD6990624, 0xF40E3585, 0x106AA070, 0x19A4C116, 0x1E376C08, 0x2748774C, 0x34B0BCB5, 0x391C0CB3, 0x4ED8AA4A, 0x5B9CCA4F, 0x682E6FF3, 0x748F82EE, 0x78A5636F, 0x84C87814, 0x8CC70208, 0x90BEFFFA, 0xA4506CEB, 0xBEF9A3F7, 0xC67178F2);

var HASH = new Array(0x6A09E667, 0xBB67AE85, 0x3C6EF372, 0xA54FF53A, 0x510E527F, 0x9B05688C, 0x1F83D9AB, 0x5BE0CD19);

var W = new Array(64);

var a, b, c, d, e, f, g, h, i, j;

var T1, T2;

m[l >> 5] |= 0x80 << (24 - l % 32);

m[((l + 64 >> 9) << 4) + 15] = l;

for (var i = 0; i < m.length; i += 16) {

a = HASH[0];

b = HASH[1];

c = HASH[2];

d = HASH[3];

e = HASH[4];

f = HASH[5];

g = HASH[6];

h = HASH[7];

for (var j = 0; j < 64; j++) {

if (j < 16) W[j] = m[j + i];

else W[j] = safe\_add(safe\_add(safe\_add(Gamma1256(W[j - 2]), W[j - 7]), Gamma0256(W[j - 15])), W[j - 16]);

T1 = safe\_add(safe\_add(safe\_add(safe\_add(h, Sigma1256(e)), Ch(e, f, g)), K[j]), W[j]);

T2 = safe\_add(Sigma0256(a), Maj(a, b, c));

h = g;

g = f;

f = e;

e = safe\_add(d, T1);

d = c;

c = b;

b = a;

a = safe\_add(T1, T2);

}

HASH[0] = safe\_add(a, HASH[0]);

HASH[1] = safe\_add(b, HASH[1]);

HASH[2] = safe\_add(c, HASH[2]);

HASH[3] = safe\_add(d, HASH[3]);

HASH[4] = safe\_add(e, HASH[4]);

HASH[5] = safe\_add(f, HASH[5]);

HASH[6] = safe\_add(g, HASH[6]);

HASH[7] = safe\_add(h, HASH[7]);

}

return HASH;

}

function str2binb(str) {

var bin = Array();

var mask = (1 << chrsz) - 1;

for (var i = 0; i < str.length \* chrsz; i += chrsz) {

bin[i >> 5] |= (str.charCodeAt(i / chrsz) & mask) << (24 - i % 32);

}

return bin;

}

function Utf8Encode(string) {

string = string.replace(/\r\n/g, "\n");

var utftext = "";

for (var n = 0; n < string.length; n++) {

var c = string.charCodeAt(n);

if (c < 128) {

utftext += String.fromCharCode(c);

}

else if ((c > 127) && (c < 2048)) {

utftext += String.fromCharCode((c >> 6) | 192);

utftext += String.fromCharCode((c & 63) | 128);

}

else {

utftext += String.fromCharCode((c >> 12) | 224);

utftext += String.fromCharCode(((c >> 6) & 63) | 128);

utftext += String.fromCharCode((c & 63) | 128);

}

}

return utftext;

}

function binb2hex(binarray) {

var hex\_tab = hexcase ? "0123456789ABCDEF" : "0123456789abcdef";

var str = "";

for (var i = 0; i < binarray.length \* 4; i++) {

str += hex\_tab.charAt((binarray[i >> 2] >> ((3 - i % 4) \* 8 + 4)) & 0xF) +

hex\_tab.charAt((binarray[i >> 2] >> ((3 - i % 4) \* 8)) & 0xF);

}

return str;

}

s = Utf8Encode(s);

return binb2hex(core\_sha256(str2binb(s), s.length \* chrsz));

}

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;

};

}

/\*\*

\* Returns a function to use in pseudos for :enabled/:disabled

\* @param {Boolean} disabled true for :disabled; false for :enabled

\*/

function createDisabledPseudo( disabled ) {

// Known :disabled false positives: fieldset[disabled] > legend:nth-of-type(n+2) :can-disable

return function( elem ) {

// Only certain elements can match :enabled or :disabled

// https://html.spec.whatwg.org/multipage/scripting.html#selector-enabled

// https://html.spec.whatwg.org/multipage/scripting.html#selector-disabled

if ( "form" in elem ) {

// Check for inherited disabledness on relevant non-disabled elements:

// \* listed form-associated elements in a disabled fieldset

// https://html.spec.whatwg.org/multipage/forms.html#category-listed

// https://html.spec.whatwg.org/multipage/forms.html#concept-fe-disabled

// \* option elements in a disabled optgroup

// https://html.spec.whatwg.org/multipage/forms.html#concept-option-disabled

// All such elements have a "form" property.

if ( elem.parentNode && elem.disabled === false ) {

// Option elements defer to a parent optgroup if present

if ( "label" in elem ) {

if ( "label" in elem.parentNode ) {

return elem.parentNode.disabled === disabled;

} else {

return elem.disabled === disabled;

}

}

// Support: IE 6 - 11

// Use the isDisabled shortcut property to check for disabled fieldset ancestors

return elem.isDisabled === disabled ||

// Where there is no isDisabled, check manually

/\* jshint -W018 \*/

elem.isDisabled !== !disabled &&

inDisabledFieldset( elem ) === disabled;

}

return elem.disabled === disabled;

// Try to winnow out elements that can't be disabled before trusting the disabled property.

// Some victims get caught in our net (label, legend, menu, track), but it shouldn't

// even exist on them, let alone have a boolean value.

} else if ( "label" in elem ) {

return elem.disabled === disabled;

}

// Remaining elements are neither :enabled nor :disabled

return false;

};

}

/\*\*

\* Returns a function to use in pseudos for positionals

\* @param {Function} fn

\*/

function createPositionalPseudo( fn ) {

return markFunction( function( argument ) {

argument = +argument;

return markFunction( function( seed, matches ) {

var j,

matchIndexes = fn( [], seed.length, argument ),

i = matchIndexes.length;

// Match elements found at the specified indexes

while ( i-- ) {

if ( seed[ ( j = matchIndexes[ i ] ) ] ) {

seed[ j ] = !( matches[ j ] = seed[ j ] );

}

}

} );

} );

}

/\*\*

\* Checks a node for validity as a Sizzle context

\* @param {Element|Object=} context

\* @returns {Element|Object|Boolean} The input node if acceptable, otherwise a falsy value

\*/

function testContext( context ) {

return context && typeof context.getElementsByTagName !== "undefined" && context;

}

// Expose support vars for convenience

support = Sizzle.support = {};

/\*\*

\* Detects XML nodes

\* @param {Element|Object} elem An element or a document

\* @returns {Boolean} True iff elem is a non-HTML XML node

\*/

isXML = Sizzle.isXML = function( elem ) {

var namespace = elem.namespaceURI,

docElem = ( elem.ownerDocument || elem ).documentElement;

// Support: IE <=8

// Assume HTML when documentElement doesn't yet exist, such as inside loading iframes

// https://bugs.jquery.com/ticket/4833

return !rhtml.test( namespace || docElem && docElem.nodeName || "HTML" );

};

/\*\*

\* Sets document-related variables once based on the current document

\* @param {Element|Object} [doc] An element or document object to use to set the document

\* @returns {Object} Returns the current document

\*/

setDocument = Sizzle.setDocument = function( node ) {

var hasCompare, subWindow,

doc = node ? node.ownerDocument || node : preferredDoc;

// Return early if doc is invalid or already selected

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( doc == document || doc.nodeType !== 9 || !doc.documentElement ) {

return document;

}

// Update global variables

document = doc;

docElem = document.documentElement;

documentIsHTML = !isXML( document );

// Support: IE 9 - 11+, Edge 12 - 18+

// Accessing iframe documents after unload throws "permission denied" errors (jQuery #13936)

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( preferredDoc != document &&

( subWindow = document.defaultView ) && subWindow.top !== subWindow ) {

// Support: IE 11, Edge

if ( subWindow.addEventListener ) {

subWindow.addEventListener( "unload", unloadHandler, false );

// Support: IE 9 - 10 only

} else if ( subWindow.attachEvent ) {

subWindow.attachEvent( "onunload", unloadHandler );

}

}

// Support: IE 8 - 11+, Edge 12 - 18+, Chrome <=16 - 25 only, Firefox <=3.6 - 31 only,

// Safari 4 - 5 only, Opera <=11.6 - 12.x only

// IE/Edge & older browsers don't support the :scope pseudo-class.

