

PlayBox JS

PlayBox is a JavaScript library created by Deepak Kumar, this library helps to displays HTML, image slider by filling the screen and dimming out the web page.

This PlayBox Library does not use any other external Library, it also contains some of the polyfill jQuery functions.

Visit : [PlayBox JS Demo](#)

Getting Started

Download the Latest version of PlayBox from the GitHub release page

Download from: [PlayBox GitHub Release](#)

How to use:

1. Include the playbox.js into your document, this PlayBox library injects necessary stylesheet into the document header when the page loads. you can override those values in custom CSS by !important property.

```
<script src='playbox.js'></script>
```

2. Create a div tag in the body section mentioned below, and you can use any number of boxes as with unique different *containerName* and *buttonName*, below are the examples of playbox use cases

Note: The *cssSelector* mentioned in below examples also support *PlayBox.getElement()* and *Javascript document selected Object*

HTML PlayBox - PlayBox.open()

```
<div class="playbox containerName">
  <div class="container">
    ...
    ...
  </div>
</div>

<button class="buttonName">Click Me</button>

<script type="text/javascript">
  PlayBox.open('.buttonName', '.containerName', 200, 300);
</script>
```

Note: 200, 300 are fadein and fadeout duration in milliseconds for box and it's optional, the default duration is 300 milliseconds

Image Slider PlayBox - PlayBox.imageOpen()

```
<div class="playboximage containerName">
    ...
    ... <!-- If you like your other elements here -->
    ...
</div>

<button class="buttonName">Click Me</button>

<script type="text/javascript">

    PlayBox.imageOpen('.buttonName', '.containerName',
        ['image1.jpg', 'image2.jpg', ...], 200, 300);

</script>
```

Note: 200, 300 are fadein and fadeout duration in milliseconds for box and it's optional, the default duration is 300 milliseconds

Fade In animation - PlayBox.fadeIn()

```
<div class="cssSelector">Sample Element</div>

<script type="text/javascript">

    PlayBox.fadeIn('.cssSelector', 300, function(){});

</script>
```

Note 1: 300 are fadein duration in milliseconds for the element and it's optional, the default delay is 100 milliseconds

Note 2: The third argument is a callback function after the fadein animation is complete the callback function executed, and it's an optional argument

Fade out animation - PlayBox.fadeOut()

```
<div class="cssSelector">Sample Element</div>

<script type="text/javascript">

    PlayBox.fadeOut('.cssSelector', 300, function(){});

</script>
```

Note 1: 300 are fadeout duration in milliseconds for the element and it's optional, the default delay is 100 milliseconds, after the animation is completed the element display property set to display:none

Note 2: The third argument is a callback function after the fadeout animation is complete the callback function executed, and it's an optional argument

Nesting PlayBox Functions

We can also nest the callback function on each other

```
PlayBox.fadeOut('.cssSelector', 1000, function(){
    PlayBox.fadeIn('.cssSelector', 1000, function(){
        alert("Animation Completed")
    });
});
```

DOM Selector PlayBox.getElement()

This Selector will return an HTMLElement of the selected DOM element,

```
PlayBox.getElement('cssSelector');
```

Note 1: This function will return the first occurrence of the element in the DOM

DOM Selector PlayBox.getElements()

This Selector will return an array of Elements, of the selected DOM element, You can perform all Array operations on those elements,

```
PlayBox.getElements('cssSelector');
```

DOM Selector PlayBox.children()

This Selector will return all the first level child element as NodeList,

```
PlayBox.children('cssSelector');
```

Note 1: If you like to do Array operations on those NodeList, pass the NodeList into PlayBox.getElements(NodeList) this will return a Array of Children

DOM Operations PlayBox.append()

This method inserts specified HTML content to the end of the selected elements,

```
PlayBox.append('cssSelector', '<tag>Element</tag>');
```

DOM Operations PlayBox.prepend()

This method inserts specified HTML content to the beginning of the selected elements,

```
PlayBox.prepend('cssSelector', '<tag>Element</tag>');
```

DOM Operations **PlayBox.hasClass()**

This method return true if the have the specifed class else it will return false,

```
PlayBox.hasClass('cssSelector', 'className');
```

DOM Operations **PlayBox.is()**

This method return true if the selected element is a specifed tag name element, else it will return false.

```
PlayBox.is('cssSelector', 'HTMLTagName');
```

DOM Operations **PlayBox.index()**

This method return index integer of the element relative to it's parent Element.

```
PlayBox.index('cssSelector');
```

DOM Event Handling **PlayBox.click()**

This method add the click event handler to the selected element and execute the callback function if the element was clicked.

```
PlayBox.click('cssSelector', function(){  
    alert('Element Clicked');  
});
```