

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

Hints

Metro 4 has an advanced, easy customizable hints system.

Create hint

The hints system is intended for informing the user about the appointment of a particular element. Example, with `hint` you can inform user about button appointment. To create hinted element add `data-role="hint"` attribute to element and set hint text with attribute `data-hint-text="..."`.

Hover me

```
<button class="button" data-role="hint" data-hint-text="This is a hinted button">
  Hover me
</button>
```

Hint position

You can set four positions for hint. To set hint position add attribute `data-hint-position="..."` to element. By default position is `top`.

Right

Top

Bottom

Left

```
<button class="button" data-role="hint" data-hint-position="right" data-hint-text="This is a hinted button">Right</button>

<button class="button" data-role="hint" data-hint-position="top" data-hint-text="This is a hinted button">Top</button>

<button class="button" data-role="hint" data-hint-position="bottom" data-hint-text="This is a hinted button">Bottom</button>

<button class="button" data-role="hint" data-hint-position="left" data-hint-text="This is a hinted button">Left</button>
```

Options

You can set hint options to specify behavior.

Option	Data-*	Default	Desc
hintHide	data-hint-hide	5000	Milliseconds to auto hiding hint
clsHint	data-cls-hint		Additional class for hint
hintText	data-hint-text		Text for hint
hintPosition	data-hint-position	top	Hint position
hintOffset	data-hint-offset	4	Hint position offset from element
onHintCreate	data-on-hint-create	Metro.noop	Fired when hint created
onHintShow	data-on-hint-show	Metro.noop	Fired when hint showing
onHintHide	data-on-hint-hide	Metro.noop	Fired when hint hiding

Hover me

```
<button class="button alert" data-role="hint" data-hint-text="This is a hinted button" data-cls-hint="bg-cyan fg-white drop-shadow">
  >Hover me</button>
```

Events

When the `hint` is running, it generates various `events` that you can use. How to define Metro 4 components events see [Events rules](#).

- onHintCreate(hint, element)** - fired when hint created
- onHintShow(hint, element)** - fired when hint showing
- onHintHide(hint, element)** - fired when hint hiding

Hover me

```
<style>
  .showHint {
    animation-name: hintIn;
    animation-duration: 0.5s;
```

Table of contents

- Hints
- Create hint
- Hint position
- Options
- Events

- 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

```
}
.hideHint {
  animation-name: hintOut;
  animation-duration: 0.5s;
}

@keyframes hintIn {
  0% {
    transform: translate3d(0, -200px, 0) scale3d(0.1, 0.1, 0.1);
    opacity: 0; }
  40% {
    opacity: 1;
    animation-timing-function: cubic-bezier(0.47, 0, 0.745, 0.715);
    transform: translate3d(0, 0, 0) scale3d(1.08, 1.08, 1.08); }
  60% {
    animation-timing-function: cubic-bezier(0.42, 0, 0.58, 1);
    transform: translate3d(0, 0, 0) scale3d(1, 1, 1); }
  80% {
    animation-timing-function: cubic-bezier(0.42, 0, 0.58, 1);
    transform: translate3d(0, 0, 0) scale3d(1.03, 1.03, 1.03); }
  100% {
    animation-timing-function: cubic-bezier(0.25, 0.46, 0.45, 0.94);
    transform: translate3d(0, 0, 0) scale3d(1, 1, 1);
  }
}

@keyframes hintOut {
  0% {
    animation-timing-function: cubic-bezier(0.25, 0.46, 0.45, 0.94);
    transform: scale3d(1, 1, 1); }
  60% {
    animation-timing-function: cubic-bezier(0.42, 0, 0.58, 1);
    transform: scale3d(1.08, 1.08, 1.08); }
  80% {
    opacity: 1;
    animation-timing-function: cubic-bezier(0.42, 0, 0.58, 1); }
  100% {
    opacity: 0;
    animation-timing-function: cubic-bezier(0.47, 0, 0.745, 0.715);
    transform: scale3d(0.3, 0.3, 0.3);
  }
}

</style>

<button class="button info"
  data-role="hint"
  data-hint-text="This is a hinted button"
  data-cls-hint="drop-shadow"
  data-on-hint-show="hintRoutines.showHint"
  data-on-hint-hide="hintRoutines.hideHint"
  data-hint-hide="0">Hover me</button>

<script>
  var hintRoutines = {
    showHint: function(hint, element){
      hint.addClass("showHint");
      setTimeout(function(){
        hint.removeClass("showHint");
      }, 500)
    },

    hideHint: function(hint, element){
      hint.addClass("hideHint");
    }
  }
</script>
```