plotly

Star 23,446

Dash Python > ***Dash HTML Components***

> Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. **Sign up for early access now.**

Dash HTML Components

Dash is a web app 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`).

Import `dash.html` with:

```
from dash import html
```

The Dash HTML Components module is part of Dash and you'll find the source for it in the **Dash GitHub repo**.

> **Tip**: In production Dash apps, we recommend using Dash Enterprise **Design Kit** to style Dash HTML Components.

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** 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.

Sign up for Dash Club → Two free cheat sheets plus updates from Chris Parmer and Adam Schroeder delivered to your inbox every two months. Includes tips and tricks, community apps, and deep dives into the Dash architecture. **Join now**.

HTML Component Properties

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>
```

n_clicks and disable_n_clicks

All Dash HTML components have an `n_clicks` property, which is an integer that represents the number of times the element has been clicked. You can use `n_clicks` to trigger a callback and use the value of `n_clicks` in your callback logic.

In this example, we capture the `n_clicks` value from the `html.Div` with ID `click-div` and output it to the `html.P` with ID `click-output`. `n_clicks` uses an event listener to capture user click events on the element and increment the n_clicks value.

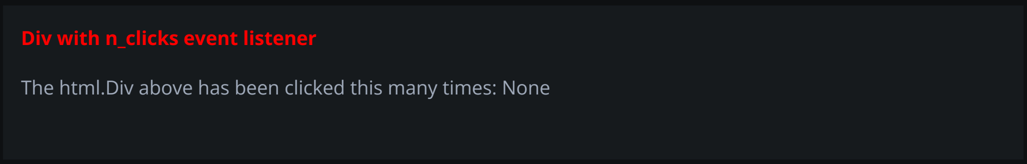
```
from dash import Dash, html, Input, Output, callback

app = Dash()

app.layout = html.Div(
    [
        html.Div(
            "Div with n_clicks event listener",
            id="click-div",
            style={"color": "red", "font-weight": "bold"},
        ),
        html.P(id="click-output"),
    ]
)


@callback(
    Output("click-output", "children"),
    Input("click-div", "n_clicks")
    )
def click_counter(n_clicks):
    return f"The html.Div above has been clicked this many times: {n_clicks}"


app.run(debug=True)
```



Div with n_clicks event listener

The html.Div above has been clicked this many times: None

Many Dash HTML components are rarely intended to be clicked (in the example above, it's unusual that the html.Div is clickable—a better design choice would be to use a button). Even when you use elements like html.Div that you don't intend for the user to click, the n_clicks event listener causes screen-reading software to interpret the elements as clickable, which can be confusing.

In Dash 2.8 and later, Dash HTML components are improved for better control over the n_clicks event listener:

- If you don't give your HTML component an ID, the n_clicks event listener is not added.
- If your HTML component does have an ID but you don't need to capture clicks, you can disable the n_clicks event listener by setting disable_n_clicks=True.

Here is the same example as above, but we've decided that we don't need to capture clicks, so we've disabled n_clicks on the html.Div (the callback is for illustrative purposes):

```
from dash import Dash, html, Input, Output, callback

app = Dash()

app.layout = html.Div(
    [
        html.Div(
            "Div without n_clicks event listener",
            id="click-div-2",
            disable_n_clicks=True,
            style={"color": "red", "font-weight": "bold"},
        ),
        html.P(id="click-output-2", disable_n_clicks=True),
    ]
)


@callback(
    Output("click-output-2", "children"),
    Input("click-div-2", "n_clicks")
    )
def click_counter(n_clicks):
    return f"The html.Div above has been clicked this many times: {n_clicks}"
```

```
app.run(debug=True)
```

Div without n_clicks event listener

The html.Div above has been clicked this many times: None

With `disable_n_clicks=True`, we convey to screen reader assisted users that the `html.Div` is not clickable.

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.Dl**
- **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.Ins**
- **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.Ol
- 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.Ul**
- **html.Var**
- **html.Video**
- **html.Wbr**
- **html.Xmp**

Dash Python ***> Dash HTML Components***

Products

Dash

Consulting and Training

Pricing

Enterprise Pricing

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

Terms of Service Privacy Policy