

این جزوه برای آموزش JAVASCRIPT در مدرسه پرنیان آماده شده است. برای کشف تمام جزئیات آن میتوانید در دوره <http://trainingsitedesign.ir> ها ثبت نام کرده و همراه ما باشید ❤️

Navigator

The **Navigator** interface represents the state and the identity of the user agent. It allows scripts to query it and to register themselves to carry on some activities.

1. `navigator.connection.effectiveType` => 3g,4g,....
2. `navigator.cookieEnabled` => true , false
3. `navigator.deviceMemory` => ~ GB
4. `navigator.hardwareConcurrency` => processor cores
5. `navigator.language` => en
6. `navigator.languages`
7. `navigator.maxTouchPoints` => maximum number of simultaneous touch
8. `navigator.onLine` => online , offline
9. `navigator.pdfViewerEnabled` => true, false, browser can display PDF files inline
10. `navigator.userAgent` => user agent current browser
11. `navigator.userAgentData.brands` => browser brands
12. `navigator.userAgentData.mobile` => is mobile?

13. navigator.userAgentData.platform => your OS?

14. navigator.webdriver => is controlled by automation?

15. navigator.appCodeName => Mozilla

16. navigator.appName => Netscape

17. navigator.appVersion => AppleWebKit/537.36....

18. navigator.platform => ~ platform of the browser

19. navigator.product => Always returns 'Gecko', in any browser

20. navigator.vibrate(200) => vibrate for 200ms

21. navigator.vibrate([100, 30, 100, 30, 100, 30, 200, 30]) => Vibrate 'SOS' in Morse

22. navigator.mimeTypes => return array application/pdf..

23. navigator.share()

24. navigator.clipboard.writeText() => copy to clipboard

Example 1:

```
<p><button>Share MDN!</button></p>
<p class="result"></p>
<script>
  const shareData = {
    title: "parnian",
    text: "Learn web web design!",
    url: "https://trainingsitedesign.ir",
  };

  const btn = document.querySelector("button");
  const resultPara = document.querySelector(".result");

  // Share must be triggered by "user activation"
  btn.addEventListener("click", async () => {
    try {
      await navigator.share(shareData);
      resultPara.textContent = "parnian shared successfully";
    } catch (err) {
      resultPara.textContent = `Error: ${err}`;
    }
  });
</script>
```

Example2:

```
25. <div>
26.   <label for="files">Select images to share:</label>
27.   <input id="files" type="file" accept="image/*" multiple />
28. </div>
29. <button id="share" type="button">Share your images!</button>
30. <output id="output"></output>
31. <script>
32.   const input = document.getElementById("files");
33.   const output = document.getElementById("output");
34.
35.   document.getElementById("share").addEventListener("click", async () => {
36.     const files = input.files;
37.
38.     if (files.length === 0) {
39.       output.textContent = "No files selected.";
40.       return;
41.     }
42.
43.     // feature detecting navigator.canShare() also implies
44.     // the same for the navigator.share()
45.     if (!navigator.canShare) {
46.       output.textContent = `Your browser doesn't support the Web Share
API.`;
47.       return;
48.     }
49.
50.     if (navigator.canShare({ files })) {
51.       try {
52.         await navigator.share({
53.           files,
54.           title: "Images",
55.           text: "Beautiful images",
56.         });
57.         output.textContent = "Shared!";
58.       } catch (error) {
59.         output.textContent = `Error: ${error.message}`;
60.       }
61.     } else {
62.       output.textContent = `Your system doesn't support sharing these
files.`;
63.     }
64.   });
65. </script>
```

navigator.getBattery()

```
let batteryIsCharging = false;

navigator.getBattery().then((battery) => {
  batteryIsCharging = battery.charging;

  battery.addEventListener("chargingchange", () => {
    batteryIsCharging = battery.charging;
    console.log(batteryIsCharging);
  });
});
```

Navigator.geolocation

```
<script>
  const options = {
    enableHighAccuracy: true,
    timeout: 5000,
    maximumAge: 0,
  };

  function success(pos) {
    const crd = pos.coords;

    console.log("Your current position is:");
    console.log(`Latitude : ${crd.latitude}`);
    console.log(`Longitude: ${crd.longitude}`);
    console.log(`More or less ${crd.accuracy} meters.`);
  }

  function error(err) {
    console.warn(`ERROR(${err.code}): ${err.message}`);
  }

  navigator.geolocation.getCurrentPosition(success, error, options);
</script>
```