// Support: Safari 6.0 only

// Safari 6.0 supports :scope but it's an alias of :root there.

support.scope = assert( function( el ) {

docElem.appendChild( el ).appendChild( document.createElement( "div" ) );

return typeof el.querySelectorAll !== "undefined" &&

!el.querySelectorAll( ":scope fieldset div" ).length;

} );

/\* Attributes

---------------------------------------------------------------------- \*/

// Support: IE<8

// Verify that getAttribute really returns attributes and not properties

// (excepting IE8 booleans)

support.attributes = assert( function( el ) {

el.className = "i";

return !el.getAttribute( "className" );

} );

/\* getElement(s)By\*

---------------------------------------------------------------------- \*/

// Check if getElementsByTagName("\*") returns only elements

support.getElementsByTagName = assert( function( el ) {

el.appendChild( document.createComment( "" ) );

return !el.getElementsByTagName( "\*" ).length;

} );

// Support: IE<9

support.getElementsByClassName = rnative.test( document.getElementsByClassName );

// Support: IE<10

// Check if getElementById returns elements by name

// The broken getElementById methods don't pick up programmatically-set names,

// so use a roundabout getElementsByName test

support.getById = assert( function( el ) {

docElem.appendChild( el ).id = expando;

return !document.getElementsByName || !document.getElementsByName( expando ).length;

} );

// ID filter and find

if ( support.getById ) {

Expr.filter[ "ID" ] = function( id ) {

var attrId = id.replace( runescape, funescape );

return function( elem ) {

return elem.getAttribute( "id" ) === attrId;

};

};

Expr.find[ "ID" ] = function( id, context ) {

if ( typeof context.getElementById !== "undefined" && documentIsHTML ) {

var elem = context.getElementById( id );

return elem ? [ elem ] : [];

}

};

} else {

Expr.filter[ "ID" ] = function( id ) {

var attrId = id.replace( runescape, funescape );

return function( elem ) {

var node = typeof elem.getAttributeNode !== "undefined" &&

elem.getAttributeNode( "id" );

return node && node.value === attrId;

};

};

// Support: IE 6 - 7 only

// getElementById is not reliable as a find shortcut

Expr.find[ "ID" ] = function( id, context ) {

if ( typeof context.getElementById !== "undefined" && documentIsHTML ) {

var node, i, elems,

elem = context.getElementById( id );

if ( elem ) {

// Verify the id attribute

node = elem.getAttributeNode( "id" );

if ( node && node.value === id ) {

return [ elem ];

}

// Fall back on getElementsByName

elems = context.getElementsByName( id );

i = 0;

while ( ( elem = elems[ i++ ] ) ) {

node = elem.getAttributeNode( "id" );

if ( node && node.value === id ) {

return [ elem ];

}

}

}

return [];

}

};

}

// Tag

Expr.find[ "TAG" ] = support.getElementsByTagName ?

function( tag, context ) {

if ( typeof context.getElementsByTagName !== "undefined" ) {

return context.getElementsByTagName( tag );

// DocumentFragment nodes don't have gEBTN

} else if ( support.qsa ) {

return context.querySelectorAll( tag );

}

} :

function( tag, context ) {

var elem,

tmp = [],

i = 0,

// By happy coincidence, a (broken) gEBTN appears on DocumentFragment nodes too

results = context.getElementsByTagName( tag );

// Filter out possible comments

if ( tag === "\*" ) {

while ( ( elem = results[ i++ ] ) ) {

if ( elem.nodeType === 1 ) {

tmp.push( elem );

}

}

return tmp;

}

return results;

};

// Class

Expr.find[ "CLASS" ] = support.getElementsByClassName && function( className, context ) {

if ( typeof context.getElementsByClassName !== "undefined" && documentIsHTML ) {

return context.getElementsByClassName( className );

}

};

/\* QSA/matchesSelector

---------------------------------------------------------------------- \*/

// QSA and matchesSelector support

// matchesSelector(:active) reports false when true (IE9/Opera 11.5)

rbuggyMatches = [];

// qSa(:focus) reports false when true (Chrome 21)

// We allow this because of a bug in IE8/9 that throws an error

// whenever `document.activeElement` is accessed on an iframe

// So, we allow :focus to pass through QSA all the time to avoid the IE error

// See https://bugs.jquery.com/ticket/13378

rbuggyQSA = [];

if ( ( support.qsa = rnative.test( document.querySelectorAll ) ) ) {

// Build QSA regex

// Regex strategy adopted from Diego Perini

assert( function( el ) {

var input;

// Select is set to empty string on purpose

// This is to test IE's treatment of not explicitly

// setting a boolean content attribute,

// since its presence should be enough

// https://bugs.jquery.com/ticket/12359

docElem.appendChild( el ).innerHTML = "<a id='" + expando + "'></a>" +

"<select id='" + expando + "-\r\\' msallowcapture=''>" +

"<option selected=''></option></select>";

// Support: IE8, Opera 11-12.16

// Nothing should be selected when empty strings follow ^= or $= or \*=

// The test attribute must be unknown in Opera but "safe" for WinRT

// https://msdn.microsoft.com/en-us/library/ie/hh465388.aspx#attribute\_section

if ( el.querySelectorAll( "[msallowcapture^='']" ).length ) {

rbuggyQSA.push( "[\*^$]=" + whitespace + "\*(?:''|\"\")" );

}

// Support: IE8

// Boolean attributes and "value" are not treated correctly

if ( !el.querySelectorAll( "[selected]" ).length ) {

rbuggyQSA.push( "\\[" + whitespace + "\*(?:value|" + booleans + ")" );

}

// Support: Chrome<29, Android<4.4, Safari<7.0+, iOS<7.0+, PhantomJS<1.9.8+

if ( !el.querySelectorAll( "[id~=" + expando + "-]" ).length ) {

rbuggyQSA.push( "~=" );

}

// Support: IE 11+, Edge 15 - 18+

// IE 11/Edge don't find elements on a `[name='']` query in some cases.

// Adding a temporary attribute to the document before the selection works

// around the issue.

// Interestingly, IE 10 & older don't seem to have the issue.

input = document.createElement( "input" );

input.setAttribute( "name", "" );

el.appendChild( input );

if ( !el.querySelectorAll( "[name='']" ).length ) {

rbuggyQSA.push( "\\[" + whitespace + "\*name" + whitespace + "\*=" +

whitespace + "\*(?:''|\"\")" );

}

// Webkit/Opera - :checked should return selected option elements

// http://www.w3.org/TR/2011/REC-css3-selectors-20110929/#checked

// IE8 throws error here and will not see later tests

if ( !el.querySelectorAll( ":checked" ).length ) {

rbuggyQSA.push( ":checked" );

}

// Support: Safari 8+, iOS 8+

// https://bugs.webkit.org/show\_bug.cgi?id=136851

// In-page `selector#id sibling-combinator selector` fails

if ( !el.querySelectorAll( "a#" + expando + "+\*" ).length ) {

rbuggyQSA.push( ".#.+[+~]" );

}

// Support: Firefox <=3.6 - 5 only

// Old Firefox doesn't throw on a badly-escaped identifier.

el.querySelectorAll( "\\\f" );

rbuggyQSA.push( "[\\r\\n\\f]" );

} );

assert( function( el ) {

el.innerHTML = "<a href='' disabled='disabled'></a>" +

"<select disabled='disabled'><option/></select>";

// Support: Windows 8 Native Apps

// The type and name attributes are restricted during .innerHTML assignment

var input = document.createElement( "input" );

input.setAttribute( "type", "hidden" );

el.appendChild( input ).setAttribute( "name", "D" );

// Support: IE8

// Enforce case-sensitivity of name attribute

if ( el.querySelectorAll( "[name=d]" ).length ) {

rbuggyQSA.push( "name" + whitespace + "\*[\*^$|!~]?=" );

}

// FF 3.5 - :enabled/:disabled and hidden elements (hidden elements are still enabled)

// IE8 throws error here and will not see later tests

if ( el.querySelectorAll( ":enabled" ).length !== 2 ) {

rbuggyQSA.push( ":enabled", ":disabled" );

}

// Support: IE9-11+

// IE's :disabled selector does not pick up the children of disabled fieldsets

docElem.appendChild( el ).disabled = true;

if ( el.querySelectorAll( ":disabled" ).length !== 2 ) {

rbuggyQSA.push( ":enabled", ":disabled" );

