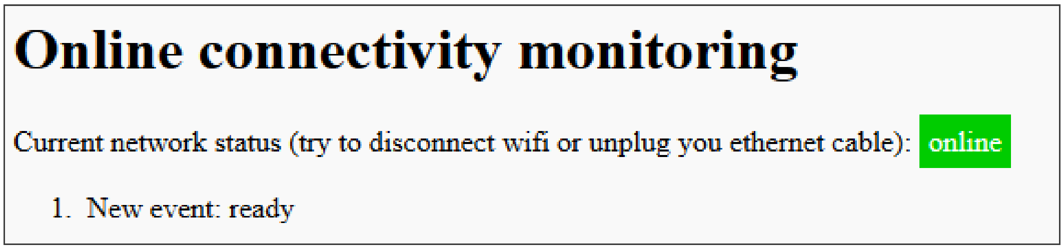
Checking that the browser is online/offline

The new HTML5 persistence APIs are very often used with the `navigator.onLine` property, part of the DOM API. This feature is available on all browsers. The `navigator.onLine`property returns `true` or `false` depending on whether or not the application has network connectivity.

Beware that a browser may be "online" but if your applications talks to a remote server, and if this one does not answer, or if your DNS server is down, being online does not mean that your application fully works. Gmail, for example, detects when the remote service is down and displays a message "trying to connect in 30s...".

> **Important: if the browser is offline, this means that your application should work in "degraded, offline mode". If it's online, it *should* work, but there is no guarantee that your remote server is up and running, that the DNS server is operational, etc.**

Check connectivity: online example on JS Bin



Online connectivity monitoring

Current network status (try to disconnect wifi or unplug you ethernet cable): online

1. New event: ready

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset=utf-8>
 <title>HTML5 Demo: Online connectivity monitoring</title>
</head>
<body>
  <header>
   <h1>Online connectivity monitoring</h1>
```

```
10.       <style>
          #status {
            color: #FFFFFF;
            padding: 5px;
          }
          .online {
             background: green;
          }
          .offline {
             background: red;
20.       }
          </style>
        </header>
        <article>
          <p>Current network status (try to disconnect wifi or unplug
            your ethernet cable):
            <span id="status">checking...</span>
          </p>
          <ol id="state"></ol>
        </article>
        <script>
32.      var statusElem =document.getElementById('status'),
         var state = document.getElementById('state');
         function online(event) {
          statusElem.className = navigator.onLine ?'online' : 'offline';
          statusElem.innerHTML = navigator.onLine ?'online' : 'offline';
          state.innerHTML += '<li>New event: ' + event.type+ '</li>';
         }
         window.addEventListener('online', online);
42.      window.addEventListener('offline', online);
         // call the online function so that it refreshes display when
         // the page is first loaded
         online({ type: 'ready' });
      </script>
      </html>
```

Usually, one checks if the application is running in online or offline mode (in this last case, data may be retrieved from the client side, using one of the various methods presented in this week's course). Here is what you should do instead of displaying messages (lines 41-42 in the code source shown above). Replace by such an implementation:

```
window.addEventListener('online', function(e) {
 // Re-sync data with server.
}, false);
window.addEventListener('offline', function(e) {
 // Queue up events for server, store them on the browser side
}, false);
```

KNOWLEDGE CHECK 6.3.8 (NOT GRADED)

My browser is online, and I verified this status by using the navigator.onLine property. Does this mean that my application will be able to talk to a remote server?

○ Yes

○ No

○ Normally it should work, but the onLine property only says that the browser is connected to the network. It does not check that the DNS is working nor that the remote server is up and running.

CHECK