FastHTML

Source > SVG

SVG

Simple SVG FT elements

```
from nbdev.showdoc import show_doc
```

You can create SVGs directly from strings, for instance (as always, use `NotStr` or `Safe` to tell FastHTML to not escape the text):

```
svg = '<svg width="50" height="50"><circle cx="20" cy="20" r="15" fill="red"></circle>
show(NotStr(svg))
```

You can also use libraries such as fa6-icons.

To create and modify SVGs using a Python API, use the FT elements in `fasthtml.svg`, discussed below.

Note: `fasthtml.common` does NOT automatically export SVG elements. To get access to them, you need to import `fasthtml.svg` like so

```
from fasthtml.svg import *
```

Svg

source

```
Svg (*args, viewBox=None, h=None, w=None, height=None, width=None,
     xmlns='http://www.w3.org/2000/svg', **kwargs)
```

An SVG tag; xmlns is added automatically, and viewBox defaults to height and width if not provided

To create your own SVGs, use `SVG`. It will automatically set the `viewBox` from height and width if not provided.

All of our shapes will have some convenient kwargs added by using `ft_svg`:

---

ft_svg

source

```
ft_svg (tag:str, *c, transform=None, opacity=None, clip=None, mask=None,
        filter=None, vector_effect=None, pointer_events=None,
        target_id=None, hx_vals=None, id=None, cls=None, title=None,
```

```
        style=None, accesskey=None, contenteditable=None, dir=None,
        draggable=None, enterkeyhint=None, hidden=None, inert=None,
        inputmode=None, lang=None, popover=None, spellcheck=None,
        tabindex=None, translate=None, hx_get=None, hx_post=None,
        hx_put=None, hx_delete=None, hx_patch=None, hx_trigger=None,
        hx_target=None, hx_swap=None, hx_swap_oob=None, hx_include=None,
        hx_select=None, hx_select_oob=None, hx_indicator=None,
        hx_push_url=None, hx_confirm=None, hx_disable=None,
        hx_replace_url=None, hx_disabled_elt=None, hx_ext=None,
        hx_headers=None, hx_history=None, hx_history_elt=None,
        hx_inherit=None, hx_params=None, hx_preserve=None,
        hx_prompt=None, hx_request=None, hx_sync=None, hx_validate=None)
```

Create a standard `FT` element with some SVG-specific attrs

Basic shapes

We'll define a simple function to display SVG shapes in this notebook:

```
def demo(el, h=50, w=50): return show(Svg(h=h,w=w)(el))
```

Rect

source

```
Rect (width, height, x=0, y=0, fill=None, stroke=None, stroke_width=None,
      rx=None, ry=None, transform=None, opacity=None, clip=None,
      mask=None, filter=None, vector_effect=None, pointer_events=None,
      target_id=None, hx_vals=None, id=None, cls=None, title=None,
      style=None, accesskey=None, contenteditable=None, dir=None,
      draggable=None, enterkeyhint=None, hidden=None, inert=None,
      inputmode=None, lang=None, popover=None, spellcheck=None,
      tabindex=None, translate=None, hx_get=None, hx_post=None,
      hx_put=None, hx_delete=None, hx_patch=None, hx_trigger=None,
      hx_target=None, hx_swap=None, hx_swap_oob=None, hx_include=None,
      hx_select=None, hx_select_oob=None, hx_indicator=None,
      hx_push_url=None, hx_confirm=None, hx_disable=None,
      hx_replace_url=None, hx_disabled_elt=None, hx_ext=None,
      hx_headers=None, hx_history=None, hx_history_elt=None,
      hx_inherit=None, hx_params=None, hx_preserve=None, hx_prompt=None,
      hx_request=None, hx_sync=None, hx_validate=None)
```

A standard SVG `rect` element

All our shapes just create regular `FT` elements. The only extra functionality provided by most of them is to add additional defined kwargs to improve auto-complete in IDEs and notebooks, and re-order parameters so that positional args can also be used to save a bit of typing, e.g:

```
demo(Rect(30, 30, fill='blue', rx=8, ry=8))
```