}

// Support: Opera 10 - 11 only

// Opera 10-11 does not throw on post-comma invalid pseudos

el.querySelectorAll( "\*,:x" );

rbuggyQSA.push( ",.\*:" );

} );

}

if ( ( support.matchesSelector = rnative.test( ( matches = docElem.matches ||

docElem.webkitMatchesSelector ||

docElem.mozMatchesSelector ||

docElem.oMatchesSelector ||

docElem.msMatchesSelector ) ) ) ) {

assert( function( el ) {

// Check to see if it's possible to do matchesSelector

// on a disconnected node (IE 9)

support.disconnectedMatch = matches.call( el, "\*" );

// This should fail with an exception

// Gecko does not error, returns false instead

matches.call( el, "[s!='']:x" );

rbuggyMatches.push( "!=", pseudos );

} );

}

rbuggyQSA = rbuggyQSA.length && new RegExp( rbuggyQSA.join( "|" ) );

rbuggyMatches = rbuggyMatches.length && new RegExp( rbuggyMatches.join( "|" ) );

/\* Contains

---------------------------------------------------------------------- \*/

hasCompare = rnative.test( docElem.compareDocumentPosition );

// Element contains another

// Purposefully self-exclusive

// As in, an element does not contain itself

contains = hasCompare || rnative.test( docElem.contains ) ?

function( a, b ) {

var adown = a.nodeType === 9 ? a.documentElement : a,

bup = b && b.parentNode;

return a === bup || !!( bup && bup.nodeType === 1 && (

adown.contains ?

adown.contains( bup ) :

a.compareDocumentPosition && a.compareDocumentPosition( bup ) & 16

) );

} :

function( a, b ) {

if ( b ) {

while ( ( b = b.parentNode ) ) {

if ( b === a ) {

return true;

}

}

}

return false;

};

/\* Sorting

---------------------------------------------------------------------- \*/

// Document order sorting

sortOrder = hasCompare ?

function( a, b ) {

// Flag for duplicate removal

if ( a === b ) {

hasDuplicate = true;

return 0;

}

// Sort on method existence if only one input has compareDocumentPosition

var compare = !a.compareDocumentPosition - !b.compareDocumentPosition;

if ( compare ) {

return compare;

}

// Calculate position if both inputs belong to the same document

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

compare = ( a.ownerDocument || a ) == ( b.ownerDocument || b ) ?

a.compareDocumentPosition( b ) :

// Otherwise we know they are disconnected

1;

// Disconnected nodes

if ( compare & 1 ||

( !support.sortDetached && b.compareDocumentPosition( a ) === compare ) ) {

// Choose the first element that is related to our preferred document

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( a == document || a.ownerDocument == preferredDoc &&

contains( preferredDoc, a ) ) {

return -1;

}

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( b == document || b.ownerDocument == preferredDoc &&

contains( preferredDoc, b ) ) {

return 1;

}

// Maintain original order

return sortInput ?

( indexOf( sortInput, a ) - indexOf( sortInput, b ) ) :

0;

}

return compare & 4 ? -1 : 1;

} :

function( a, b ) {

// Exit early if the nodes are identical

if ( a === b ) {

hasDuplicate = true;

return 0;

}

var cur,

i = 0,

aup = a.parentNode,

bup = b.parentNode,

ap = [ a ],

bp = [ b ];

// Parentless nodes are either documents or disconnected

if ( !aup || !bup ) {

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

/\* eslint-disable eqeqeq \*/

return a == document ? -1 :

b == document ? 1 :

/\* eslint-enable eqeqeq \*/

aup ? -1 :

bup ? 1 :

sortInput ?

( indexOf( sortInput, a ) - indexOf( sortInput, b ) ) :

0;

// If the nodes are siblings, we can do a quick check

} else if ( aup === bup ) {

return siblingCheck( a, b );

}

// Otherwise we need full lists of their ancestors for comparison

cur = a;

while ( ( cur = cur.parentNode ) ) {

ap.unshift( cur );

}

cur = b;

while ( ( cur = cur.parentNode ) ) {

bp.unshift( cur );

}

// Walk down the tree looking for a discrepancy

while ( ap[ i ] === bp[ i ] ) {

i++;

}

return i ?

// Do a sibling check if the nodes have a common ancestor

siblingCheck( ap[ i ], bp[ i ] ) :

// Otherwise nodes in our document sort first

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

/\* eslint-disable eqeqeq \*/

ap[ i ] == preferredDoc ? -1 :

bp[ i ] == preferredDoc ? 1 :

/\* eslint-enable eqeqeq \*/

0;

};

return document;

};

Sizzle.matches = function( expr, elements ) {

return Sizzle( expr, null, null, elements );

};

Sizzle.matchesSelector = function( elem, expr ) {

setDocument( elem );

if ( support.matchesSelector && documentIsHTML &&

!nonnativeSelectorCache[ expr + " " ] &&

( !rbuggyMatches || !rbuggyMatches.test( expr ) ) &&

( !rbuggyQSA || !rbuggyQSA.test( expr ) ) ) {

try {

var ret = matches.call( elem, expr );

// IE 9's matchesSelector returns false on disconnected nodes

if ( ret || support.disconnectedMatch ||

// As well, disconnected nodes are said to be in a document

// fragment in IE 9

elem.document && elem.document.nodeType !== 11 ) {

return ret;

}

} catch ( e ) {

nonnativeSelectorCache( expr, true );

}

}

return Sizzle( expr, document, null, [ elem ] ).length > 0;

};

Sizzle.contains = function( context, elem ) {

// Set document vars if needed

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( ( context.ownerDocument || context ) != document ) {

setDocument( context );

}

return contains( context, elem );

};

Sizzle.attr = function( elem, name ) {

// Set document vars if needed

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( ( elem.ownerDocument || elem ) != document ) {

setDocument( elem );

}

var fn = Expr.attrHandle[ name.toLowerCase() ],

// Don't get fooled by Object.prototype properties (jQuery #13807)

val = fn && hasOwn.call( Expr.attrHandle, name.toLowerCase() ) ?

fn( elem, name, !documentIsHTML ) :

undefined;

return val !== undefined ?

val :

support.attributes || !documentIsHTML ?

elem.getAttribute( name ) :

( val = elem.getAttributeNode( name ) ) && val.specified ?

val.value :

null;

};

Sizzle.escape = function( sel ) {

return ( sel + "" ).replace( rcssescape, fcssescape );

};

Sizzle.error = function( msg ) {

throw new Error( "Syntax error, unrecognized expression: " + msg );

};

/\*\*

\* Document sorting and removing duplicates

\* @param {ArrayLike} results

\*/

Sizzle.uniqueSort = function( results ) {

var elem,

duplicates = [],

j = 0,

i = 0;

// Unless we \*know\* we can detect duplicates, assume their presence

hasDuplicate = !support.detectDuplicates;

sortInput = !support.sortStable && results.slice( 0 );

results.sort( sortOrder );

if ( hasDuplicate ) {

while ( ( elem = results[ i++ ] ) ) {

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

j = duplicates.push( i );

}

}

while ( j-- ) {

results.splice( duplicates[ j ], 1 );

}

}

// Clear input after sorting to release objects

// See https://github.com/jquery/sizzle/pull/225

sortInput = null;

return results;

};

/\*\*

\* Utility function for retrieving the text value of an array of DOM nodes

\* @param {Array|Element} elem

\*/

getText = Sizzle.getText = function( elem ) {

var node,

ret = "",

i = 0,

nodeType = elem.nodeType;

if ( !nodeType ) {

// If no nodeType, this is expected to be an array

while ( ( node = elem[ i++ ] ) ) {

// Do not traverse comment nodes

ret += getText( node );

}

} else if ( nodeType === 1 || nodeType === 9 || nodeType === 11 ) {

// Use textContent for elements

// innerText usage removed for consistency of new lines (jQuery #11153)

if ( typeof elem.textContent === "string" ) {

return elem.textContent;

} else {

// Traverse its children

for ( elem = elem.firstChild; elem; elem = elem.nextSibling ) {

ret += getText( elem );

}

}

} else if ( nodeType === 3 || nodeType === 4 ) {

return elem.nodeValue;

}

// Do not include comment or processing instruction nodes

return ret;

};

Expr = Sizzle.selectors = {

// Can be adjusted by the user

cacheLength: 50,

createPseudo: markFunction,

match: matchExpr,

attrHandle: {},

find: {},

relative: {

">": { dir: "parentNode", first: true },

" ": { dir: "parentNode" },

"+": { dir: "previousSibling", first: true },

"~": { dir: "previousSibling" }

},

preFilter: {

"ATTR": function( match ) {

match[ 1 ] = match[ 1 ].replace( runescape, funescape );

// Move the given value to match[3] whether quoted or unquoted

match[ 3 ] = ( match[ 3 ] || match[ 4 ] ||

match[ 5 ] || "" ).replace( runescape, funescape );

if ( match[ 2 ] === "~=" ) {

match[ 3 ] = " " + match[ 3 ] + " ";

}

return match.slice( 0, 4 );

},

"CHILD": function( match ) {

/\* matches from matchExpr["CHILD"]

1 type (only|nth|...)

