

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

Input control

Enhancement for standard HTML input control.

About

In Metro 4, you can use enhancement version of input control. To define it, add attribute `data-role="input"` to HTML input element. You can assign role `input` to any text inputs: `text` , `password` , `email` , etc.

```
<input type="text" data-role="input">
```

Prepend data

You can add prepend data to input field with attribute `data-prepend="..."` .

User name:

```
<input type="text" data-role="input" data-prepend="User name: ">
```

Append data

You can add append data to input field with attribute `data-append="..."` .

.00%

```
<input type="text" data-role="input" data-append=".00%">
```

Clear button

When Metro 4 create input component, him add to element two special buttons: `clear-button` and `reveal-button` . These buttons shown when input focused or user move mouse over input.

This button clear current value and flash it to default if default value is defined. You can set default value with attribute `data-default-value="..."` . You can disable `clear-button` with attribute `data-clear-button="false"` . You can change default icon for this button. To set your icon, use attribute `data-clear-button-icon="..."` .

Input

Input with default value 100

Input without clear button

Custom clear button icon

```
<p>Input</p>
<input type="text" data-role="input" class="">

<p>Input with default value 100</p>
<input type="text" data-role="input" data-default-value="100">

<p>Input without clear button</p>
<input type="text" data-role="input" data-clear-button="false">

<p>Custom clear button icon</p>
<input type="text" data-role="input" data-clear-button-icon="<span class='mif-cancel'></span>">
```

Reveal button

This button works with `input [type=password]` field. When user press this button, him can see input value. You can disable `reveal-button` with attribute `data-reveal-button="false"` . You can change default icon for this button. To set your icon, use attribute `data-reveal-button-icon="..."` .

Input password

Input without reveal button

Custom reveal button icon

```
<p>Input password</p>
<input type="password" data-role="input">
```

Table of contents

- Input control
- About
- Prepend data
- Append data
- Clear button
- Reveal button
- Search button
- History inputs
- Custom buttons
- Autocomplete
- Options
- Events
- Methods
- Customize
- Additional classes

- 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

```
<p>Input without reveal button</p>
<input type="password" data-role="input" data-reveal-button="false">

<p>Custom reveal button icon</p>
<input type="password" data-role="input" data-reveal-button-icon="<span class='mif-lamp mif-2x'></span>">
```

Search button

You can enable `search-button` with attribute `data-search-button="true"` . You can change default icon for this button. To set your icon, use attribute `data-search-button-icon="..."` .

When user press this button:

If You define attribute `data-search-button-click="custom"` , Metro 4 execute your function, defined with attribute `data-on-search-button-click="..."` .
If you define attribute `data-search-button-click="submit"` , Metro 4 submit a form, where search input is defined

Default search button icon

Custom search button icon

```
<p>Input without search button</p>
<input type="text" data-role="input" data-search-button="true">

<p>Custom search button icon</p>
<input type="text" data-role="input" data-search-button="true"
      data-search-button-icon="<span class='mif-rocket mif-2x'></span>">
```

Define search button click event

Custom func

Form submit func

```
<script>
  function mySubmitSearch(val){
    alert('Custom submit function');
  }
</script>
<form action="javascript:" onsubmit="alert('Form submit with self function');">
  <p>Custom func</p>
  <input type="text" data-role="input"
        data-search-button-click="custom"
        data-on-search-button-click="mySubmitSearch">

  <p>Form submit func</p>
  <input type="text" data-role="input">
</form>
```

History

Metro 4 input support `history` feature. To enable it feature, add attribute `data-history="true"` to input. This enhance the input field that shows you the last input values that has been recently entered. Use the up and down key's to scroll through previously typed input values. Use attribute `data-prevent-submit="true"` to cancel submitting form when user press `Enter` .

Enter value and press `Enter`, repeat, then press `Up` or `Down` arrows to access entered values.

```
<input type="text" data-role="input" data-history="true">
```

Custom buttons

You can create custom button for input control with attribute `data-custom-buttons="..."` . To add custom button, first-off - create array with your buttons and add array name as value for attribute `data-custom-buttons` . This array must be created before input component initiated.