Circle

source

```
Circle (r, cx=0, cy=0, fill=None, stroke=None, stroke_width=None,
        transform=None, opacity=None, clip=None, mask=None, filter=None,
        vector_effect=None, pointer_events=None, target_id=None,
        hx_vals=None, id=None, cls=None, title=None, style=None,
        accesskey=None, contenteditable=None, dir=None, draggable=None,
        enterkeyhint=None, hidden=None, inert=None, inputmode=None,
        lang=None, popover=None, spellcheck=None, tabindex=None,
        translate=None, hx_get=None, hx_post=None, hx_put=None,
        hx_delete=None, hx_patch=None, hx_trigger=None, hx_target=None,
        hx_swap=None, hx_swap_oob=None, hx_include=None, hx_select=None,
        hx_select_oob=None, hx_indicator=None, hx_push_url=None,
        hx_confirm=None, hx_disable=None, hx_replace_url=None,
        hx_disabled_elt=None, hx_ext=None, hx_headers=None,
        hx_history=None, hx_history_elt=None, hx_inherit=None,
        hx_params=None, hx_preserve=None, hx_prompt=None,
        hx_request=None, hx_sync=None, hx_validate=None)
```

A standard SVG `circle` element

```
demo(Circle(20, 25, 25, stroke='red', stroke_width=3))
```

Ellipse

source

```
Ellipse (rx, ry, cx=0, cy=0, fill=None, stroke=None, stroke_width=None,
         transform=None, opacity=None, clip=None, mask=None, filter=None,
         vector_effect=None, pointer_events=None, target_id=None,
         hx_vals=None, id=None, cls=None, title=None, style=None,
         accesskey=None, contenteditable=None, dir=None, draggable=None,
         enterkeyhint=None, hidden=None, inert=None, inputmode=None,
         lang=None, popover=None, spellcheck=None, tabindex=None,
         translate=None, hx_get=None, hx_post=None, hx_put=None,
         hx_delete=None, hx_patch=None, hx_trigger=None, hx_target=None,
         hx_swap=None, hx_swap_oob=None, hx_include=None, hx_select=None,
         hx_select_oob=None, hx_indicator=None, hx_push_url=None,
         hx_confirm=None, hx_disable=None, hx_replace_url=None,
         hx_disabled_elt=None, hx_ext=None, hx_headers=None,
         hx_history=None, hx_history_elt=None, hx_inherit=None,
         hx_params=None, hx_preserve=None, hx_prompt=None,
         hx_request=None, hx_sync=None, hx_validate=None)
```

A standard SVG `ellipse` element

```
demo(Ellipse(20, 10, 25, 25))
```

---

transformd

source

```
transformd (translate=None, scale=None, rotate=None, skewX=None,
            skewY=None, matrix=None)
```

Create an SVG `transform` kwarg dict

```
rot = transformd(rotate=(45, 25, 25))
rot
```

```
{'transform': 'rotate(45,25,25)'}
```

```
demo(Ellipse(20, 10, 25, 25, **rot))
```

Line

source

```
Line (x1, y1, x2=0, y2=0, stroke='black', w=None, stroke_width=1,
      transform=None, opacity=None, clip=None, mask=None, filter=None,
      vector_effect=None, pointer_events=None, target_id=None,
      hx_vals=None, id=None, cls=None, title=None, style=None,
      accesskey=None, contenteditable=None, dir=None, draggable=None,
      enterkeyhint=None, hidden=None, inert=None, inputmode=None,
      lang=None, popover=None, spellcheck=None, tabindex=None,
      translate=None, hx_get=None, hx_post=None, hx_put=None,
      hx_delete=None, hx_patch=None, hx_trigger=None, hx_target=None,
      hx_swap=None, hx_swap_oob=None, hx_include=None, hx_select=None,
      hx_select_oob=None, hx_indicator=None, hx_push_url=None,
      hx_confirm=None, hx_disable=None, hx_replace_url=None,
      hx_disabled_elt=None, hx_ext=None, hx_headers=None,
      hx_history=None, hx_history_elt=None, hx_inherit=None,
      hx_params=None, hx_preserve=None, hx_prompt=None, hx_request=None,
      hx_sync=None, hx_validate=None)
```

A standard SVG `line` element

```
demo(Line(20, 30, w=3))
```

Polyline

source

```
Polyline (*args, points=None, fill=None, stroke=None, stroke_width=None,
          transform=None, opacity=None, clip=None, mask=None,
          filter=None, vector_effect=None, pointer_events=None,
          target_id=None, hx_vals=None, id=None, cls=None, title=None,
          style=None, accesskey=None, contenteditable=None, dir=None,
          draggable=None, enterkeyhint=None, hidden=None, inert=None,
          inputmode=None, lang=None, popover=None, spellcheck=None,
          tabindex=None, translate=None, hx_get=None, hx_post=None,
          hx_put=None, hx_delete=None, hx_patch=None, hx_trigger=None,
          hx_target=None, hx_swap=None, hx_swap_oob=None,
          hx_include=None, hx_select=None, hx_select_oob=None,
          hx_indicator=None, hx_push_url=None, hx_confirm=None,
          hx_disable=None, hx_replace_url=None, hx_disabled_elt=None,
          hx_ext=None, hx_headers=None, hx_history=None,
          hx_history_elt=None, hx_inherit=None, hx_params=None,
          hx_preserve=None, hx_prompt=None, hx_request=None,
          hx_sync=None, hx_validate=None)
```

