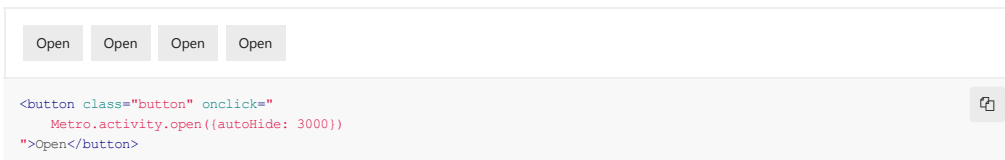


set and get value



```
<button class="button" onclick="
  Metro.activity.open({
    type: 'metro',
    overlayClickClose: true
  })
">Open</button>

<button class="button" onclick="
  var activity = Metro.activity.open({
    type: 'square',
    overlayColor: '#fff',
    overlayAlpha: 1
  });
  setTimeout(function() {
    Metro.activity.close(activity);
  }, 5000)
">Open</button>

<button class="button" onclick="
  Metro.activity.open({
    type: 'square',
    overlayColor: '#fff',
    overlayAlpha: 1,
    text: '<div class=\'mt-2 text-small\'>Please, wait...</div>',
    overlayClickClose: true
  });
">Open</button>
```

Progress indicator

Metro 4 implements four types of progress indicator. To create progress indicator add `data-role="progress"` attribute to element and define additional options. The progress indicator type you can define with `data-type=*` attribute. To set `value` and `buffer` add attributes `data-value` or/and `data-buffer`.

Progress bar

```
<div data-role="progress" data-value="35"></div>

<div data-role="progress" data-value="35" data-small="true"></div>
```

Progress bar with buffer

```
<div data-role="progress" data-type="buffer"
  data-value="35" data-buffer="60"></div>

<div data-role="progress" data-type="buffer"
  data-value="35" data-buffer="60" data-small="true"></div>
```

Progress bar with buffer and load indicator

```
<div data-role="progress" data-type="load"
  data-value="35" data-buffer="75"></div>

<div data-role="progress" data-type="load"
  data-value="35" data-buffer="75" data-small="true"></div>
```

Progress line

```
<div data-role="progress" data-type="line"></div>

<div data-role="progress" data-type="line" data-small="true"></div>
```

Customize

To set your own custom color use `data-cls-back`, `data-cls-bar` and `data-cls-buffer` attributes.

```
<div data-role="progress"
  data-type="buffer"
  data-cls-back="bg-yellow"
  data-cls-bar="bg-blue"
  data-cls-buffer="bg-pink"
  data-value="25" data-buffer="65"></div>
```

Events

When the `value` or `buffer` changes in the `progress`, a `change` or/and `buffer` events is fired. You can use this events to observe progress value and buffer.

```
$("#progress").on("valuechange", function (val) {
  console.log(val);
});

$("#progress").on("bufferchange", function (val) {
  console.log(val);
});
```

```
});
```

Callbacks

Progress indicator implements several `callbacks` to respond to a change in the status of the indicator:

Function	Data-*	Desc
onValueChange(val, el)	data-on-value-change	Fired when value changes
onBufferChange(val, el)	data-on-buffer-change	Fired when buffer changes
onComplete(val, el)	data-on-complete	Fired when value is 100%
onBuffered(val, el)	data-on-buffered	Fired when buffer is 100%
onProgressCreate(el)	data-on-progress-create	Fired when element was created

```
<div data-role="progress" data-type="load"
    data-value="35" data-buffer="75"
    data-on-complete="alert('Complete!!!')"
    data-on-value-change="console.log(arguments)"
    data-on-buffer-change="console.log(arguments)">
</div>
```

Observe

If you change `data-value` or `data-buffer` attributes at runtime, your `progress` will be updated.

Value

Buffer

```
<div id="progress-observe"
    data-role="progress"
    data-type="buffer" class="mb-4"></div>

<div>
    <input id="progress-observe-value"
        class="w-100" type="range"
        min="0" max="100" value="0">

    <input id="progress-observe-buffer"
        class="w-100" type="range"
        min="0" max="100" value="0">
</div>

<script>
    $(function () {
        $("#progress-observe-value").on("input change", function () {
            $("#progress-observe").attr('data-value', this.value);
        });
        $("#progress-observe-buffer").on("input change", function () {
            $("#progress-observe").attr('data-buffer', this.value);
        });
    })
</script>
```

Set and get value

Component `progress` provides method to get and set value for progress. To `set` or `get` value use method `val()`. Also you can `set` and `get` values for `buffer` with method `buff()`.

Value

0

Get value

Buffer

0

Get value

```
var progress = $("#element").data("progress");
var progress_value;
var buffer_value;

// set value
progress.val(35);
progress.buff(65);

// get value
progress_value = progress.val();
buffer_value = progress.buff();
```