User button

Cog button

```
<script>
  var customButtons = [
    {
      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>

<input type="text" data-role="input" data-custom-buttons="customButtons">
```

Each button must be defined as object with three properties: `html` - button caption, `cls` - classes for custom button, `onclick` - event for button when user clicked on it.

Autocomplete

You can create input with `autocomplete` feature. To create it, use attributes: `data-autocomplete` , `data-autocomplete-divider` , `data-autocomplete-list-height` .

Begin type one of Ukraine, USA, Canada, Marokko, Singapur

`<input type="text" data-role="input" data-autocomplete="Ukraine, USA, Canada, Marokko, Singapur">`

Options

Option	Data-*	Default	Desc
<code>history</code>	<code>data-history</code>	false	Enable history feature
<code>historyPreset</code>	<code>data-history-preset</code>		Preset values for history
<code>historyDivider</code>	<code>data-history-divider</code>		Divider for Preset values for history
<code>preventSubmit</code>	<code>data-prevent-submit</code>	false	Prevent submitting form when enabled history and user press <code>Enter</code> in input
<code>defaultValue</code>	<code>data-default-value</code>		Default value for input. Used when user click clear button or init input value
<code>size</code>	<code>data-size</code>		Set specific size for input
<code>prepend</code>	<code>data-prepend</code>		Add prepend label
<code>append</code>	<code>data-append</code>		Add append label
<code>clearButton</code>	<code>data-clear-button</code>	true	Add clear button to input
<code>clearButtonIcon</code>	<code>data-clear-button-icon</code>	<code></code>	Icon for clear button
<code>revealButton</code>	<code>data-reveal-button</code>	true	Add reveal button to input
<code>revealButtonIcon</code>	<code>data-reveal-button-icon</code>	<code></code>	Icon for reveal button
<code>customButtons</code>	<code>data-custom-buttons</code>		Array name with custom buttons

Events

Event	Data-*	Context
<code>onHistoryChange(val, history, index)</code>	<code>data-on-history-change</code>	input
<code>onHistoryUp(val, history, index)</code>	<code>data-on-history-up</code>	input
<code>onHistoryDown(val, history, index)</code>	<code>data-on-history-down</code>	input
<code>onClearClick(curr, new)</code>	<code>data-on-clear-click</code>	input
<code>onRevealClick(curr)</code>	<code>data-on-reveal-click</code>	input
<code>onInputCreate(el)</code>	<code>data-on-input-create</code>	input

Methods

You can use methods to interact with input component: `getHistory()` , `setHistory(history, append)` - history: array or string, append - bool, `clear()` , `toDefault()` , `enable()` , `disable()` , `toggleState()` .

```
var input = $(el).data('input');
input.clear();
input.toDefault();
input.toggleState();
```

Customize

You can customize your input with special attributes:

Option	Data-*	Desc
<code>clsComponent</code>	<code>data-cls-component</code>	Additional classes for input control.
<code>clsInput</code>	<code>data-cls-input</code>	Additional classes for input element.
<code>clsPrepend</code>	<code>data-cls-prepend</code>	Additional classes for input prepend data.
<code>clsAppend</code>	<code>data-cls-append</code>	Additional classes for input append data.
<code>clsClearButton</code>	<code>data-cls-clear-button</code>	Additional classes for input clear button.
<code>clsRevealButton</code>	<code>data-cls-reveal-button</code>	Additional classes for input reveal button.
<code>clsCustomButton</code>	<code>data-cls-custom-button</code>	Additional classes for input custom button.

Test value

```
<input type="text" data-role="input"
  data-cls-input="text-bold bg-dark fg-white text-center"
  data-clear-button="false">
```



Additional classes

You can use additional classes to change input size: `.input-large` and `.input-small`.

Input small

input

Input value

data-role=input

Input value

Input regular

input

Input value

data-role=input

Input value

Input large

input

Input value

data-role=input

Input value