

Search...

Getting started
Components

- M4Q
- About
- Population
- Constructor
- Ajax
- Animation
- Loops
- Visibility
- Effects
- Subtree functions
- Attributes
- Html, text and value
- Css and classes
- Position and size
- Manipulation
- DataSet
- Events
- Utils
- Base
- Containers
- Grid system
- Typography
- Tables
- Forms
- Buttons
- Images
- Figures
- Lists
- Form components
- Checkbox
- File input
- Input
- Input material
- Keypad
- Rating
- Radio
- Select
- Slider
- Double Slider
- Spinner
- Switch
- Tag input
- Textarea
- Menus
- App bar
- Bottom navigation
- Bottom sheet
- Menu
- Ribbon Menu
- Side bar
- Side navigation
- Controls
- Accordion
- Badge
- Carousel
- Cards
- Cube
- Counter
- Charms
- Chat
- Donut
- Image compare
- Image magnifier
- Gravatar
- List
- ListView
- Master
- NavView
- Panels
- Progress & Activity
- Streamer
- Stepper
- Splitter
- Tabs
- Tabs material
- Table
- Tiles
- TreeView
- Wizard
- Information
- Dialogs
- Info box
- Hints
- Notify system
- Popovers
- Toasts
- Windows
- Date & time
- Calendar
- Calendar picker
- Date picker
- Time picker
- Countdown
- Formatting date
- Media

Hotkeys

Metro 4 lets you easily add handlers for keyboard events anywhere in your site.

This feature works for desktop browsers.

Binding

To bind `hotkey` add attribute `data-hotkey` with hotkey value to your control.

Press Ctrl+1 Ctrl+f1

Press Alt+2

Press Shift+3

```
<ul class="v-menu">
  <li>
    <a href="#"
      onclick="alert('Ctrl+1 is clicked!')
      data-hotkey="Ctrl+1">Press Ctrl+1</a>
    </li>
</ul>

<button class="button"
  data-hotkey="Alt+2"
  onclick="alert('Alt+2 clicked!')">Press Alt+2</button>

<a href="#"
  data-hotkey="Shift+3"
  onclick="alert('Shift+3 clicked!')">Press Shift+3</a>
```

Note

Modifiers are not case sensitive (Ctrl == ctrl == cTRL). If you want to use more than one modifier (e.g. alt+ctrl+z) you should define them by an alphabetical order e.g. `alt+ctrl+shift`. Hotkeys aren't tracked if you're inside of an input element (unless you explicitly bind the hotkey directly to the input). This helps to avoid conflict with normal user typing.

Press alt+ctrl+shift+z

```
<a href="#"
  data-hotkey="alt+ctrl+shift+z"
  onclick="alert('alt+ctrl+shift+z clicked!')">Press alt+ctrl+shift+z</a>
```

Hotkeys aren't tracked if the user is focused within an input element or any element that has `contenteditable="true"` unless you bind the hotkey directly to the element. This helps to avoid conflict with normal user typing. If you don't want this, you can change the booleans of the following to suit before `metro.js` loaded:

- METRO_HOTKEYS_FILTER_CONTENT_EDITABLE
- METRO_HOTKEYS_FILTER_INPUT_ACCEPTING_ELEMENTS
- METRO_HOTKEYS_FILTER_TEXT_INPUTS

```
<script>
  METRO_HOTKEYS_FILTER_CONTENT_EDITABLE = false;
</script>
<script src="metro/js/metro.js?ver=@@b-version"></script>
```

Events

In Metro 4 hotkeys triggered when `keyup` fired. To bubble up your hotkey event set `METRO_HOTKEYS_BUBBLE_UP` to `true` before `metro.js` loaded.

Addendum

Firefox is the most liberal one in the manner of letting you capture all short-cuts even those that are built-in in the browser such as Ctrl-t for new tab, or Ctrl-a for selecting all text. You can always bubble them up to the browser by returning true in your handler.

Others, (IE) either let you handle built-in short-cuts, but will add their functionality after your code has executed. Or (Opera/Safari) will not pass those events to the DOM at all.

So, if you bind Ctrl-Q or Alt-F4 and your Safari/Opera window is closed don't be surprised.

Custom binding

For custom hotkeys you can use method `$(el).hotkey(key, fn)`. To create own event:

Type `$` to replace it to `EUR`

```
<input type="text" id="hotkey-input">

$("#hotkey-input").hotkey("shift+4", function() {
  return this.value = this.value.replace('$', 'EUR');
})
```

- Video player
- Audio player
- Tools
- Collapse
- Color module
- Draggable
- Dropdown
- Form validator
- Hotkeys**
- Micro templates
- Ripple
- Storage
- Session storage
- Sorter
- Touch and swipe

- Utilities**
- Animations**
- Additional**