

App Setup

```
from fasthtml.common import *
app, rt = fast_app()

@rt("/")
def get():
    return Titled("Home",
        P("Welcome!"))

serve()
Run with python app.py, open http://localhost:5001.
```

Common FastTags (HTML Elements)

FastTag	HTML	Purpose
H1("...")	<h1>	Main heading
H2("...")	<h2>	Subheading
P("...")	<p>	Paragraph
A("...", href="/")	<a>	Link
Div(...)	<div>	Container
Ul(Li(...)), Ol(...)	/	Lists
Table(Tr(Td(...)))	<table>	Table
Br()	 	Line break
Hr()	<hr>	Horizontal line

Nest elements by passing them as arguments:
Div(H1("Title"), P("Text"))
Add CSS classes: P("text", cls="highlight")
Add IDs: Div("...", id="main")

Routes and Pages

Each route maps a URL to a Python function.

```
@rt("/") # home page
def get():
    return Titled("Home", P("Hi"))

@rt("/about") # about page
def get():
    return Titled("About", P("Info"))

@rt("/panel/{panel_id}") # URL param
def get(panel_id: int):
    return P(f"Panel #{panel_id}")
```

Forms

```
Form(
    Label("Bill ($):"), Br(),
    Input(type="number", name="bill",
        required=True),
    Br(), Br(),
    Label("State:"), Br(),
    Select(
        Option("Victoria", value="VIC"),
        Option("NSW", value="NSW"),
        name="state"
    ),
    Br(), Br(),
    Button("Calculate", type="submit"),
    method="post"
)
```

Attribute	Purpose
name	Identifies field in handler
type	"text", "number", "email"
required	Must be filled before submit
placeholder	Hint text shown in empty field
method	"post" for form submission

Handling Form Submissions

```
@rt("/")
def post(bill: float, state: str):
    savings = bill * 4 * 0.75
    return Titled("Results",
        P(f"Annual savings: ${savings:,.0f}"),
        A("Back", href="/"))
```

Parameter names **must match** form field **name** attributes. FastHTML auto-converts types.

Dynamic Content

```
Generate HTML from data:
rows = [Tr(Td(p["brand"]), Td(p["watts"]))
        for p in panels]
```

```
return Table(
    Tr(Th("Brand"), Th("Watts")),
    *rows) # unpack list with *
```

```
Conditional content:
if payback < 5:
    return P("Great!", cls="success")
else:
    return P("Consider options", cls="warn")
```

Common Errors

Missing **method="post"** on form – submits as GET, handler not called.
Parameter name mismatch – form name="bill" must match def post(bill: ...).
Forgetting **serve()** – app won't start without it.