



# JavaScript Navigator

**Summary:** in this tutorial, you will learn about the JavaScript Navigator object and its properties.

## Introduction to the JavaScript Navigator object

The JavaScript Navigator provides information about the web browser and its capabilities. You can reference the Navigator object via the read-only `window.navigator` property.

The `Navigator` object has properties that convey the browser's information. For example, the `userAgent` is a property of the `window.navigator` object. It is a long string that identifies the web browser.

```
window.navigator.userAgent
```

In Google Chrome, you may see the following output:

```
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.4046.202 Safari/537.36"
```

Note that the `userAgent` may be a little bit different, depending on the Google Chrome version.

The different web browser provides specific capabilities which are not standardized. It's better not to use the `userAgent` to identify the web browser because some web browsers allow users to modify the `userAgent` to pretend they are using a different browser.

For example, you may use the following code to detect if the current web browser is Internet Explorer:

```
if(navigator.userAgent.includes('MSIE')) {  
    // IE, use specific features of IE  
} else {
```

```
// not IE
}
```

To use a specific feature of a web browser, you can use the capability detection. For example:

```
if( typeof window.addEventListener === 'function' ) {
    // Let's use addEventListener
} else {
    // addEventListener is not supported, use another way
}
```

## JavaScript Navigator Properties & Methods

The following table illustrates the JavaScript **Navigator** properties and methods:

Property / Method	Description
activeVrDisplays	Returns an array of every VRDisplay instance with its presenting property is set to true
appCodeName	Returns "Mozilla" even in non-Mozilla browsers.
appName	Returns the full browser name.
appVersion	Returns the browser version. However, it typically does not correspond to the actual version of the browser.
battery	Returns a BatteryManager object to interact with the Battery status API
buildId	Return the build number for the web browser.
connection	Returns a NetworkInformation object to interact with the Network information API
cookieEnabled	Returns true if cookies are enabled; otherwise false.

Property / Method	Description
credentials	Returns a <code>CredentialsContainer</code> to interact with the Credentials Management API
deviceMemory	Returns the amount of device memory in gigabytes.
doNotTrack	Returns the user's preference of do-not-track .
geolocation	Returns a <code>Geolocation</code> object to interact with the Geolocation API.
getVRDisplays()	Returns an array of every <code>VRDisplay</code> instance if available.
getUserMedia()	Returns the stream associated with the available media device hardware.
hardwareConcurrency	Returns the number of processor cores of the device
javaEnabled	Determines if Java is enabled in the browser.
language	Returns the browser's primary language.
languages	Returns an array of all the browser's preferred languages.
locks	Returns a <code>LockManager</code> object to interact with the Web Locks API.
mediaCapabilities	Returns a <code>MediaCapabilities</code> object to interact with the Media capabilities API
mediaDevices	Returns the available media devices.
maxTouchPoints	Returns the maximum number of supported touch points for the device's touchscreen
mimeTypes	Returns an array of MIME types registered with the browser.

Property / Method	Description
onLine	Specifies if the browser is connected to the Internet.
oscpu	The operating system (OS) and/or CPU on which the browser is running.
permissions	Returns the Permissions object to interact with the Permissions API.
platform	Returns the system platform on which the browser is running.
plugins	Returns an array of installed browser's plug-ins.
product	Returns the name of the product.
productSub	Returns the extra information about the product.
registerProtocolHandler()	Registers a website as a handler for a particular protocol.
requestMediaKeySystemAccess()	Returns a <a href="#">Promise</a> which resolves to a MediaKeySystemAccess object.
sendBeacon()	Asynchronously transmits a small payload.
serviceWorker	Returns the ServiceWorkerContainer used to interact with ServiceWorker object
share()	Calls the current platform's native sharing mechanism.
storage	Returns the StorageManager object to interact with the Storage API.
userAgent	Represents the user-agent string of the browser.
vendor	Returns the brand name of the browser.
vendorSub	Returns extra information about the browser's vendor.

Property / Method	Description
vibrate()	Triggers the device to vibrate if vibration is supported.
webdriver	Determines if the browser is currently controlled by automation.