Package: aos (via r-universe)

September 16, 2024

Type Package

Title Animate on Scroll Library for 'shiny'

Version 0.1.1

Description Trigger animation effects on scroll on any HTML element of 'shiny' and 'rmarkdown', such as any text or plot, thanks to the 'AOS' Animate On Scroll jQuery library <http://michalsnik.github.io/aos/>.

License MIT + file LICENSE

URL https://felixluginbuhl.com/aos, https://github.com/lgnbhl/aos

BugReports https://github.com/lgnbhl/aos/issues

Imports jsonlite, htmltools

Suggests shiny

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Repository https://lgnbhl.r-universe.dev

RemoteUrl https://github.com/lgnbhl/aos

RemoteRef HEAD

RemoteSha 50749d1466a354edf15e4d3e05f9e17645050f0b

Contents

---

aos   *Animate element on scroll*

Description

Animate on scroll a Shiny or R Markdown element.

Usage

```
aos(
  element,
  animation,
  offset = "120",
  duration = "400",
  easing = "ease",
  delay = "0",
  anchor = "",
  anchor_placement = "top-bottom",
  once = FALSE,
  ...
)
```

Arguments

`element`	An shiny or rmarkdown element.
`animation`	An animation from [AOS](#)
`offset`	string. Change offset to trigger animations sooner or later, px
`duration`	string. Duration of animation in ms
`easing`	string. Choose timing function to ease elements in different ways
`delay`	string. Delay animation in ms
`anchor`	string. Anchor placement
`anchor_placement`	
	string. Anchor placement - which one position of element on the screen should trigger animation
`once`	boolean.
`...`	Additional class element

Value

Javascript code animating the element.

Examples

```
if (interactive()) {
 library(shiny)
  shinyApp(
   ui = fluidPage(
     use_aos(), # add use_aos() in the UI
     aos(h1("Shiny with AOS - Animate On Scroll"), animation = "fade-zoom-in"),
     aos(textOutput("text"), animation = "fade-up")
   ),
  server <- function(input, output, session){
    output$text <- renderText({ print("An animated text.") })
  }
 )
 }
```

`html_dependencies_aos`   *aos dependencies*

Description

Get aos html dependencies.

Usage

```
html_dependencies_aos(use_cdn = TRUE)
```

Arguments

use_cdn   boolean. TRUE to use a Content Delivery Network (CDN), true by default, or
                 false to use locally stored files.

`use_aos`   *Use aos*

Description

Use aos

Usage

```
use_aos(
  disable = FALSE,
  startEvent = "DOMContentLoaded",
  initClassName = "aos-init",
  animatedClassName = "aos-animate",
  useClassNames = FALSE,
  disableMutationObserver = FALSE,
  debounceDelay = "50",
  throttleDelay = "99",
  offset = "120",
  duration = "400",
  easing = "ease",
  delay = "0",
  use_cdn = TRUE
)
```

Arguments

disable	string. Condition when AOS should be disabled, for example 'mobile'
startEvent	string. Name of the event dispatched on the document, that AOS should initialize on
initClassName	string. Class applied after initialization
animatedClassName	
	string. Class applied on animation
useClassNames	boolean. If true, will add content of data-aos as classes on scroll
disableMutationObserver	
	boolean. Disables automatic mutations detections (advanced)
debounceDelay	string. The delay on debounce used while resizing window (advanced)
throttleDelay	string. The delay on throttle used while scrolling the page (advanced)
offset	string. Change offset to trigger animations sooner or later, in px
duration	string. Duration of animation in ms.
easing	string. Timing function to animate elements nicely
delay	string. Delay animation (ms)
use_cdn	boolean. TRUE to use a Content Delivery Network (CDN), true by default, or false to use locally stored files.

Value

Javascript code initializing aos with dependencies.

Examples

```
if (interactive()) {
 library(shiny)
```

```
  shinyApp(
   ui = fluidPage(
     use_aos(), # add use_aos() in the UI
     aos(h1("Shiny with AOS - Animate On Scroll"), animation = "fade-zoom-in"),
     aos(textOutput("text"), animation = "fade-up")
   ),
  server <- function(input, output, session){
    output$text <- renderText({ print("An animated text.") })
  }
 )
}
```

---

use_aos_refresh	*Refresh aos*

Description

Recalculate all offsets and positions of elements (called on window resize).

Usage

```
use_aos_refresh()
```

Details

By default AOS is watching for DOM changes and if there are any new elements loaded asynchronously or when something is removed from DOM it calls refreshHard automatically. In browsers that don't support MutationObserver like IE you might need to call AOS.refreshHard() by yourself.

Value

Javascript code

Examples

```
use_aos_refresh()
```

use_aos_refresh_hard   *Refresh hard aos*

Description

Reinit array with AOS elements and trigger refresh (called on DOM changes that are related to aos elements)

Usage

```
use_aos_refresh_hard()
```

Value

Javascript code

Examples

```
use_aos_refresh_hard()
```

Index