

[Source](#) > Javascript examples

## Javascript examples

### Basic external Javascript lib wrappers

To expedite fast development, FastHTML comes with several built-in Javascript and formatting components. These are largely provided to demonstrate FastHTML JS patterns. There's far too many JS libs for FastHTML to wrap them all, and as shown here the code to add FastHTML support is very simple anyway.

### light\_media

[source](#)

```
light_media (css:str)
```

*Render light media for day mode views*

Type		Details
css	str	CSS to be included in the light media query

```
light_media('.body {color: green;})
```

```
<style>@media (prefers-color-scheme: light) {.body {color: green;}}</style>
```

### dark\_media

[source](#)

```
dark_media (css:str)
```

*Render dark media for night mode views*

Type		Details
css	str	CSS to be included in the dark media query

```
dark_media('.body {color: white;})
```

```
<style>@media (prefers-color-scheme: dark) {.body {color: white;}}</style>
```

### MarkdownJS

[source](#)

```
MarkdownJS (sel='.marked')
```

*Implements browser-based markdown rendering.*

	Type	Default	Details
sel	str	.marked	CSS selector for markdown elements

Usage example [here](#).

```
__file__ = '../..fasthtml/katex.js'
```

source

```
KatexMarkdownJS (sel='.marked', inline_delim='$', display_delim='$$',  
math_envs=None)
```

	Type	Default	Details
sel	str	.marked	CSS selector for markdown elements
inline_delim	str	\$	Delimiter for inline math
display_delim	str	\$\$	Delimiter for long math
math_envs	NoneType	None	List of environments to render as display math

KatexMarkdown usage example:

```

longexample = r"""
Long example:


$$\begin{array}{c} \\ \\ \nabla \times \vec{\mathbf{B}} = \frac{1}{c} \nabla \times \frac{\partial \vec{\mathbf{E}}}{\partial t} \\ = \frac{4\pi}{c} \vec{\mathbf{j}} \quad \nabla \cdot \vec{\mathbf{E}} = 4\pi \rho \\ \nabla \times \vec{\mathbf{E}} = -\frac{1}{c} \frac{\partial \vec{\mathbf{B}}}{\partial t} \\ \nabla \cdot \vec{\mathbf{B}} = 0 \end{array}$$


"""

app, rt = fast_app(hdrs=[KatexMarkdownJS()])

@rt('/')
def get():

```

```
return Titled("Katex Examples",
    # Assigning 'marked' class to components renders content as markdown
    P(cls='marked')("Inline example:  $\sqrt{3x-1}+(1+x)^2$ "),
    Div(cls='marked')(longexample)
)
```

HighlightJS

[source](#)

```
HighlightJS (sel='pre code', langs:str|list|tuple='python', light='atom-
one-light', dark='atom-one-dark')
```

Implements browser-based syntax highlighting. Usage example [here](#).

Type		Default	Details
sel	str	pre code	CSS selector for code elements. Default is industry standard, be careful before adjusting it
langs	str   list   tuple	python	Language(s) to highlight
light	str	atom-one-light	Light theme
dark	str	atom-one-dark	Dark theme

SortableJS

[source](#)

```
SortableJS (sel='.sortable', ghost_class='blue-background-class')
```

Type		Default	Details
sel	str	.sortable	CSS selector for sortable elements
ghost_class	str	blue-background-class	When an element is being dragged, this is the class used to distinguish it from the rest