2 what (child|of-type)

3 argument (even|odd|\d\*|\d\*n([+-]\d+)?|...)

4 xn-component of xn+y argument ([+-]?\d\*n|)

5 sign of xn-component

6 x of xn-component

7 sign of y-component

8 y of y-component

\*/

match[ 1 ] = match[ 1 ].toLowerCase();

if ( match[ 1 ].slice( 0, 3 ) === "nth" ) {

// nth-\* requires argument

if ( !match[ 3 ] ) {

Sizzle.error( match[ 0 ] );

}

// numeric x and y parameters for Expr.filter.CHILD

// remember that false/true cast respectively to 0/1

match[ 4 ] = +( match[ 4 ] ?

match[ 5 ] + ( match[ 6 ] || 1 ) :

2 \* ( match[ 3 ] === "even" || match[ 3 ] === "odd" ) );

match[ 5 ] = +( ( match[ 7 ] + match[ 8 ] ) || match[ 3 ] === "odd" );

// other types prohibit arguments

} else if ( match[ 3 ] ) {

Sizzle.error( match[ 0 ] );

}

return match;

},

"PSEUDO": function( match ) {

var excess,

unquoted = !match[ 6 ] && match[ 2 ];

if ( matchExpr[ "CHILD" ].test( match[ 0 ] ) ) {

return null;

}

// Accept quoted arguments as-is

if ( match[ 3 ] ) {

match[ 2 ] = match[ 4 ] || match[ 5 ] || "";

// Strip excess characters from unquoted arguments

} else if ( unquoted && rpseudo.test( unquoted ) &&

// Get excess from tokenize (recursively)

( excess = tokenize( unquoted, true ) ) &&

// advance to the next closing parenthesis

( excess = unquoted.indexOf( ")", unquoted.length - excess ) - unquoted.length ) ) {

// excess is a negative index

match[ 0 ] = match[ 0 ].slice( 0, excess );

match[ 2 ] = unquoted.slice( 0, excess );

}

// Return only captures needed by the pseudo filter method (type and argument)

return match.slice( 0, 3 );

}

},

filter: {

"TAG": function( nodeNameSelector ) {

var nodeName = nodeNameSelector.replace( runescape, funescape ).toLowerCase();

return nodeNameSelector === "\*" ?

function() {

return true;

} :

function( elem ) {

return elem.nodeName && elem.nodeName.toLowerCase() === nodeName;

};

},

"CLASS": function( className ) {

var pattern = classCache[ className + " " ];

return pattern ||

( pattern = new RegExp( "(^|" + whitespace +

")" + className + "(" + whitespace + "|$)" ) ) && classCache(

className, function( elem ) {

return pattern.test(

typeof elem.className === "string" && elem.className ||

typeof elem.getAttribute !== "undefined" &&

elem.getAttribute( "class" ) ||

""

);

} );

},

"ATTR": function( name, operator, check ) {

return function( elem ) {

var result = Sizzle.attr( elem, name );

if ( result == null ) {

return operator === "!=";

}

if ( !operator ) {

return true;

}

result += "";

/\* eslint-disable max-len \*/

return operator === "=" ? result === check :

operator === "!=" ? result !== check :

operator === "^=" ? check && result.indexOf( check ) === 0 :

operator === "\*=" ? check && result.indexOf( check ) > -1 :

operator === "$=" ? check && result.slice( -check.length ) === check :

operator === "~=" ? ( " " + result.replace( rwhitespace, " " ) + " " ).indexOf( check ) > -1 :

operator === "|=" ? result === check || result.slice( 0, check.length + 1 ) === check + "-" :

false;

/\* eslint-enable max-len \*/

};

},

"CHILD": function( type, what, \_argument, first, last ) {

var simple = type.slice( 0, 3 ) !== "nth",

forward = type.slice( -4 ) !== "last",

ofType = what === "of-type";

return first === 1 && last === 0 ?

// Shortcut for :nth-\*(n)

function( elem ) {

return !!elem.parentNode;

} :

function( elem, \_context, xml ) {

var cache, uniqueCache, outerCache, node, nodeIndex, start,

dir = simple !== forward ? "nextSibling" : "previousSibling",

parent = elem.parentNode,

name = ofType && elem.nodeName.toLowerCase(),

useCache = !xml && !ofType,

diff = false;

if ( parent ) {

// :(first|last|only)-(child|of-type)

if ( simple ) {

while ( dir ) {

node = elem;

while ( ( node = node[ dir ] ) ) {

if ( ofType ?

node.nodeName.toLowerCase() === name :

node.nodeType === 1 ) {

return false;

}

}

// Reverse direction for :only-\* (if we haven't yet done so)

start = dir = type === "only" && !start && "nextSibling";

}

return true;

}

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;

};

}

/\*\*

\* Returns a function to use in pseudos for :enabled/:disabled

\* @param {Boolean} disabled true for :disabled; false for :enabled

\*/

function createDisabledPseudo( disabled ) {

// Known :disabled false positives: fieldset[disabled] > legend:nth-of-type(n+2) :can-disable

return function( elem ) {

// Only certain elements can match :enabled or :disabled

// https://html.spec.whatwg.org/multipage/scripting.html#selector-enabled

// https://html.spec.whatwg.org/multipage/scripting.html#selector-disabled

if ( "form" in elem ) {

// Check for inherited disabledness on relevant non-disabled elements:

// \* listed form-associated elements in a disabled fieldset

// https://html.spec.whatwg.org/multipage/forms.html#category-listed

// https://html.spec.whatwg.org/multipage/forms.html#concept-fe-disabled

// \* option elements in a disabled optgroup

// https://html.spec.whatwg.org/multipage/forms.html#concept-option-disabled

// All such elements have a "form" property.

if ( elem.parentNode && elem.disabled === false ) {

// Option elements defer to a parent optgroup if present

if ( "label" in elem ) {

if ( "label" in elem.parentNode ) {

return elem.parentNode.disabled === disabled;

} else {

return elem.disabled === disabled;

}

}

// Support: IE 6 - 11

// Use the isDisabled shortcut property to check for disabled fieldset ancestors

return elem.isDisabled === disabled ||

// Where there is no isDisabled, check manually

/\* jshint -W018 \*/

elem.isDisabled !== !disabled &&

inDisabledFieldset( elem ) === disabled;

}

return elem.disabled === disabled;

// Try to winnow out elements that can't be disabled before trusting the disabled property.

// Some victims get caught in our net (label, legend, menu, track), but it shouldn't

// even exist on them, let alone have a boolean value.

