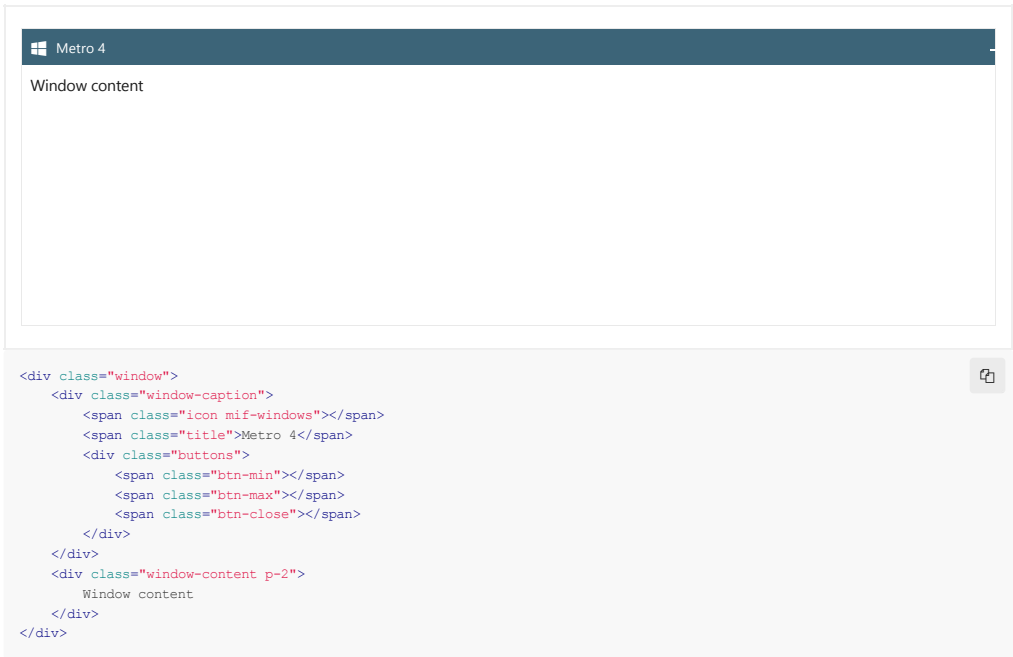


Windows

Metro 4 provides special classes to create window elements and manipulate them.

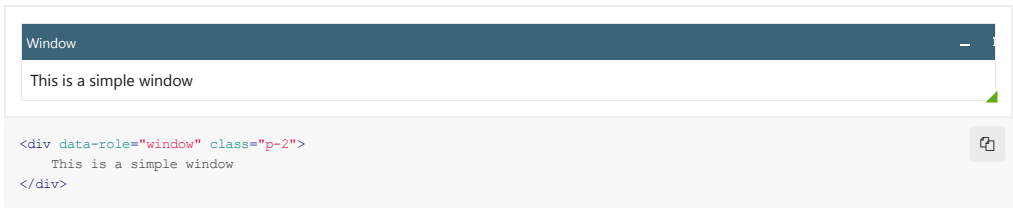
About

Metro 4 build windows with using next classes: `.window` , `.window-caption` , `.window-content` and result window has next structure:



Create window

To create window add attribute `data-role="window"` to your block element.

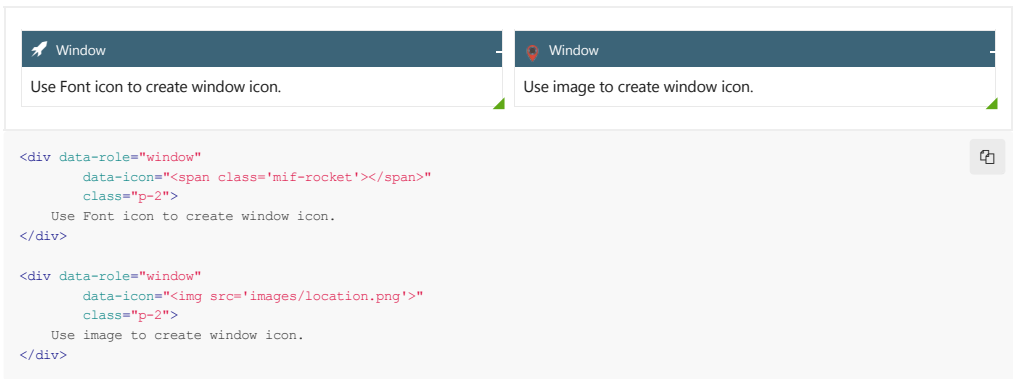


Caption

The caption may include the following elements: `icon` , `title` , `system buttons` .