A standard SVG `polyline` element

```
demo(Polyline((0,0), (10,10), (20,0), (30,10), (40,0),
              fill='yellow', stroke='blue', stroke_width=2))
```

```
demo(Polyline(points='0,0 10,10 20,0 30,10 40,0', fill='purple', stroke_width=2))
```

Polygon

source

```
Polygon (*args, points=None, fill=None, stroke=None, stroke_width=None,
         transform=None, opacity=None, clip=None, mask=None, filter=None,
         vector_effect=None, pointer_events=None, target_id=None,
         hx_vals=None, id=None, cls=None, title=None, style=None,
         accesskey=None, contenteditable=None, dir=None, draggable=None,
         enterkeyhint=None, hidden=None, inert=None, inputmode=None,
         lang=None, popover=None, spellcheck=None, tabindex=None,
         translate=None, hx_get=None, hx_post=None, hx_put=None,
         hx_delete=None, hx_patch=None, hx_trigger=None, hx_target=None,
         hx_swap=None, hx_swap_oob=None, hx_include=None, hx_select=None,
         hx_select_oob=None, hx_indicator=None, hx_push_url=None,
         hx_confirm=None, hx_disable=None, hx_replace_url=None,
         hx_disabled_elt=None, hx_ext=None, hx_headers=None,
         hx_history=None, hx_history_elt=None, hx_inherit=None,
         hx_params=None, hx_preserve=None, hx_prompt=None,
         hx_request=None, hx_sync=None, hx_validate=None)
```

A standard SVG `polygon` element

```
demo(Polygon((25,5), (43.3,15), (43.3,35), (25,45), (6.7,35), (6.7,15),
             fill='lightblue', stroke='navy', stroke_width=2))
```

```
demo(Polygon(points='25,5 43.3,15 43.3,35 25,45 6.7,35 6.7,15',
             fill='lightgreen', stroke='darkgreen', stroke_width=2))
```

Text

source

```
Text (*args, x=0, y=0, font_family=None, font_size=None, fill=None,
      text_anchor=None, dominant_baseline=None, font_weight=None,
      font_style=None, text_decoration=None, transform=None,
      opacity=None, clip=None, mask=None, filter=None,
      vector_effect=None, pointer_events=None, target_id=None,
      hx_vals=None, id=None, cls=None, title=None, style=None,
      accesskey=None, contenteditable=None, dir=None, draggable=None,
      enterkeyhint=None, hidden=None, inert=None, inputmode=None,
      lang=None, popover=None, spellcheck=None, tabindex=None,
      translate=None, hx_get=None, hx_post=None, hx_put=None,
      hx_delete=None, hx_patch=None, hx_trigger=None, hx_target=None,
      hx_swap=None, hx_swap_oob=None, hx_include=None, hx_select=None,
      hx_select_oob=None, hx_indicator=None, hx_push_url=None,
      hx_confirm=None, hx_disable=None, hx_replace_url=None,
      hx_disabled_elt=None, hx_ext=None, hx_headers=None,
      hx_history=None, hx_history_elt=None, hx_inherit=None,
      hx_params=None, hx_preserve=None, hx_prompt=None, hx_request=None,
      hx_sync=None, hx_validate=None)
```

A standard SVG `text` element

```
demo(Text("Hello!", x=10, y=30))
```

Hello!

Paths

Paths in SVGs are more complex, so we add a small (optional) fluent interface for constructing them:

PathFT

source

```
PathFT (tag:str, cs:tuple, attrs:dict=None, void_=False, **kwargs)
```

