Source > Components

# Components

ft_html and ft_hx functions to add some conveniences to ft , along with a full set of basic HTML components, and functions to work with forms and FT conversion

```
from lxml import html as lx
from pprint import pprint
```

## show

source

```
show (ft, *rest)
```

*Renders FT Components into HTML within a Jupyter notebook.*

```
sentence = P(Strong("FastHTML is ", I("Fast")))

# When placed within the `show()` function, this will render
# the HTML in Jupyter notebooks.
show(sentence)
```

**FastHTML is *Fast***

```
# Called without the `show()` function, the raw HTML is displayed
sentence
```

```
<p>
<strong>FastHTML is <i>Fast</i></strong></p>
```

## attrmap_x

source

```
attrmap_x (o)
```

## ft_html

source

```
ft_html (tag:str, *c, id=None, cls=None, title=None, style=None,
         attrmap=None, valmap=None, ft_cls=None, auto_id=None, **kwargs)
```

## ft_hx

source

```
ft_hx (tag:str, *c, 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, **kwargs)
```

```
ft_html('a', _at_click_dot_away=1)
```

```
<a @click_dot_away="1"></a>
```

```
ft_html('a', **{'@click.away':1})
```

```
<a @click.away="1"></a>
```

```
ft_html('a', {'@click.away':1})
```

```
<a @click.away="1"></a>
```

```
ft_hx('a', hx_vals={'a':1})
```

```
<a hx-vals='{"a": 1}'></a>
```

## File

source

```
File (fname)
```

*Use the unescaped text in file `fname` directly*

For tags that have a `name` attribute, it will be set to the value of `id` if not provided explicitly:

```
Form(Button(target_id='foo', id='btn'),
     hx_post='/', target_id='tgt', id='frm')
```

```
<form hx-post="/" hx-target="#tgt" id="frm" name="frm"><button hx-target="#foo" id="b
```

## fill_form

source

```
fill_form (form:fastcore.xml.FT, obj)
```

*Fills named items in `form` using attributes in `obj`*

```
@dataclass
class TodoItem:
    title:str; id:int; done:bool; details:str; opt:str='a'

todo = TodoItem(id=2, title="Profit", done=True, details="Details", opt='b')
check = Label(Input(type="checkbox", cls="checkboxer", name="done", data_foo="bar"), "
form = Form(Fieldset(Input(cls="char", id="title", value="a"), check, Input(type="hidd
                     Select(Option(value='a'), Option(value='b'), name='opt'),
                     Textarea(id='details'), Button("Save"),
                     name="stuff"))
form = fill_form(form, todo)
assert '<textarea id="details" name="details">Details</textarea>' in to_xml(form)
form
```

```
<form><fieldset name="stuff">    <input value="Profit" id="title" class="char" name='
<label class="px-2">      <input type="checkbox" name="done" data-foo="bar" class="ch
Done</label>    <input type="hidden" id="id" name="id" value="2">
<select name="opt"><option value="a"></option><option value="b" selected="1"></optio
```

## fill_dataclass

source

```
fill_dataclass (src, dest)
```

*Modifies dataclass in-place and returns it*

```
nt = TodoItem('', 0, False, '')
fill_dataclass(todo, nt)
nt
```

```
TodoItem(title='Profit', id=2, done=True, details='Details', opt='b')
```

## find_inputs

source

```
find_inputs (e, tags='input', **kw)
```

*Recursively find all elements in `e` with `tags` and attrs matching `kw`*

```
inps = find_inputs(form, id='title')
test_eq(len(inps), 1)
inps
```

```
[input((),{'value': 'Profit', 'id': 'title', 'class': 'char', 'name': 'title'})]
```

You can also use lxml for more sophisticated searching:

```
elem = lx.fromstring(to_xml(form))
test_eq(elem.xpath("//input[@id='title']/@value"), ['Profit'])
```

## getattr

source

```
__getattr__ (tag)
```

## html2ft

source

```
html2ft (html, attr1st=False)
```

*Convert HTML to an `ft` expression*

```
h = to_xml(form)
hl_md(html2ft(h), 'python')
```

```
Form(
    Fieldset(
        Input(value='Profit', id='title', name='title', cls='char'),
        Label(
            Input(type='checkbox', name='done', data_foo='bar', checked='1', cls='che
            'Done',
            cls='px-2'
        ),
        Input(type='hidden', id='id', name='id', value='2'),
        Select(
            Option(value='a'),
            Option(value='b', selected='1'),
            name='opt'
        ),
        Textarea('Details', id='details', name='details'),
        Button('Save'),
        name='stuff'
    )
)
```

```
hl_md(html2ft(h, attr1st=True), 'python')
```

```
Form(
    Fieldset(name='stuff')(
        Input(value='Profit', id='title', name='title', cls='char'),
        Label(cls='px-2')(
            Input(type='checkbox', name='done', data_foo='bar', checked='1', cls='che
            'Done'
        ),
        Input(type='hidden', id='id', name='id', value='2'),
        Select(name='opt')(
            Option(value='a'),
            Option(value='b', selected='1')
        ),
        Textarea('Details', id='details', name='details'),
        Button('Save')
    )
)
```

---

## sse_message

source

> sse_message (elm, event='message')

*Convert element `elm` into a format suitable for SSE streaming*

```
print(sse_message(Div(P('hi'), P('there'))))
```

```
event: message
data: <div>
data:   <p>hi</p>
data:   <p>there</p>
data: </div>
```

Report an issue