



Dash HTML Components

Dash is a web application framework that provides pure Python abstraction around HTML, CSS, and JavaScript.

Instead of writing HTML or using an HTML templating engine, you compose your layout using Python with the Dash HTML Components module ([dash.html](#)).

The Dash HTML Components module is part of Dash and you'll find the source for it in the [Dash GitHub repo](#).

For production Dash apps, styling and layout of Dash HTML Components should be managed with Dash Enterprise [Design Kit](#).

Here is an example of a simple HTML structure:

```
from dash import html

html.Div([
    html.H1('Hello Dash'),
    html.Div([
        html.P('Dash converts Python classes into HTML'),
        html.P('This conversion happens behind the scenes by Dash's JavaScript front-end')
    ])
])
```

which gets converted (behind the scenes) into the following HTML in your web-app:

```
<div>
  <h1>Hello Dash</h1>
  <div>
    <p>Dash converts Python classes into HTML</p>
    <p>This conversion happens behind the scenes by Dash's JavaScript front-end</p>
  </div>
</div>
```

If you're not comfortable with HTML, don't worry! You can get 95% of the way there with just a few elements and attributes.

If you want to use [Markdown](#) in your app, you can use the [Dash Core Components](#) (`dash.dcc`) [Markdown](#) component:

```
from dash import dcc

dcc.Markdown('''
#### Dash and Markdown

Dash supports [Markdown](http://commonmark.org/help).

Markdown is a simple way to write and format text.
It includes a syntax for things like bold text and italics*,
[links](http://commonmark.org/help), inline `code` snippets, lists,
quotes, and more.
''')
```

Dash and Markdown

Dash supports [Markdown](#).

Markdown is a simple way to write and format text. It includes a syntax for things like **bold text** and *italics*, [links](#), inline code snippets, lists, quotes, and more.

If you're using HTML components, then you also have access to properties like `style`, `class`, and `id`. All of these attributes are available in the Python classes.

The HTML elements and Dash classes are mostly the same but there are a few key differences:

- The `style` property is a dictionary
- Properties in the `style` dictionary are camelCased
- The `class` key is renamed as `className`
- Style properties in pixel units can be supplied as just numbers without the `px` unit

Let's take a look at an example.

```
from dash import html

html.Div([
    html.Div('Example Div', style={'color': 'blue', 'fontSize': 14}),
    html.P('Example P', className='my-class', id='my-p-element')
], style={'marginBottom': 50, 'marginTop': 25})
```

That Dash code will render this HTML markup:

```
<div style="margin-bottom: 50px; margin-top: 25px;">

  <div style="color: blue; font-size: 14px">
    Example Div
  </div>

  <p class="my-class", id="my-p-element">
    Example P
  </p>

</div>
```

Note: If you need to directly render a string of raw, unescaped HTML, you can use the `DangerouslySetInnerHTML` component which is provided by the `dash-dangerously-set-inner-html` library.

Full elements reference:

- html.A
- html.Abbr
- html.Acronym
- html.Address
- html.Area
- html.Article
- html.Aside
- html.Audio
- html.B
- html.Base
- html.Basefont
- html.Bdi
- html.Bdo
- html.Big
- html.Blink
- html.Blockquote
- html.Br
- html.Button
- html.Canvas
- html.Caption
- html.Center
- html.Cite
- html.Code
- html.Col
- html.Colgroup
- html.Content
- html.Data
- html.Datalist
- html.Dd
- html.Del
- html.Details
- html.Dfn
- html.Dialog
- html.Div
- html.DI
- html.Dt
- html.Em
- html.Embed
- html.Fieldset
- html.Figcaption
- html.Figure
- html.Font
- html.Footer
- html.Form
- html.Frame
- html.Frameset
- html.H1
- html.H2
- html.H3
- html.H4
- html.H5
- html.H6
- html.Header
- html.Hgroup
- html.Hr
- html.I
- html.Iframe
- html.Img
- html.Jns
- html.Kbd
- html.Keygen
- html.Label
- html.Legend
- html.Li
- html.Link
- html.Main
- html.MapEl
- html.Mark
- html.Marquee
- html.Meta
- html.Meter
- html.Nav
- html.Nobr
- html.Noscript
- html.ObjectEl
- html.OI
- html.Optgroup
- html.Option
- html.Output
- html.P
- html.Param
- html.Picture
- html.Plaintext
- html.Pre
- html.Progress
- html.Q
- html.Rb
- html.Rp
- html.Rt
- html.Rtc
- html.Ruby
- html.S
- html.Samp
- html.Script
- html.Section
- html.Select
- html.Shadow
- html.Slot
- html.Small
- html.Source
- html.Spacer
- html.Span
- html.Strike
- html.Strong
- html.Sub
- html.Summary
- html.Sup
- html.Table
- html.Tbody
- html.Td
- html.Template
- html.Textarea
- html.Tfoot
- html.Th
- html.Thead
- html.Time
- html.Title
- html.Tr
- html.Track
- html.U
- html.UI
- html.Var
- html.Video
- html.Wbr
- html.Xmp

Products

Dash
Consulting and Training

Pricing

Enterprise Pricing
Resources

About Us

Careers
Resources
Blog

Support

Community Support
Graphing Documentation

Join our mailing list

Sign up to stay in the loop with all things Plotly — from Dash Club to product updates, webinars, and more!

SUBSCRIBE