JavaScript Page Load Events

Summary: in this tutorial, you will learn about the events that are related to the page load including `DOMContentLoaded`, `load`, `beforeunload`, and `unload`.

Overview of JavaScript page load events

When you open a page, the following events occur in sequence:

- `DOMContentLoaded` – the browser fully loaded HTML and completed building the DOM tree. However, it hasn't loaded external resources like stylesheets and images. In this event, you can start selecting DOM nodes or initialize the interface.
- `load` – the browser fully loaded the HTML and external resources like images and stylesheets.

When you leave the page, the following events fire in sequence:

- `beforeunload` – fires before the page and resources are unloaded. You can use this event to show a confirmation dialog to confirm if you want to leave the page. By doing this, you can prevent data loss in case the user is filling out a form and accidentally clicks a link that navigates to another page.
- `unload` – fires when the page has completely unloaded. You can use this event to send the analytic data or to clean up resources.

Handling JavaScript page load events

To handle the page events, you can call the `addEventListener()` method on the `document` object:

```
document.addEventListener('DOMContentLoaded',() => {
    // handle DOMContentLoaded event
});
```

```
document.addEventListener('load',() => {
    // handle load event
});

document.addEventListener('beforeunload',() => {
    // handle beforeunload event
});

document.addEventListener('unload',() => {
    // handle unload event
});
```

The following example illustrates how to handle the page load events:

```
<!DOCTYPE html>
<html>
<head>
    <title>JS Page Load Events</title>
</head>

<body>
    <script>
        addEventListener('DOMContentLoaded', (event) => {
            console.log('The DOM is fully loaded.');
        });

        addEventListener('load', (event) => {
            console.log('The page is fully loaded.');
        });

        addEventListener('beforeunload', (event) => {
            // show the confirmation dialog
            event.preventDefault();
            // Google Chrome requires returnValue to be set.
            event.returnValue = '';
        });

        addEventListener('unload', (event) => {
            // send analytic data
```

```
        });
    </script>
</body>
</html>
```