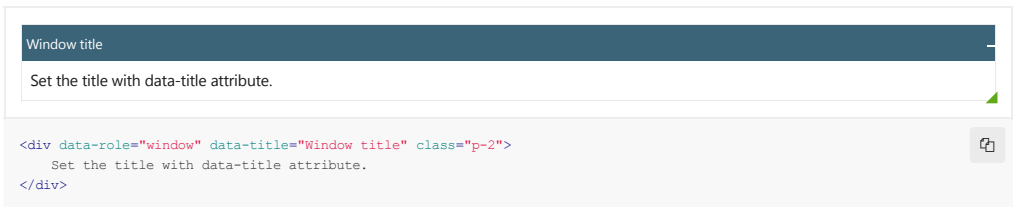
Icon

To add `icon` to window you must add attribute `data-icon="..."` to element.



Title

To add `title` to window you must add attribute `data-title="..."` to element.



Buttons

- Windows
- About
- Create window
- Window caption
 - Icon
 - Title
- Buttons
- Buttons events
- Custom buttons
- Shadow
- Customize
- Draggable
- Resizable
- Options
- Events
- Methods
- Metro.window object

You can add system buttons to your window such as: `close`, `min` and `max`. To create buttons you must add relevant attributes:
`data-btn-close="true"`, `data-btn-min="true"`, `data-btn-max="true"`.

Window title

On/off window system buttons with attributes.

```
<div data-role="window" id="-window-buttons"
  data-title="Window title"
  data-btn-close="true"
  data-btn-min="true"
  data-btn-max="true"
  class="p-2">
  On/off window system buttons with attributes.
</div>
```

☒ btn-close ☒ btn-min ☒ btn-max

```
<input checked="" type="checkbox"
  data-role="checkbox"
  data-caption="btn-close"
  onclick="$('#-window-buttons').attr('data-btn-close', $(this).is(':checked'))">

<input checked="" type="checkbox"
  data-role="checkbox"
  data-caption="btn-min"
  onclick="$('#-window-buttons').attr('data-btn-min', $(this).is(':checked'))">

<input checked="" type="checkbox"
  data-role="checkbox"
  data-caption="btn-max"
  onclick="$('#-window-buttons').attr('data-btn-max', $(this).is(':checked'))">
```

Buttons events

When Metro 4 created window buttons, also created default event for this buttons. You can extend buttons events with custom events over special attributes: `data-on-close-click="..."`, `data-on-min-click="..."`, `data-on-max-click="..."`, `data-on-caption-dbl-click="..."`.

The default events are shown in the example below:

Window

Click to buttons to view default buttons events.

Custom buttons

You can add a custom buttons to window caption. To add custom buttons, use attribute `data-custom-buttons`

Window

Click to buttons to view default buttons events.

```
<script>
  var customButtons = [
    {
      html: "<span class='mif-rocket'></span>",
      cls: "sys-button",
      onclick: "alert('You press rocket button!')"
    },
    {
      html: "<span class='mif-user'></span>",
      cls: "alert",
      onclick: "alert('You press user button!')"
    },
    {
      html: "<span class='mif-cog'></span>",
      cls: "warning",
      onclick: "alert('You press cog button!')"
    }
  ];
</script>
<div class="p-2"
  data-role="window"
  data-draggable="true"
  data-title="Window"
  data-btn-close="true"
  data-btn-min="true"
  data-btn-max="true"
  data-custom-buttons="customButtons"
  data-width="300" data-height="160">
  Click to buttons to view default buttons events.
```

</div>

Add shadow to window

You can add `shadow` to window with attribute `data-shadow="true"`.

Window title


Add shadow to window with attribute data-shadow

```
<div data-role="window" data-title="Window title" data-shadow="true" class="p-2">
  Add shadow to window with attribute data-shadow
</div>
```

☒ Enable shadow

Customize

You can change `default colors` for `caption` and `content`. To set it use attributes `data-cls-caption="..."` and `data-cls-content="..."`.

 The rocket

You can customize window with own classes.

```
<div class="p-2" data-role="window" data-icon="<span class='mif-rocket'></span>" data-title="The rocket" data-cls-caption="bg-red" data-cls-content="bg-dark fg-yellow">
  This is a simple DIV transformed to window with data-role.
</div>
```

Draggable window

To add `draggable` feature to window add attribute `data-draggable="true"` to your element. About `draggable` you can read [in this article](#).

