Bootstrap

Bootstrap Plugins

Capgemini

Lesson Objectives

- **Working with jQuery Plugins**

Bootstrap jQuery Plugins

➢ Bootstrap comes bundled with 12 jQuery plugins that extend the features and can add more interaction to your site.

➢Plugins can be included individually (using Bootstrap's individual *.js files) or all at once (using bootstrap.js or the minified bootstrap.min.js).

➢Both bootstrap.js and bootstrap.min.js contain all plugins in a single file.

➢We can use all Bootstrap plugins either through the markup API using data-* attributes without writing a single line of JavaScript or purely through the JavaScript API.

➢To disable data attribute API and access plug-ins through JavaScript API.

- Disable all plugins *:* **$(document).off('.data-api')**
- Disable a specific plugin*:* **$(document).off('.alert.data-api')**

Working with Methods

- All JavaScript API methods can accept an optional options object, a string which targets a particular method, or nothing (which initiates plugin with default behavior)
- To initialized with defaults
  - $(selector).ApiMethod()
- To initialized with Options
  - $(selector). ApiMethod({options})
- To initialized with methods to be invoked
  - $(selector).ApiMethod('methodName')

Working with Events

- Bootstrap provides custom events for most plugins' unique actions.
- All Bootstrap events are namespaced i.e. <event>.bs.<component>
  - **eg**: show.bs.modal
- If (show) is triggered at the start of an event, its past participle form (shown) is triggered on the completion of an action.

Transition Plug-in

- Transition plugin provides simple transition effects.
- We can apply transitions on Modal, Tab, Alerts and Carousel plugins
- These are the classes for the different transitions:
  - .fade (fade out)
  - .fade.in (fade in)
  - .collapse (collapse out)
  - .collapse.in (collapse in)
- If you use the modal then the fade class adds animation to the modal
  - .fade (fade out to top 0)
  - .fade.in (fade in from top to 50% height)

Modal Plug-in

- A Modal dialog is a dialog that overlays the rest of the web page and take the control of page.
- A modal is a child window that is layered over its parent window.
- To invoke the modal window, we can use a button or a link with data-toggle ="modal" along with a data-target="#identifier" or href="#identifier" to target a specific modal
- There are 3 distinct parts inside a Modal: header, body and footer
- To include this plugin functionality individually we need to use modal.js

Modal Plug-in options

Options	Default Value	Data attribute	Description
backdrop	boolean or the string 'static' Default: true	data-backdrop	specify static for a backdrop, if you don't want the modal to be closed when the user clicks outside of the modal.
keyboard	boolean Default: true	data-keyboard	Closes the modal when escape key is pressed; set to false to disable.
show	boolean Default: true	data-show	Shows the modal when initialized.
remote	path	data-remote	Using the jQuery .load method, inject content into the modal body. If an href with a valid URL is added, it will load that content. An example of this is shown below:
	Default: false		<a data-toggle="modal" href="remote.html" data-target="#modal">Click me

Modal Plug-in methods

Method	Description	Example
.modal(options)	Activates your content as a modal. Accepts an optional options object.	$('#identifier').modal({ keyboard: false })
.modal('toggle')	Manually toggles a modal	$('#identifier').modal('toggle')
.modal('show')	Manually opens a modal	$('#identifier').modal('show')
.modal('hide')	Manually hides a modal	$('#identifier').modal('hide')

Modal Plug-in Events

Event	Description	Example

show.bs.modal	Fired after the show method is called.	$('#identifier').on('show.bs.modal', function () {
		// do something…
		})

