



 Source > Component extensions

## Component extensions

```
from pprint import pprint
```

### A

[source](#)

```
A (*c, hx_get=None, target_id=None, hx_swap=None, href='#', 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_post=None,
    hx_put=None, hx_delete=None, hx_patch=None, hx_trigger=None,
    hx_target=None, hx_include=None, hx_select=None, hx_indicator=None,
    hx_push_url=None, hx_confirm=None, hx_disable=None,
    hx_replace_url=None, hx_on=None, **kwargs)
```

An A tag: *href* defaults to '#' for more concise use with HTMX

```
A('text', ht_get='/get', target_id='id')
<a href="#" ht-get="/get" hx-target="#id">text</a>
```

### Form

[source](#)

```
Form (*c, enctype='multipart/form-data', 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_include=None,
      hx_select=None, hx_indicator=None, hx_push_url=None,
      hx_confirm=None, hx_disable=None, hx_replace_url=None, hx_on=None,
      **kwargs)
```

A Form tag: identical to plain [ft\\_hx](#) version except default *enctype* = 'multipart/form-data'

## AX

```
AX (txt, hx_get=None, target_id=None, hx_swap=None, href='#',
    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_post=None, hx_put=None, hx_delete=None,
    hx_patch=None, hx_trigger=None, hx_target=None, hx_include=None,
    hx_select=None, hx_indicator=None, hx_push_url=None, hx_confirm=None,
    hx_disable=None, hx_replace_url=None, hx_on=None, **kwargs)
```

*An A tag with just one text child, allowing `hx_get`, `target_id`, and `hx_swap` to be positional params*

```
AX('text', '/get', 'id')
```

```
<a href="#" hx-get="/get" hx-target="#id">text</a>
```

## Hidden

```
Hidden (value:Any='', id:Any=None, target_id=None, hx_vals=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_include=None, hx_select=None, hx_indicator=None,
        hx_push_url=None, hx_confirm=None, hx_disable=None,
        hx_replace_url=None, hx_on=None, **kwargs)
```

*An Input of type ‘hidden’*

## CheckboxX

```
CheckboxX (checked:bool=False, label=None, value='1', id=None, name=None,
            target_id=None, hx_vals=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_include=None, hx_select=None,
```

```
hx_indicator=None, hx_push_url=None, hx_confirm=None,
hx_disable=None, hx_replace_url=None, hx_on=None, **kwargs)
```

A *Checkbox* optionally inside a *Label*, preceded by a [Hidden](#) with matching name

```
show(CheckboxX(True, 'Check me out!'))
```

☒ Check me out!

## Script

[source](#)

```
Script (code:str='', id=None, cls=None, title=None, style=None,
      attrmap=None, valmap=None, **kwargs)
```

A *Script* tag that doesn't escape its code

## Style

[source](#)

```
Style (*c, id=None, cls=None, title=None, style=None, attrmap=None,
      valmap=None, **kwargs)
```

A *Style* tag that doesn't escape its code

## double\_braces

[source](#)

```
double_braces (s)
```

Convert single braces to double braces if next to special chars or newline

## undouble\_braces

[source](#)

```
undouble_braces (s)
```

Convert double braces to single braces if next to special chars or newline

## loose\_format

[source](#)

```
loose_format (s, **kw)
```

String format *s* using *kw*, without being strict about braces outside of template params

## ScriptX

[source](#)

```
ScriptX (fname, src=None, nomodule=None, type=None, _async=None,
        defer=None, charset=None, crossorigin=None, integrity=None,
        **kw)
```

A *script* element with contents read from *fname*

---

## replace\_css\_vars

[source](#)

```
replace_css_vars (css, pre='tpl', **kwargs)
```

Replace *var(--)* CSS variables with *kwargs* if name prefix matches *pre*

---

## StyleX

[source](#)

```
StyleX (fname, **kw)
```

A *style* element with contents read from *fname* and variables replaced from *kw*

---

## On

[source](#)

```
On (code:str, event:str='click', sel:str='', me=True)
```

An *async surreal.js* script block event handler for *event* on selector *sel*

---

## Any

[source](#)

```
Any (sel:str, code:str, event:str='click')
```

An *any* *async surreal.js* script block event handler for *event* on selector *sel*

---

## Prev

[source](#)

```
Prev (code:str, event:str='click')
```

An *async surreal.js* script block event handler for *event* on previous sibling

---

## Now

[source](#)

```
Now (code:str, sel:str='')
```

An *async surreal.js* script block on selector *me(sel)*

---

## AnyNow

[source](#)

```
AnyNow (sel:str, code:str)
```

An *async surreal.js* script block on selector *any(sel)*

---

## run\_js

[source](#)

```
run_js (js, id=None, **kw)
```

Run *js* script, auto-generating *id* based on name of caller if needed, and *js*-escaping any *kw* params

---

## Titled

[source](#)

```
Titled (title:str='FastHTML app', *args, cls='container', target_id=None,
        hx_vals=None, id=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_include=None, hx_select=None, hx_indicator=None,
        hx_push_url=None, hx_confirm=None, hx_disable=None,
        hx_replace_url=None, hx_on=None, **kwargs)
```

An *HTML* partial containing a *Title*, and *H1*, and any provided children

---

## Socials

[source](#)

```
Socials (title, site_name, description, image, url=None, w=1200, h=630,
         twitter_site=None, creator=None, card='summary')
```

*OG* and *Twitter* social card headers

---

## Favicon

[source](#)

```
Favicon (light_icon, dark_icon)
```

## Light and dark favicon headers

---

### jsd

[source](#)

```
jsd (org, repo, root, path, prov='gh', typ='script', ver=None, esm=False,  
    **kwargs)
```

*jsdelivr* [Script](#) or CSS [Link](#) tag, or URL

---

### clear

[source](#)

```
clear (id)
```

 [Report an issue](#)