Window

Drag this window on the caption.

```
<div class="p-2" data-role="window" data-draggable="true" data-width="200" data-height="160">
  Drag this window on the caption.
</div>
```

☒ Enable draggable

Resizable window

To make window as `resizable` add attribute `data-resizable="true"` to your window.

Window

Resize the window on hold and move bottom right corner of the window.

```
<div class="p-2" data-role="window" data-resizable="true" data-draggable="true" data-width="300" data-height="160">
  Resize the window on hold and move bottom right corner of the window.
</div>
```

Available options

All options you can set over `data-*` attribute:

Option	Data attribute	Default	Desc	Observe
width	<code>data-width</code>	auto	Set initial width to window	✓
height	<code>data-height</code>	auto	Set initial height to window	✓
btnClose	<code>data-btn-close</code>	true	Add close button	✓
btnMin	<code>data-btn-min</code>	true	Add min button	✓
btnMax	<code>data-btn-max</code>	true	Add max button	✓
clsWindow	<code>data-cls-window</code>		Additional classes for window	✓
clsCaption	<code>data-cls-caption</code>		Additional classes for caption	✓
clsContent	<code>data-cls-content</code>		Additional classes for content	✓
draggable	<code>data-draggable</code>	true	Add draggable to window	✓
dragElement	<code>data-drag-element</code>	.window-caption	Set the drag element	⊗
dragArea	<code>data-drag-area</code>	parent	Set the area where element can be draggable	⊗
shadow	<code>data-shadow</code>	false	Add shadow to window	✓
icon	<code>data-icon</code>		Window icon	✓
title	<code>data-title</code>	Window	Window title	✓
content	<code>data-content</code>		Window content	✓
resizable	<code>data-resizable</code>	true	Add resizable to window	✓
top	<code>data-top</code>	auto	Set window top position	✓
left	<code>data-left</code>	auto	Set window left position	✓
place	<code>data-place</code>	auto	Set window place (can be: center, top-left, top-center, top-right, right-center, bottom-right, bottom-center, bottom-left, left-center)	✓
closeAction	<code>data-close-action</code>	Metro.actions.REMOVE	Action when window close. Metro.actions.REMOVE or Metro.actions.HIDE	⊗

Events

Event	Data-*	Desc
<code>onDragStart</code>	<code>data-on-drag-start</code>	Generated when the user starts dragging the window
<code>onDragStop</code>	<code>data-on-drag-stop</code>	Generated when the user stops dragging the window
<code>onDragMove</code>	<code>data-on-drag-move</code>	Generated when the user dragging the window
<code>onCaptionDbClick</code>	<code>data-on-caption-dbl-click</code>	Generated when the user double clicks on the window caption
<code>onCloseClick</code>	<code>data-on-close-click</code>	Generated when the user click close button
<code>onMaxClick</code>	<code>data-on-max-click</code>	Generated when the user click max button
<code>onMinClick</code>	<code>data-on-min-click</code>	Generated when the user click min button
<code>onResizeStart</code>	<code>data-on-resize-start</code>	Generated when the user starts resizing the window
<code>onResizeStop</code>	<code>data-on-resize-stop</code>	Generated when the user stops resizing the window
<code>onResize</code>	<code>data-on-resize</code>	Generated when the user resize the window
<code>onWindowCreate</code>	<code>data-on-window-create</code>	Generated when window was created
<code>onDestroy</code>	<code>data-on-window-destroy</code>	Generated when the window destroyed

How to use event you can see in [this example](#). The example source you can find on [github](#).

Methods

The window has a several methods to interact with it.

Method	Desc
<code>minimized()</code>	Toggle minimized state
<code>maximized()</code>	Toggle maximized state
<code>close()</code>	Close window
<code>hide()</code>	Add class <code>.no-visible</code> to window
<code>show()</code>	Remove class <code>.no-visible</code> from window

Metro.window object

Added in 4.2.1

You can use object `Metro.window` to manipulate of windows components. This object contains next methods:

- `isWindow(el)` - checks, if element has role `window`
- `show(el)` - show hidden window
- `hide(el)` - hide window
- `toggle(el)` - toggle show/hide window
- `isOpen(el)` - checks, if a window is showed
- `close(el)` - close window
- `create(...)` - create window
- `min(el, bool)` - minimized or unminimized window
- `max(el, bool)` - maximized or unmaximized window