} else if ( "label" in elem ) {

return elem.disabled === disabled;

}

// Remaining elements are neither :enabled nor :disabled

return false;

};

}

/\*\*

\* Returns a function to use in pseudos for positionals

\* @param {Function} fn

\*/

function createPositionalPseudo( fn ) {

return markFunction( function( argument ) {

argument = +argument;

return markFunction( function( seed, matches ) {

var j,

matchIndexes = fn( [], seed.length, argument ),

i = matchIndexes.length;

// Match elements found at the specified indexes

while ( i-- ) {

if ( seed[ ( j = matchIndexes[ i ] ) ] ) {

seed[ j ] = !( matches[ j ] = seed[ j ] );

}

}

} );

} );

}

/\*\*

\* Checks a node for validity as a Sizzle context

\* @param {Element|Object=} context

\* @returns {Element|Object|Boolean} The input node if acceptable, otherwise a falsy value

\*/

function testContext( context ) {

return context && typeof context.getElementsByTagName !== "undefined" && context;

}

// Expose support vars for convenience

support = Sizzle.support = {};

/\*\*

\* Detects XML nodes

\* @param {Element|Object} elem An element or a document

\* @returns {Boolean} True iff elem is a non-HTML XML node

\*/

isXML = Sizzle.isXML = function( elem ) {

var namespace = elem.namespaceURI,

docElem = ( elem.ownerDocument || elem ).documentElement;

// Support: IE <=8

// Assume HTML when documentElement doesn't yet exist, such as inside loading iframes

// https://bugs.jquery.com/ticket/4833

return !rhtml.test( namespace || docElem && docElem.nodeName || "HTML" );

};

/\*\*

\* Sets document-related variables once based on the current document

\* @param {Element|Object} [doc] An element or document object to use to set the document

\* @returns {Object} Returns the current document

\*/

setDocument = Sizzle.setDocument = function( node ) {

var hasCompare, subWindow,

doc = node ? node.ownerDocument || node : preferredDoc;

// Return early if doc is invalid or already selected

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( doc == document || doc.nodeType !== 9 || !doc.documentElement ) {

return document;

}

// Update global variables

document = doc;

docElem = document.documentElement;

documentIsHTML = !isXML( document );

// Support: IE 9 - 11+, Edge 12 - 18+

// Accessing iframe documents after unload throws "permission denied" errors (jQuery #13936)

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( preferredDoc != document &&

( subWindow = document.defaultView ) && subWindow.top !== subWindow ) {

// Support: IE 11, Edge

if ( subWindow.addEventListener ) {

subWindow.addEventListener( "unload", unloadHandler, false );

// Support: IE 9 - 10 only

} else if ( subWindow.attachEvent ) {

subWindow.attachEvent( "onunload", unloadHandler );

}

}

// Support: IE 8 - 11+, Edge 12 - 18+, Chrome <=16 - 25 only, Firefox <=3.6 - 31 only,

// Safari 4 - 5 only, Opera <=11.6 - 12.x only

// IE/Edge & older browsers don't support the :scope pseudo-class.

// Support: Safari 6.0 only

// Safari 6.0 supports :scope but it's an alias of :root there.

support.scope = assert( function( el ) {

docElem.appendChild( el ).appendChild( document.createElement( "div" ) );

return typeof el.querySelectorAll !== "undefined" &&

!el.querySelectorAll( ":scope fieldset div" ).length;

} );

/\* Attributes

---------------------------------------------------------------------- \*/

// Support: IE<8

// Verify that getAttribute really returns attributes and not properties

// (excepting IE8 booleans)

support.attributes = assert( function( el ) {

el.className = "i";

return !el.getAttribute( "className" );

} );

/\* getElement(s)By\*

---------------------------------------------------------------------- \*/

// Check if getElementsByTagName("\*") returns only elements

support.getElementsByTagName = assert( function( el ) {

el.appendChild( document.createComment( "" ) );

return !el.getElementsByTagName( "\*" ).length;

} );

// Support: IE<9

support.getElementsByClassName = rnative.test( document.getElementsByClassName );

// Support: IE<10

// Check if getElementById returns elements by name

// The broken getElementById methods don't pick up programmatically-set names,

// so use a roundabout getElementsByName test

support.getById = assert( function( el ) {

docElem.appendChild( el ).id = expando;

return !document.getElementsByName || !document.getElementsByName( expando ).length;

} );

// ID filter and find

if ( support.getById ) {

Expr.filter[ "ID" ] = function( id ) {

var attrId = id.replace( runescape, funescape );

return function( elem ) {

return elem.getAttribute( "id" ) === attrId;

};

};

Expr.find[ "ID" ] = function( id, context ) {

if ( typeof context.getElementById !== "undefined" && documentIsHTML ) {

var elem = context.getElementById( id );

return elem ? [ elem ] : [];

}

};

} else {

Expr.filter[ "ID" ] = function( id ) {

var attrId = id.replace( runescape, funescape );

return function( elem ) {

var node = typeof elem.getAttributeNode !== "undefined" &&

elem.getAttributeNode( "id" );

return node && node.value === attrId;

};

};

// Support: IE 6 - 7 only

// getElementById is not reliable as a find shortcut

Expr.find[ "ID" ] = function( id, context ) {

if ( typeof context.getElementById !== "undefined" && documentIsHTML ) {

var node, i, elems,

elem = context.getElementById( id );

if ( elem ) {

// Verify the id attribute

node = elem.getAttributeNode( "id" );

if ( node && node.value === id ) {

return [ elem ];

}

// Fall back on getElementsByName

elems = context.getElementsByName( id );

i = 0;

while ( ( elem = elems[ i++ ] ) ) {

node = elem.getAttributeNode( "id" );

if ( node && node.value === id ) {

return [ elem ];

}

}

}

return [];

}

};

}

// Tag

Expr.find[ "TAG" ] = support.getElementsByTagName ?

function( tag, context ) {

if ( typeof context.getElementsByTagName !== "undefined" ) {

return context.getElementsByTagName( tag );

// DocumentFragment nodes don't have gEBTN

} else if ( support.qsa ) {

return context.querySelectorAll( tag );

}

} :

function( tag, context ) {

var elem,

tmp = [],

i = 0,

// By happy coincidence, a (broken) gEBTN appears on DocumentFragment nodes too

results = context.getElementsByTagName( tag );

// Filter out possible comments

if ( tag === "\*" ) {

while ( ( elem = results[ i++ ] ) ) {

if ( elem.nodeType === 1 ) {

tmp.push( elem );

}

}

return tmp;

}

return results;

};

// Class

Expr.find[ "CLASS" ] = support.getElementsByClassName && function( className, context ) {

if ( typeof context.getElementsByClassName !== "undefined" && documentIsHTML ) {

return context.getElementsByClassName( className );

}

};

/\* QSA/matchesSelector

---------------------------------------------------------------------- \*/

// QSA and matchesSelector support

// matchesSelector(:active) reports false when true (IE9/Opera 11.5)

rbuggyMatches = [];

// qSa(:focus) reports false when true (Chrome 21)

// We allow this because of a bug in IE8/9 that throws an error

// whenever `document.activeElement` is accessed on an iframe

// So, we allow :focus to pass through QSA all the time to avoid the IE error

// See https://bugs.jquery.com/ticket/13378

rbuggyQSA = [];

if ( ( support.qsa = rnative.test( document.querySelectorAll ) ) ) {

// Build QSA regex

// Regex strategy adopted from Diego Perini

assert( function( el ) {

var input;

// Select is set to empty string on purpose

// This is to test IE's treatment of not explicitly

// setting a boolean content attribute,

// since its presence should be enough

// https://bugs.jquery.com/ticket/12359

docElem.appendChild( el ).innerHTML = "<a id='" + expando + "'></a>" +

"<select id='" + expando + "-\r\\' msallowcapture=''>" +

"<option selected=''></option></select>";

// Support: IE8, Opera 11-12.16

// Nothing should be selected when empty strings follow ^= or $= or \*=

// The test attribute must be unknown in Opera but "safe" for WinRT

// https://msdn.microsoft.com/en-us/library/ie/hh465388.aspx#attribute\_section

if ( el.querySelectorAll( "[msallowcapture^='']" ).length ) {

rbuggyQSA.push( "[\*^$]=" + whitespace + "\*(?:''|\"\")" );

}

// Support: IE8

// Boolean attributes and "value" are not treated correctly

if ( !el.querySelectorAll( "[selected]" ).length ) {

rbuggyQSA.push( "\\[" + whitespace + "\*(?:value|" + booleans + ")" );

}

// Support: Chrome<29, Android<4.4, Safari<7.0+, iOS<7.0+, PhantomJS<1.9.8+

if ( !el.querySelectorAll( "[id~=" + expando + "-]" ).length ) {

rbuggyQSA.push( "~=" );

}

// Support: IE 11+, Edge 15 - 18+

// IE 11/Edge don't find elements on a `[name='']` query in some cases.

// Adding a temporary attribute to the document before the selection works

// around the issue.