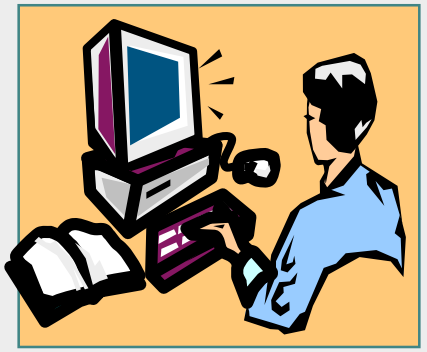
shown.bs.modal	Fired when the modal has been made visible to the user (will wait for CSS transitions to complete).	$('#identifier').on('shown.bs.modal', function () {
		// do something…
		})

hide.bs.modal	Fired when the hide instance method has been called.	$('#identifier').on('hide.bs.modal', function () {
		// do something…
		})

hidden.bs.modal	Fired when the modal has finished being hidden from the user.	$('#identifier').on('hidden.bs.modal', function () {
		// do something…
		})

Demo

- Modal-PluginJS
- Modal-Plugin



Carousel Plug-in

- A Carousel gives you a picture Carousel that can cycle through a series of images.
- Carousel works best when all of the images have the same resolution
- Carousel can also have a caption inside of each item that typically consists of some header text and some additional text
- We can set the interval to how frequently to swap the pictures
- We can also control the carousel using user pause, previous and next buttons.
- To include this plugin functionality individually we need carousel.js

Carousel Plug-in options

Option Name	Default Value	Data attribute	Description
interval	number Default: 5000	data-interval	The amount of time to delay between automatically cycling an item. If false, carousel will not automatically cycle.
pause	string Default: "hover"	data-pause	Pauses the cycling of the carousel on mouseenter and resumes the cycling of the carousel on mouseleave.
wrap	boolean Default: true	data-wrap	Whether the carousel should cycle continuously or have hard stops.

Carousel Plug-in methods

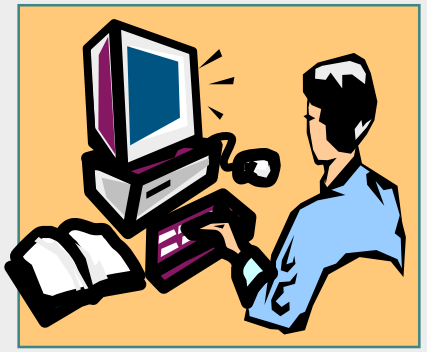
Method	Description	Example
.carousel(options)	Initializes the carousel with an optional options object and starts cycling through items.	$('#identifier').carousel({ interval: 2000 })
.carousel('cycle')	Cycles through the carousel items from left to right.	$('#identifier').carousel('cycle')
.carousel('pause')	Stops the carousel from cycling through items.	$('#identifier')..carousel('pause')
.carousel(number)	Cycles the carousel to a particular frame (0 based, similar to an array).	$('#identifier').carousel(number)
.carousel('prev')	Cycles to the previous item.	$('#identifier').carousel('prev')
.carousel('next')	Cycles to the next item..	$('#identifier').carousel('next')

Carousel Plug-in Events

Event	Description	Example
slide.bs.carousel	This event fires immediately when the slide instance method is invoked..	$('#identifier').on('slide.bs.carousel', function () { // do something… })
slid.bs.carousel	This event is fired when the carousel has completed its slide transition.	$('#identifier').on('slid.bs.carousel', function () { // do something… })

Demo

- Carousel Plug-in



Collapse Plug-in

- Collapse plugin makes it easy to make collapsing divisions of the page.
- It is used to create collapsible groups or accordion
  - **data-toggle="collapse"** is added to the link on which we click to expand or collapse the component.
  - **href** or a **data-target attribute** is added to the parent component, whose value is id of the child component.
  - **data-parent** attribute is added to is added for creating accordion like effect.
  - ***.collapse*** class is used to hide the content and ***.show*** class is used to show the content
- To include this plugin functionality individually we need to use collapse.js

Collapse Plug-in options

Option	Default Value	Data attribute	Description
parent	selector Default: false	data-parent	If selector then all collapsible elements under the specified parent will be closed when this collapsible item is shown. (similar to traditional accordion behavior - this dependent on the accordion-group class)
toggle	boolean Default: true	data-toggle	Toggles the collapsible element on invocation.