A 'Fast Tag' structure, containing `tag`, `children`, and `attrs`

Path

source

```
Path (d='', fill=None, stroke=None, stroke_width=None, transform=None,
      opacity=None, clip=None, mask=None, filter=None,
      vector_effect=None, pointer_events=None, target_id=None,
      hx_vals=None, id=None, cls=None, title=None, style=None,
      accesskey=None, contenteditable=None, dir=None, draggable=None,
      enterkeyhint=None, hidden=None, inert=None, inputmode=None,
      lang=None, popover=None, spellcheck=None, tabindex=None,
      translate=None, hx_get=None, hx_post=None, hx_put=None,
      hx_delete=None, hx_patch=None, hx_trigger=None, hx_target=None,
      hx_swap=None, hx_swap_oob=None, hx_include=None, hx_select=None,
      hx_select_oob=None, hx_indicator=None, hx_push_url=None,
      hx_confirm=None, hx_disable=None, hx_replace_url=None,
      hx_disabled_elt=None, hx_ext=None, hx_headers=None,
      hx_history=None, hx_history_elt=None, hx_inherit=None,
      hx_params=None, hx_preserve=None, hx_prompt=None, hx_request=None,
      hx_sync=None, hx_validate=None)
```

Create a standard `path` SVG element. This is a special object

Let's create a square shape, but using `Path` instead of `Rect`:

- M(10, 10): Move to starting point (10, 10)
- L(40, 10): Line to (40, 10) - top edge
- L(40, 40): Line to (40, 40) - right edge
- L(10, 40): Line to (10, 40) - bottom edge
- Z(): Close path - connects back to start

M = Move to, L = Line to, Z = Close path

```
demo(Path(fill='none', stroke='purple', stroke_width=2
         ).M(10, 10).L(40, 10).L(40, 40).L(10, 40).Z())
```

Using curves we can create a spiral:

```
p = (Path(fill='none', stroke='purple', stroke_width=2)
     .M(25, 25)
     .C(25, 25, 20, 20, 30, 20)
     .C(40, 20, 40, 30, 30, 30)
     .C(20, 30, 20, 15, 35, 15)
     .C(50, 15, 50, 35, 25, 35)
     .C(0, 35, 0, 10, 40, 10)
     .C(80, 10, 80, 40, 25, 40))
demo(p, 50, 100)
```

Using arcs and curves we can create a map marker icon:

```
p = (Path(fill='red')
     .M(25,45)
     .C(25,45,10,35,10,25)
     .A(15,15,0,1,1,40,25)
     .C(40,35,25,45,25,45)
     .Z())
demo(p)
```

Behind the scenes it's just creating regular SVG path `d` attr – you can pass `d` in directly if you prefer.

```
print(p.d)
```

```
M25 45 C25 45 10 35 10 25 A15 15 0 1 1 40 25 C40 35 25 45 25 45 Z
```

```
demo(Path(d='M25 45 C25 45 10 35 10 25 A15 15 0 1 1 40 25 C40 35 25 45 25 45 Z'))
```

PathFT.M

source

```
PathFT.M (x, y)
```

Move to.

PathFT.L

source

```
PathFT.L (x, y)
```

Line to.

PathFT.H

source

```
PathFT.H (x)
```

Horizontal line to.

PathFT.V

source

```
PathFT.V (y)
```

Vertical line to.

PathFT.Z

source

```
PathFT.Z ()
```

Close path.

PathFT.C

source

```
PathFT.C (x1, y1, x2, y2, x, y)
```

Cubic Bézier curve.

PathFT.S

source

```
PathFT.S (x2, y2, x, y)
```

Smooth cubic Bézier curve.

PathFT.Q

source

```
PathFT.Q (x1, y1, x, y)
```

Quadratic Bézier curve.

PathFT.T

source

```
PathFT.T (x, y)
```

Smooth quadratic Bézier curve.

PathFT.A

source

```
PathFT.A (rx, ry, x_axis_rotation, large_arc_flag, sweep_flag, x, y)
```

Elliptical Arc.

HTMX helpers

SvgOob

source

```
SvgOob (*args, **kwargs)
```

Wraps an SVG shape as required for an HTMX OOB swap

When returning an SVG shape out-of-band (OOB) in HTMX, you need to wrap it with `SvgOob` to have it appear correctly. (`SvgOob` is just a shortcut for `Template(Svg(...))`, which is the trick that makes SVG OOB swaps work.)

SvgInb

source

```
SvgInb (*args, **kwargs)
```

Wraps an SVG shape as required for an HTMX inband swap

When returning an SVG shape in-band in HTMX, either have the calling element include `hx_select='svg>*'`, or `**svg_inb` (which are two ways of saying the same thing), or wrap the response with `SvgInb` to have it appear correctly. (`SvgInb` is just a shortcut for the tuple `(Svg(...), HtmxResponseHeaders(hx_reselect='svg>*'))`, which is the trick that makes SVG in-band swaps work.)

Report an issue