# Configuration - ### `text` \*\*Type:\*\* `string` \*\*Default:\*\* `""` (\*empty string\*) \*\*Description:\*\* The modal's text. It can either be added as a configuration under the key `text` (as in the example below), or passed as the first and only parameter (e.g. `swal("Hello world!")`), or the second one if you have multiple string parameters (e.g. `swal("A title", "Hello world!")`). \*\*Example:\*\* ```js swal({ text: "Hello world!", }); ``` - ### `title` \*\*Type:\*\* `string` \*\*Default:\*\* `""` (\*empty string\*) \*\*Description:\*\* The title of the modal. It can either be added as a configuration under the key `title` (as in the example below), or passed as the first string parameter – as long as it's not the only one – of the `swal` function (e.g. `swal("Here's a title!", "Here's some text")`). \*\*Example:\*\* ```js swal({ title: "Here's a title!", }); ``` - ### `icon` \*\*Type:\*\* `string` \*\*Default:\*\* `""` (\*empty string\*) \*\*Description:\*\* An icon for the modal. SweetAlert comes with 4 built-in icons that you can use: - `"warning"` - `"error"` - `"success"` - `"info"` It can either be added as a configuration under the key `icon`, or passed as the third string parameter of the `swal` function (e.g. `swal("Title", "Text", "success")`). \*\*Example:\*\* ```js swal({ icon: "success", }); ``` - ### `button` \*\*Type:\*\* `string|boolean|ButtonOptions` \*\*Default:\*\* ```js { text: "OK", value: true, visible: true, className: "", closeModal: true, } ``` \*\*Description:\*\* The confirm button that's shown by default. You can change its text by setting `button` to a string, or you can tweak more setting by passing a `ButtonOptions` object. Setting it to `false` hides the button. \*\*Examples:\*\* ```js swal({ button: "Coolio", }); ``` ```js swal({ button: { text: "Hey ho!", }, }); ``` ```js swal("Hello world!", { button: false, }); ``` - ### `buttons` \*\*Type:\*\* `boolean|string[]|ButtonOptions[]|ButtonList` \*\*Default:\*\* ```js { cancel: { text: "Cancel", value: null, visible: false, className: "", closeModal: true, }, confirm: { text: "OK", value: true, visible: true, className: "", closeModal: true } } ``` \*\*Description:\*\* Specify the exact amount of buttons and their behaviour. If you use an array, you can set the elements as strings (to set only the text), a list of `ButtonOptions`, or a combination of both. You can also set one of the elements to `true` to simply get the default options. If you want more than just the predefined cancel and confirm buttons, you need to specify a `ButtonList` object, with keys (the button's namespace) pointing to `ButtonOptions`. You can also specify `false` to hide all buttons (same behaviour as the `button` option). \*\*Examples:\*\* ```js swal({ buttons: ["Stop", "Do it!"], }); ``` ```js swal({ buttons: [true, "Do it!"], }); ``` ```js swal("Hello world!", { buttons: false, }); ``` ```js swal({ buttons: { cancel: true, confirm: true, }, }); ``` ```js swal({ buttons: { cancel: true, confirm: "Confirm", roll: { text: "Do a barrell roll!", value: "roll", }, }, }); ``` - ### `content` \*\*Type:\*\* `Node|string` \*\*Default:\*\* `null` \*\*Description:\*\* For custom content, beyond just text and icons. \*\*Examples:\*\* ```js swal({ content: "input", }); ``` ```js swal({ content: { element: "input", attributes: { placeholder: "Type your password", type: "password", }, }, }); ``` ```js var slider = document.createElement("input"); slider.type = "range"; swal({ content: slider, }); ``` - ### `className` \*\*Type:\*\* `string` \*\*Default\*\*: `""` (\*empty string\*) \*\*Description\*\*: Add a custom class to the SweetAlert modal. This is handy for changing the appearance. \*\*Example\*\*: ```js swal("Hello world!", { className: "red-bg", }); ``` - ### `closeOnClickOutside` \*\*Type:\*\* `boolean` \*\*Default:\*\* `true` \*\*Description:\*\* Decide whether the user should be able to dismiss the modal by clicking outside of it, or not. \*\*Example:\*\* ```js swal({ closeOnClickOutside: false, }); ``` - ### `closeOnEsc` \*\*Type:\*\* `boolean` \*\*Default:\*\* `true` \*\*Description:\*\* Decide whether the user should be able to dismiss the modal by hitting the ESC key, or not. \*\*Example:\*\* ```js swal({ closeOnEsc: false, }); ``` - ### `dangerMode` \*\*Type:\*\* `boolean` \*\*Default:\*\* `false` \*\*Description:\*\* If set to `true`, the confirm button turns red and the default focus is set on the cancel button instead. This is handy when showing warning modals where the confirm action is dangerous (e.g. deleting an item). \*\*Example:\*\* ```js swal("Are you sure?", { dangerMode: true, buttons: true, }); ``` - ### `timer` \*\*Type:\*\* `number` \*\*Default:\*\* `null` \*\*Description:\*\* Closes the modal after a certain amount of time (specified in ms). Useful to combine with `buttons: false`. \*\*Example:\*\* ```js swal("This modal will disappear soon!", { buttons: false, timer: 3000, }); ``` # Methods | Name | Description | Example | | ---- | ----------- | ------- | | `close` | Closes the currently open SweetAlert, as if you pressed the cancel button. | `swal.close()` | | `getState` | Get the state of the current SweetAlert modal. | `swal.getState()` | | `setActionValue` | Change the promised value of one of the modal's buttons. You can either pass in just a string (by default it changes the value of the confirm button), or an object. | `swal.setActionValue({ confirm: 'Text from input' })` | | `stopLoading` | Removes all loading states on the modal's buttons. Use it in combination with the button option `closeModal: false`. | `swal.stopLoading()` # Theming - ### `swal-overlay` \*\*Example:\*\* ```css .swal-overlay { background-color: rgba(43, 165, 137, 0.45); } ``` - ### `swal-modal` \*\*Example:\*\* ```css .swal-modal { background-color: rgba(63,255,106,0.69); border: 3px solid white; } ``` - ### `swal-title` \*\*Example:\*\* ```css .swal-title { margin: 0px; font-size: 16px; box-shadow: 0px 1px 1px rgba(0, 0, 0, 0.21); margin-bottom: 28px; } ``` - ### `swal-text` \*\*Example:\*\* ```css .swal-text { background-color: #FEFAE3; padding: 17px; border: 1px solid #F0E1A1; display: block; margin: 22px; text-align: center; color: #61534e; } ``` - ### `swal-footer` \*\*Example:\*\* ```css .swal-footer { background-color: rgb(245, 248, 250); margin-top: 32px; border-top: 1px solid #E9EEF1; overflow: hidden; } ``` - ### `swal-button` \*\*Description:\*\* The modal's button(s). It has an extra class that changes depending on the button's type, in the format `swal-button--{type}`. The extra class for the confirm button for example is `swal-button--confirm`. \*\*Example:\*\* ```css .swal-button { padding: 7px 19px; border-radius: 2px; background-color: #4962B3; font-size: 12px; border: 1px solid #3e549a; text-shadow: 0px -1px 0px rgba(0, 0, 0, 0.3); } ```