// Interestingly, IE 10 & older don't seem to have the issue.

input = document.createElement( "input" );

input.setAttribute( "name", "" );

el.appendChild( input );

if ( !el.querySelectorAll( "[name='']" ).length ) {

rbuggyQSA.push( "\\[" + whitespace + "\*name" + whitespace + "\*=" +

whitespace + "\*(?:''|\"\")" );

}

// Webkit/Opera - :checked should return selected option elements

// http://www.w3.org/TR/2011/REC-css3-selectors-20110929/#checked

// IE8 throws error here and will not see later tests

if ( !el.querySelectorAll( ":checked" ).length ) {

rbuggyQSA.push( ":checked" );

}

// Support: Safari 8+, iOS 8+

// https://bugs.webkit.org/show\_bug.cgi?id=136851

// In-page `selector#id sibling-combinator selector` fails

if ( !el.querySelectorAll( "a#" + expando + "+\*" ).length ) {

rbuggyQSA.push( ".#.+[+~]" );

}

// Support: Firefox <=3.6 - 5 only

// Old Firefox doesn't throw on a badly-escaped identifier.

el.querySelectorAll( "\\\f" );

rbuggyQSA.push( "[\\r\\n\\f]" );

} );

assert( function( el ) {

el.innerHTML = "<a href='' disabled='disabled'></a>" +

"<select disabled='disabled'><option/></select>";

// Support: Windows 8 Native Apps

// The type and name attributes are restricted during .innerHTML assignment

var input = document.createElement( "input" );

input.setAttribute( "type", "hidden" );

el.appendChild( input ).setAttribute( "name", "D" );

// Support: IE8

// Enforce case-sensitivity of name attribute

if ( el.querySelectorAll( "[name=d]" ).length ) {

rbuggyQSA.push( "name" + whitespace + "\*[\*^$|!~]?=" );

}

// FF 3.5 - :enabled/:disabled and hidden elements (hidden elements are still enabled)

// IE8 throws error here and will not see later tests

if ( el.querySelectorAll( ":enabled" ).length !== 2 ) {

rbuggyQSA.push( ":enabled", ":disabled" );

}

// Support: IE9-11+

// IE's :disabled selector does not pick up the children of disabled fieldsets

docElem.appendChild( el ).disabled = true;

if ( el.querySelectorAll( ":disabled" ).length !== 2 ) {

rbuggyQSA.push( ":enabled", ":disabled" );

}

// Support: Opera 10 - 11 only

// Opera 10-11 does not throw on post-comma invalid pseudos

el.querySelectorAll( "\*,:x" );

rbuggyQSA.push( ",.\*:" );

} );

}

if ( ( support.matchesSelector = rnative.test( ( matches = docElem.matches ||

docElem.webkitMatchesSelector ||

docElem.mozMatchesSelector ||

docElem.oMatchesSelector ||

docElem.msMatchesSelector ) ) ) ) {

assert( function( el ) {

// Check to see if it's possible to do matchesSelector

// on a disconnected node (IE 9)

support.disconnectedMatch = matches.call( el, "\*" );

// This should fail with an exception

// Gecko does not error, returns false instead

matches.call( el, "[s!='']:x" );

rbuggyMatches.push( "!=", pseudos );

} );

}

rbuggyQSA = rbuggyQSA.length && new RegExp( rbuggyQSA.join( "|" ) );

rbuggyMatches = rbuggyMatches.length && new RegExp( rbuggyMatches.join( "|" ) );

/\* Contains

---------------------------------------------------------------------- \*/

hasCompare = rnative.test( docElem.compareDocumentPosition );

// Element contains another

// Purposefully self-exclusive

// As in, an element does not contain itself

contains = hasCompare || rnative.test( docElem.contains ) ?

function( a, b ) {

var adown = a.nodeType === 9 ? a.documentElement : a,

bup = b && b.parentNode;

return a === bup || !!( bup && bup.nodeType === 1 && (

adown.contains ?

adown.contains( bup ) :

a.compareDocumentPosition && a.compareDocumentPosition( bup ) & 16

) );

} :

function( a, b ) {

if ( b ) {

while ( ( b = b.parentNode ) ) {

if ( b === a ) {

return true;

}

}

}

return false;

};

/\* Sorting

---------------------------------------------------------------------- \*/

// Document order sorting

sortOrder = hasCompare ?

function( a, b ) {

// Flag for duplicate removal

if ( a === b ) {

hasDuplicate = true;

return 0;

}

// Sort on method existence if only one input has compareDocumentPosition

var compare = !a.compareDocumentPosition - !b.compareDocumentPosition;

if ( compare ) {

return compare;

}

// Calculate position if both inputs belong to the same document

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

compare = ( a.ownerDocument || a ) == ( b.ownerDocument || b ) ?

a.compareDocumentPosition( b ) :

// Otherwise we know they are disconnected

1;

// Disconnected nodes

if ( compare & 1 ||

( !support.sortDetached && b.compareDocumentPosition( a ) === compare ) ) {

// Choose the first element that is related to our preferred document

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( a == document || a.ownerDocument == preferredDoc &&

contains( preferredDoc, a ) ) {

return -1;

}

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( b == document || b.ownerDocument == preferredDoc &&

contains( preferredDoc, b ) ) {

return 1;

}

// Maintain original order

return sortInput ?

( indexOf( sortInput, a ) - indexOf( sortInput, b ) ) :

0;

}

return compare & 4 ? -1 : 1;

} :

function( a, b ) {

// Exit early if the nodes are identical

if ( a === b ) {

hasDuplicate = true;

return 0;

}

var cur,

i = 0,

aup = a.parentNode,

bup = b.parentNode,

ap = [ a ],

bp = [ b ];

// Parentless nodes are either documents or disconnected

if ( !aup || !bup ) {

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

/\* eslint-disable eqeqeq \*/

return a == document ? -1 :

b == document ? 1 :

/\* eslint-enable eqeqeq \*/

aup ? -1 :

bup ? 1 :

sortInput ?

( indexOf( sortInput, a ) - indexOf( sortInput, b ) ) :

0;

// If the nodes are siblings, we can do a quick check

} else if ( aup === bup ) {

return siblingCheck( a, b );

}

// Otherwise we need full lists of their ancestors for comparison

cur = a;

while ( ( cur = cur.parentNode ) ) {

ap.unshift( cur );

}

cur = b;

while ( ( cur = cur.parentNode ) ) {

bp.unshift( cur );

}

// Walk down the tree looking for a discrepancy

while ( ap[ i ] === bp[ i ] ) {

i++;

}

return i ?

// Do a sibling check if the nodes have a common ancestor

siblingCheck( ap[ i ], bp[ i ] ) :

// Otherwise nodes in our document sort first

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

/\* eslint-disable eqeqeq \*/

ap[ i ] == preferredDoc ? -1 :

bp[ i ] == preferredDoc ? 1 :

/\* eslint-enable eqeqeq \*/

0;

};

return document;

};

Sizzle.matches = function( expr, elements ) {

return Sizzle( expr, null, null, elements );

};

Sizzle.matchesSelector = function( elem, expr ) {

setDocument( elem );

if ( support.matchesSelector && documentIsHTML &&

!nonnativeSelectorCache[ expr + " " ] &&

( !rbuggyMatches || !rbuggyMatches.test( expr ) ) &&

( !rbuggyQSA || !rbuggyQSA.test( expr ) ) ) {

try {

var ret = matches.call( elem, expr );

// IE 9's matchesSelector returns false on disconnected nodes

if ( ret || support.disconnectedMatch ||

// As well, disconnected nodes are said to be in a document

// fragment in IE 9

elem.document && elem.document.nodeType !== 11 ) {

return ret;

}

} catch ( e ) {

nonnativeSelectorCache( expr, true );

}

}

return Sizzle( expr, document, null, [ elem ] ).length > 0;

};

Sizzle.contains = function( context, elem ) {

// Set document vars if needed

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( ( context.ownerDocument || context ) != document ) {

setDocument( context );

}

return contains( context, elem );

};

Sizzle.attr = function( elem, name ) {

// Set document vars if needed

// Support: IE 11+, Edge 17 - 18+

// IE/Edge sometimes throw a "Permission denied" error when strict-comparing

// two documents; shallow comparisons work.