CollapsePlug-in methods

Method	Description	Example
Options: .collapse(options)	Activates your content as a collapsible element. Accepts an optional options object.	$('#identifier').collapse({ toggle: false })
Toggle: .collapse('toggle')	Toggles a collapsible element to shown or hidden.	$('#identifier').collapse('toggle')
Show: .collapse('show')	Shows a collapsible element.	$('#identifier').collapse('show')
Hide: .collapse('hide')	Hides a collapsible element.	$('#identifier').collapse('hide')

Collapse Plug-in Events

Event	Description	Example

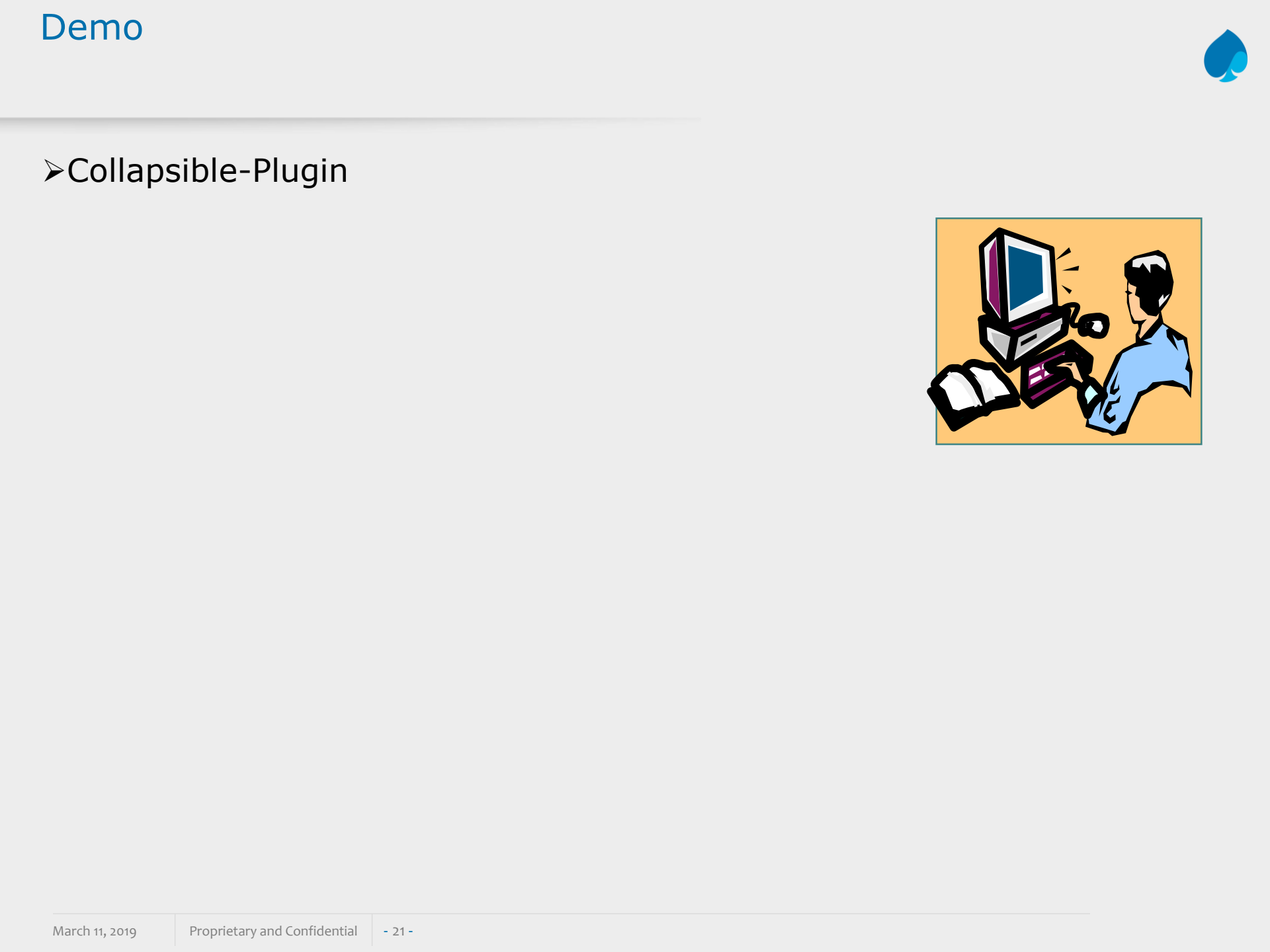
show.bs.collapse	Fired after the show method is called.	$('#identifier').on('show.bs.collapse', function () {
		// do something…
		})

