JS JavaScript
TUTORIAL

JavaScript Window

Summary: in this tutorial, you will learn about the JavaScript `window` object which is the global object of JavaScript in the browser and exposes the browser's functionality.

The window object is global

The global object of JavaScript in the web browser is the `window` object. It means that all variables and functions declared globally with the `var` keyword become the properties and methods of the `window` object. For example:

```
var counter = 1;
var showCounter = () => console.log({ counter });

console.log(window.counter);
window.showCounter();
```

Output:

```
1
{counter: 1}
```

Because the `counter` variable and the `showCounter()` function are declared globally with the `var` keyword, they are automatically added to the `window` object.

If you don't want to pollute the `window` object, you can use the `let` keyword to declare variables and functions.

The window object exposes the browser's functionality

The `window` object exposes the functionality of the web browser to the webpage.

1) Window size

The `window` object has four properties related to the size of the window:

- The `innerWidth` and `innerHeight` properties return the size of the page viewport inside the browser window (not including the borders and toolbars).
- The `outerWidth` and `outerHeight` properties return the size of the browser window itself.

Also, `document.documentElement.clientWidth` and `document.documentElement.clientHeight` properties indicate the width and height of the page viewport.

To get the size of the window, you use the following snippet:

```
const width = window.innerWidth
|| document.documentElement.clientWidth
|| document.body.clientWidth;

const height = window.innerHeight
|| document.documentElement.clientHeight
|| document.body.clientHeight;
```

2) Open a new window

To open a new window or tab, you use the `window.open()` method:

```
window.open(url, windowName, [windowFeatures]);
```

The `window.open()` method accepts three arguments:

- The URL to load
- The window target
- A string that represents the window's features.

The third argument (`windowFeatures`) is a comma-delimited string of settings, specifying displaying information for the new window such as width, height, menubar, and resizable.

The `window.open()` method returns a `WindowProxy` object, which is a thin wrapper of the `window` object. In case the new window cannot be opened, it returns `null` .

For example, to open a new window that loads the page `about.html` at localhost, you use the following code:

```
let url = 'http://localhost/js/about.html';
let jsWindow = window.open(url,'about');
```

> Note that the web browser will block the popup automatically. To see the new window, you need to unblock the popup from your web browser.

The code opens the page `about.html` in a new tab. If you specify the `height` and `width` of the window, it will open the URL in a new separated window instead of a new tab:

```
const features = 'height=600,width=800';
const url = 'about.html';

const jsWindow = window.open(url, 'about', features);
```

To load another URL on an existing window, you pass an existing window name to the `window.open()` method.

For example, the following loads the `contact.html` webpage to the `contact` window:

```
window.open('http://localhost/js/contact.html','about');
```

Put it all together.

The following code opens a new window that loads the webpage `about.html` and then after 3 seconds, it loads the webpage `contact.html` in the same window:

```
let features = 'height=600,width=800',
    url = 'http://localhost/js/about.html';
```

```
let jsWindow = window.open(url, 'about', features);

setTimeout(() => {
    window.open('http://localhost/js/contact.html', 'about')
}, 3000);
```

3) Resize a window

To resize a window you use the `resizeTo()` method of the `window` object:

```
window.resizeTo(width,height);
```

The following example opens a new window that loads the `http://localhost/js/about.html` page and resize it to `(600,300)` after 3 seconds:

```
let jsWindow = window.open(
    'http://localhost/js/about.html',
    'about',
    'height=600,width=800');

setTimeout(() => {
    jsWindow.resizeTo(600, 300);
}, 3000);
```

The `window.resizeBy()` method allows you to resize the current window by a specified amount:

```
window.resizeBy(deltaX,deltaY);
```

For example:

```
let jsWindow = window.open(
    'http://localhost/js/about.html',
    'about',
    'height=600,width=600');
```

```
// shrink the window, or resize the window
// to 500x500
setTimeout(() => {
    jsWindow.resizeBy(-100, -100);
}, 3000);
```

4) Move a window

To move a window to a specified coordinate, you use the `moveTo()` method:

```
window.moveTo(x, y);
```

In this method, `x` and `y` are horizontal and vertical coordinates to be moved to. The following example opens a new window and moves it to `(0,0)` coordinate after 3 seconds:

```
let jsWindow = window.open(
    'http://localhost/js/about.html',
    'about',
    'height=600,width=600');

setTimeout(() => {
    jsWindow.moveTo(500, 500);
}, 3000);
```

Similarly, you can move the current window by a specified amount:

```
let jsWindow = window.open(
    'http://localhost/js/about.html',
    'about',
    'height=600,width=600');

setTimeout(() => {
    jsWindow.moveBy(100, -100);
}, 3000);
```

5) Close a window

To close a window, you use the `window.close()` method:

```
window.close()
```

The following example opens a new window and closes it after 3 seconds:

```
let jsWindow = window.open(
    'http://localhost/js/about.html',
    'about',
    'height=600,width=600');

setTimeout(() => {
    jsWindow.close();
}, 3000);
```

6) The window.opener property

A newly created window can reference back to the window that opened it via the `window.opener` property. This allows you to exchange data between the two windows.

Summary

- The `window` is the global object in the web browser.
- The `window` object exposes the functionality of the web browser.
- The window object provides methods for manipulating a window such as `open()` , `resize()` , `resizeBy()` , `moveTo()` , `moveBy()` , and `close()` .