// eslint-disable-next-line eqeqeq

if ( ( elem.ownerDocument || elem ) != document ) {

setDocument( elem );

}

var fn = Expr.attrHandle[ name.toLowerCase() ],

// Don't get fooled by Object.prototype properties (jQuery #13807)

val = fn && hasOwn.call( Expr.attrHandle, name.toLowerCase() ) ?

fn( elem, name, !documentIsHTML ) :

undefined;

return val !== undefined ?

val :

support.attributes || !documentIsHTML ?

elem.getAttribute( name ) :

( val = elem.getAttributeNode( name ) ) && val.specified ?

val.value :

null;

};

Sizzle.escape = function( sel ) {

return ( sel + "" ).replace( rcssescape, fcssescape );

};

Sizzle.error = function( msg ) {

throw new Error( "Syntax error, unrecognized expression: " + msg );

};

/\*\*

\* Document sorting and removing duplicates

\* @param {ArrayLike} results

\*/

Sizzle.uniqueSort = function( results ) {

var elem,

duplicates = [],

j = 0,

i = 0;

// Unless we \*know\* we can detect duplicates, assume their presence

hasDuplicate = !support.detectDuplicates;

sortInput = !support.sortStable && results.slice( 0 );

results.sort( sortOrder );

if ( hasDuplicate ) {

while ( ( elem = results[ i++ ] ) ) {

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

j = duplicates.push( i );

}

}

while ( j-- ) {

results.splice( duplicates[ j ], 1 );

}

}

// Clear input after sorting to release objects

// See https://github.com/jquery/sizzle/pull/225

sortInput = null;

return results;

};

/\*\*

\* Utility function for retrieving the text value of an array of DOM nodes

\* @param {Array|Element} elem

\*/

getText = Sizzle.getText = function( elem ) {

var node,

ret = "",

i = 0,

nodeType = elem.nodeType;

if ( !nodeType ) {

// If no nodeType, this is expected to be an array

while ( ( node = elem[ i++ ] ) ) {

// Do not traverse comment nodes

ret += getText( node );

}

} else if ( nodeType === 1 || nodeType === 9 || nodeType === 11 ) {

// Use textContent for elements

// innerText usage removed for consistency of new lines (jQuery #11153)

if ( typeof elem.textContent === "string" ) {

return elem.textContent;

} else {

// Traverse its children

for ( elem = elem.firstChild; elem; elem = elem.nextSibling ) {

ret += getText( elem );

}

}

} else if ( nodeType === 3 || nodeType === 4 ) {

return elem.nodeValue;

}

// Do not include comment or processing instruction nodes

return ret;

};

Expr = Sizzle.selectors = {

// Can be adjusted by the user

cacheLength: 50,

createPseudo: markFunction,

match: matchExpr,

attrHandle: {},

find: {},

relative: {

">": { dir: "parentNode", first: true },

" ": { dir: "parentNode" },

"+": { dir: "previousSibling", first: true },

"~": { dir: "previousSibling" }

},

preFilter: {

"ATTR": function( match ) {

match[ 1 ] = match[ 1 ].replace( runescape, funescape );

// Move the given value to match[3] whether quoted or unquoted

match[ 3 ] = ( match[ 3 ] || match[ 4 ] ||

match[ 5 ] || "" ).replace( runescape, funescape );

if ( match[ 2 ] === "~=" ) {

match[ 3 ] = " " + match[ 3 ] + " ";

}

return match.slice( 0, 4 );

},

"CHILD": function( match ) {

/\* matches from matchExpr["CHILD"]

1 type (only|nth|...)

2 what (child|of-type)

3 argument (even|odd|\d\*|\d\*n([+-]\d+)?|...)

4 xn-component of xn+y argument ([+-]?\d\*n|)

5 sign of xn-component

6 x of xn-component

7 sign of y-component

8 y of y-component

\*/

match[ 1 ] = match[ 1 ].toLowerCase();

if ( match[ 1 ].slice( 0, 3 ) === "nth" ) {

// nth-\* requires argument

if ( !match[ 3 ] ) {

Sizzle.error( match[ 0 ] );

}

// numeric x and y parameters for Expr.filter.CHILD

// remember that false/true cast respectively to 0/1

match[ 4 ] = +( match[ 4 ] ?

match[ 5 ] + ( match[ 6 ] || 1 ) :

2 \* ( match[ 3 ] === "even" || match[ 3 ] === "odd" ) );

match[ 5 ] = +( ( match[ 7 ] + match[ 8 ] ) || match[ 3 ] === "odd" );

// other types prohibit arguments

} else if ( match[ 3 ] ) {

Sizzle.error( match[ 0 ] );

}

return match;

},

"PSEUDO": function( match ) {

var excess,

unquoted = !match[ 6 ] && match[ 2 ];

if ( matchExpr[ "CHILD" ].test( match[ 0 ] ) ) {

return null;

}

// Accept quoted arguments as-is

if ( match[ 3 ] ) {

match[ 2 ] = match[ 4 ] || match[ 5 ] || "";

// Strip excess characters from unquoted arguments

} else if ( unquoted && rpseudo.test( unquoted ) &&

// Get excess from tokenize (recursively)

( excess = tokenize( unquoted, true ) ) &&

// advance to the next closing parenthesis

( excess = unquoted.indexOf( ")", unquoted.length - excess ) - unquoted.length ) ) {

// excess is a negative index

match[ 0 ] = match[ 0 ].slice( 0, excess );

match[ 2 ] = unquoted.slice( 0, excess );

}

// Return only captures needed by the pseudo filter method (type and argument)

return match.slice( 0, 3 );

}

},

filter: {

"TAG": function( nodeNameSelector ) {

var nodeName = nodeNameSelector.replace( runescape, funescape ).toLowerCase();

return nodeNameSelector === "\*" ?

function() {

return true;

} :

function( elem ) {

return elem.nodeName && elem.nodeName.toLowerCase() === nodeName;

};

},

"CLASS": function( className ) {

var pattern = classCache[ className + " " ];

return pattern ||

( pattern = new RegExp( "(^|" + whitespace +

")" + className + "(" + whitespace + "|$)" ) ) && classCache(

className, function( elem ) {

return pattern.test(

typeof elem.className === "string" && elem.className ||

typeof elem.getAttribute !== "undefined" &&

elem.getAttribute( "class" ) ||

""

);

} );

},

"ATTR": function( name, operator, check ) {

return function( elem ) {

var result = Sizzle.attr( elem, name );

if ( result == null ) {

return operator === "!=";

}

if ( !operator ) {

return true;

}

result += "";

/\* eslint-disable max-len \*/

return operator === "=" ? result === check :

operator === "!=" ? result !== check :

operator === "^=" ? check && result.indexOf( check ) === 0 :

operator === "\*=" ? check && result.indexOf( check ) > -1 :

operator === "$=" ? check && result.slice( -check.length ) === check :

operator === "~=" ? ( " " + result.replace( rwhitespace, " " ) + " " ).indexOf( check ) > -1 :

operator === "|=" ? result === check || result.slice( 0, check.length + 1 ) === check + "-" :

false;

/\* eslint-enable max-len \*/

};

},

"CHILD": function( type, what, \_argument, first, last ) {

var simple = type.slice( 0, 3 ) !== "nth",

forward = type.slice( -4 ) !== "last",

ofType = what === "of-type";

return first === 1 && last === 0 ?

// Shortcut for :nth-\*(n)

function( elem ) {

return !!elem.parentNode;

} :

function( elem, \_context, xml ) {

var cache, uniqueCache, outerCache, node, nodeIndex, start,

dir = simple !== forward ? "nextSibling" : "previousSibling",

parent = elem.parentNode,

name = ofType && elem.nodeName.toLowerCase(),

useCache = !xml && !ofType,

diff = false;

if ( parent ) {

// :(first|last|only)-(child|of-type)

if ( simple ) {

while ( dir ) {

node = elem;

while ( ( node = node[ dir ] ) ) {

if ( ofType ?

node.nodeName.toLowerCase() === name :

node.nodeType === 1 ) {

return false;

}

}

// Reverse direction for :only-\* (if we haven't yet done so)

start = dir = type === "only" && !start && "nextSibling";

}

return true;

}

start = [ forward ? parent.firstChild : parent.lastChild ];

// non-xml :nth-child(...) stores cache data on `parent`

if ( forward && useCache ) {

// Seek `elem` from a previously-cached index

// ...in a gzip-friendly way

node = parent;

outerCache = node[ expando ] || ( node[ expando ] = {} );

// Support: IE <9 only

// Defend against cloned attroperties (jQuery gh-1709)

uniqueCache = outerCache[ node.uniqueID ] ||

( outerCache[ node.uniqueID ] = {} );

cache = uniqueCache[ type ] || [];

nodeIndex = cache[ 0 ] === dirruns && cache[ 1 ];

diff = nodeIndex && cache[ 2 ];

node = nodeIndex && parent.childNodes[ nodeIndex ];

while ( ( node = ++nodeIndex && node && node[ dir ] ||

// Fallback to seeking `elem` from the start

( diff = nodeIndex = 0 ) || start.pop() ) ) {

// When found, cache indexes on `parent` and break

if ( node.nodeType === 1 && ++diff && node === elem ) {

uniqueCache[ type ] = [ dirruns, nodeIndex, diff ];

break;

}

}

} else {

// Use previously-cached element index if available

if ( useCache ) {

// ...in a gzip-friendly way

node = elem;

outerCache = node[ expando ] || ( node[ expando ] = {} );

// Support: IE <9 only

// Defend against cloned attroperties (jQuery gh-1709)

uniqueCache = outerCache[ node.uniqueID ] ||

( outerCache[ node.uniqueID ] = {} );

cache = uniqueCache[ type ] || [];

nodeIndex = cache[ 0 ] === dirruns && cache[ 1 ];

diff = nodeIndex;

}

// xml :nth-child(...)