shown.bs.collapse	This event is fired when a collapse element has been made visible to the user (will wait for CSS transitions to complete).	$('#identifier').on('shown.bs.collapse', function () {
		// do something…
		})

hide.bs.collapse	Fired when the hide instance method has been called.	$('#identifier').on('hide.bs.collapse', function () {
		// do something…
		})

hidden.bs.collapse	This event is fired when a collapse element has been hidden from the user (will wait for CSS transitions to complete).	$('#identifier').on('hidden.bs.collapse', function () {
		// do something…
		})

Demo



- Collapsible-Plugin

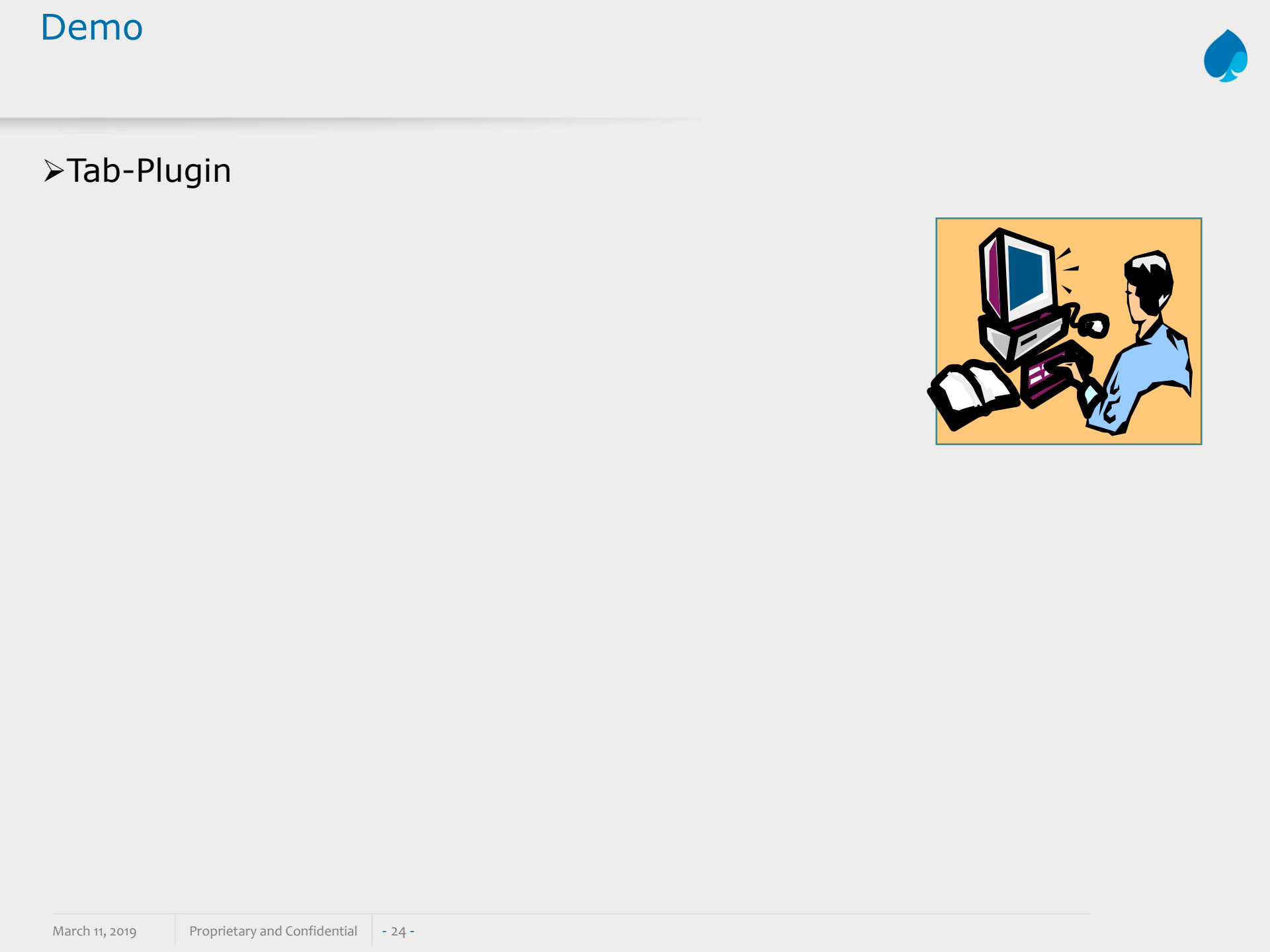
Tab Plug-in

- Tabs are used to arrange a navigational menu with links to other pages or sections of the site.
- Using Tab plugin, we can transition through panes of local content in tabs or pills
- Adding the *.nav* and *.nav-tabs* classes to the tab ul will apply the Bootstrap tab styling, while adding the *.nav* and *.nav-pills* classes will apply pill styling.
- .tab() method activates a tab element and content container. Tab should have either a data-target or an href targeting a container node in the DOM.
- To include this plugin functionality individually we need tab.js

Tab Plug-in Events

Event	Description	Example
show.bs.tab	This event fires on tab show, but before the new tab has been shown. Use event.target and event.relatedTarget to target the active tab and the previous active tab (if available) respectively.	$('a[data-toggle="tab"]').on('show.bs.tab', function (e) { e.target // activated tab e.relatedTarget // previous tab })
shown.bs.tab	This event fires on tab show after a tab has been shown. Use event.target and event.relatedTarget to target the active tab and the previous active tab (if available) respectively.	$('a[data-toggle="tab"]').on('shown.bs.tab', function (e) { e.target // activated tab e.relatedTarget // previous tab })

Demo



- Tab-Plugin

Tooltip Plug-in

- The Tooltip component can be used to display small pop-up box that appears when the user moves the mouse pointer over an element.
- Add the data-toggle="tooltip" attribute to an element, to create a tooltip. Use the ***title*** attribute to specify the text that should be displayed inside the tooltip.
- To display Tooltip in four directions such as top, bottom, left or right side by use the ***data-placement*** attribute on the element