// or :nth-last-child(...) or :nth(-last)?-of-type(...)

if ( diff === false ) {

// Use the same loop as above to seek `elem` from the start

while ( ( node = ++nodeIndex && node && node[ dir ] ||

( diff = nodeIndex = 0 ) || start.pop() ) ) {

if ( ( ofType ?

node.nodeName.toLowerCase() === name :

node.nodeType === 1 ) &&

++diff ) {

// Cache the index of each encountered element

if ( useCache ) {

outerCache = node[ expando ] ||

( node[ expando ] = {} );

// Support: IE <9 only

// Defend against cloned attroperties (jQuery gh-1709)

uniqueCache = outerCache[ node.uniqueID ] ||

( outerCache[ node.uniqueID ] = {} );

uniqueCache[ type ] = [ dirruns, diff ];

}

if ( node === elem ) {

break;

}

}

}

}

}

// Incorporate the offset, then check against cycle size

diff -= last;

return diff === first || ( diff % first === 0 && diff / first >= 0 );

}

};

},

"PSEUDO": function( pseudo, argument ) {

// pseudo-class names are case-insensitive

// http://www.w3.org/TR/selectors/#pseudo-classes

// Prioritize by case sensitivity in case custom pseudos are added with uppercase letters

// Remember that setFilters inherits from pseudos

var args,

fn = Expr.pseudos[ pseudo ] || Expr.setFilters[ pseudo.toLowerCase() ] ||

Sizzle.error( "unsupported pseudo: " + pseudo );

// The user may use createPseudo to indicate that

// arguments are needed to create the filter function

// just as Sizzle does

if ( fn[ expando ] ) {

return fn( argument );

}

// But maintain support for old signatures

if ( fn.length > 1 ) {

args = [ pseudo, pseudo, "", argument ];

return Expr.setFilters.hasOwnProperty( pseudo.toLowerCase() ) ?

markFunction( function( seed, matches ) {

var idx,

matched = fn( seed, argument ),

i = matched.length;

while ( i-- ) {

idx = indexOf( seed, matched[ i ] );

seed[ idx ] = !( matches[ idx ] = matched[ i ] );

}

} ) :

function( elem ) {

return fn( elem, 0, args );

};

}

return fn;

}

},

pseudos: {

// Potentially complex pseudos

"not": markFunction( function( selector ) {

// Trim the selector passed to compile

// to avoid treating leading and trailing

// spaces as combinators

var input = [],

results = [],

matcher = compile( selector.replace( rtrim, "$1" ) );

return matcher[ expando ] ?

markFunction( function( seed, matches, \_context, xml ) {

var elem,

unmatched = matcher( seed, null, xml, [] ),

i = seed.length;

// Match elements unmatched by `matcher`

while ( i-- ) {

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

seed[ i ] = !( matches[ i ] = elem );

}

}

} ) :

function( elem, \_context, xml ) {

input[ 0 ] = elem;

matcher( input, null, xml, results );

// Don't keep the element (issue #299)

input[ 0 ] = null;

return !results.pop();

};

} ),

"has": markFunction( function( selector ) {

return function( elem ) {

return Sizzle( selector, elem ).length > 0;

};

} ),

"contains": markFunction( function( text ) {

text = text.replace( runescape, funescape );

return function( elem ) {

return ( elem.textContent || getText( elem ) ).indexOf( text ) > -1;

};

} ),

// "Whether an element is represented by a :lang() selector

// is based solely on the element's language value

// being equal to the identifier C,

// or beginning with the identifier C immediately followed by "-".

// The matching of C against the element's language value is performed case-insensitively.

// The identifier C does not have to be a valid language name."

// http://www.w3.org/TR/selectors/#lang-pseudo

"lang": markFunction( function( lang ) {

// lang value must be a valid identifier

if ( !ridentifier.test( lang || "" ) ) {

Sizzle.error( "unsupported lang: " + lang );

}

lang = lang.replace( runescape, funescape ).toLowerCase();

return function( elem ) {

var elemLang;

do {

if ( ( elemLang = documentIsHTML ?

elem.lang :

elem.getAttribute( "xml:lang" ) || elem.getAttribute( "lang" ) ) ) {

elemLang = elemLang.toLowerCase();

return elemLang === lang || elemLang.indexOf( lang + "-" ) === 0;

}

} while ( ( elem = elem.parentNode ) && elem.nodeType === 1 );

return false;

};

} ),

// Miscellaneous

"target": function( elem ) {

var hash = window.location && window.location.hash;

return hash && hash.slice( 1 ) === elem.id;

},

"root": function( elem ) {

return elem === docElem;

},

"focus": function( elem ) {

return elem === document.activeElement &&

( !document.hasFocus || document.hasFocus() ) &&

!!( elem.type || elem.href || ~elem.tabIndex );

},

// Boolean properties

"enabled": createDisabledPseudo( false ),

"disabled": createDisabledPseudo( true ),

"checked": function( elem ) {

// In CSS3, :checked should return both checked and selected elements

// http://www.w3.org/TR/2011/REC-css3-selectors-20110929/#checked

var nodeName = elem.nodeName.toLowerCase();

return ( nodeName === "input" && !!elem.checked ) ||

( nodeName === "option" && !!elem.selected );

},

"selected": function( elem ) {

// Accessing this property makes selected-by-default

// options in Safari work properly

if ( elem.parentNode ) {

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

elem.parentNode.selectedIndex;

}

return elem.selected === true;

},

// Contents

"empty": function( elem ) {

// http://www.w3.org/TR/selectors/#empty-pseudo

// :empty is negated by element (1) or content nodes (text: 3; cdata: 4; entity ref: 5),

// but not by others (comment: 8; processing instruction: 7; etc.)

// nodeType < 6 works because attributes (2) do not appear as children

for ( elem = elem.firstChild; elem; elem = elem.nextSibling ) {

if ( elem.nodeType < 6 ) {

return false;

}

}

return true;

},

"parent": function( elem ) {

return !Expr.pseudos[ "empty" ]( elem );

},

// Element/input types

"header": function( elem ) {

return rheader.test( elem.nodeName );

},

"input": function( elem ) {

return rinputs.test( elem.nodeName );

},

"button": function( elem ) {

var name = elem.nodeName.toLowerCase();

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

},

"text": function( elem ) {

var attr;

return elem.nodeName.toLowerCase() === "input" &&

elem.type === "text" &&

// Support: IE<8

// New HTML5 attribute values (e.g., "search") appear with elem.type === "text"

( ( attr = elem.getAttribute( "type" ) ) == null ||

attr.toLowerCase() === "text" );

},

// Position-in-collection

"first": createPositionalPseudo( function() {

return [ 0 ];

} ),

"last": createPositionalPseudo( function( \_matchIndexes, length ) {

return [ length - 1 ];

} ),

"eq": createPositionalPseudo( function( \_matchIndexes, length, argument ) {

return [ argument < 0 ? argument + length : argument ];

} ),

"even": createPositionalPseudo( function( matchIndexes, length ) {

var i = 0;

for ( ; i < length; i += 2 ) {

matchIndexes.push( i );

}

return matchIndexes;

} ),

"odd": createPositionalPseudo( function( matchIndexes, length ) {

var i = 1;

for ( ; i < length; i += 2 ) {

matchIndexes.push( i );

}

return matchIndexes;

} ),

"lt": createPositionalPseudo( function( matchIndexes, length, argument ) {

var i = argument < 0 ?

argument + length :

argument > length ?

length :

argument;

for ( ; --i >= 0; ) {

matchIndexes.push( i );

}

return matchIndexes;

} ),

"gt": createPositionalPseudo( function( matchIndexes, length, argument ) {

var i = argument < 0 ? argument + length : argument;

for ( ; ++i < length; ) {

matchIndexes.push( i );

}

return matchIndexes;

} )

}

};