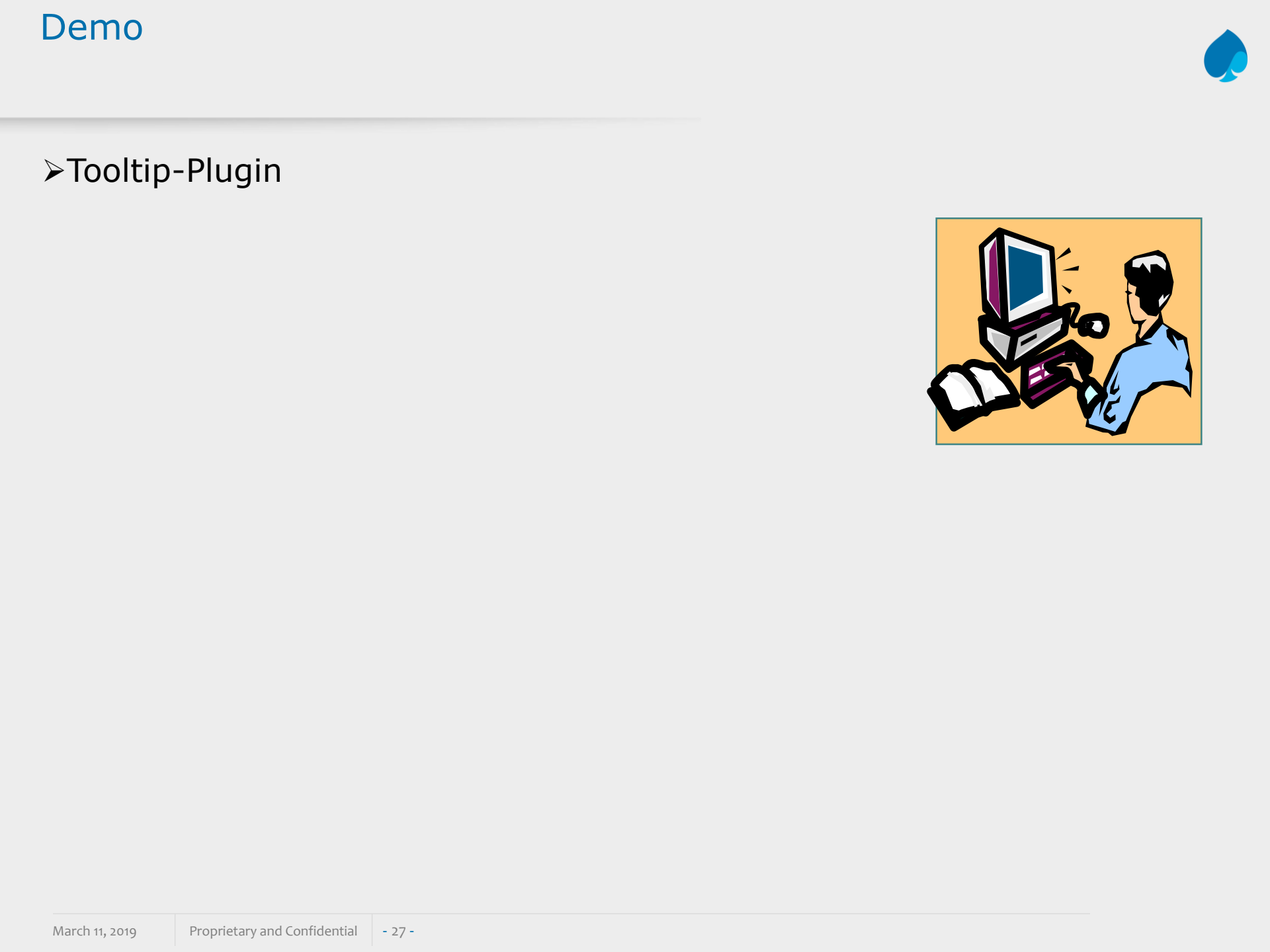
Tooltips are not CSS-only plugins, and must therefore be initialized with jQuery: select the specified element and call the tooltip() method.

- To include this plugin functionality individually we need tooltip.js

Tooltip Plug-in Events

Event	Description	Example
show.bs.tooltip	This event triggers when the tooltip is about to be shown	$("[data-toggle='tooltip']").on ('show.bs.tooltip', function(){ alert('The tooltip is about to be shown.'); });
shown.bs.tooltip	This event triggers when the tooltip is fully shown (after CSS transitions have completed)	$("[data-toggle='tooltip']").on ('shown.bs.tooltip', function(){ alert('The tooltip is now fully shown.'); });
hide.bs.tooltip	This event triggers when the tooltip is about to be hidden	$("[data-toggle='tooltip']").on('hide.bs.tooltip', function(){ alert('The tooltip is about to be hidden.'); });
hidden.bs.tooltip	This event triggers when the tooltip is about to be hidden	$("[data-toggle='tooltip']").on('hidden.bs.tooltip', function(){ alert('The tooltip is now hidden.'); });

Demo



- Tooltip-Plugin