In [ ]:
```
import marimo as mo
```

Batch and Form

Make custom UI elements using `batch()`, and turn any UI element into a form with `form()`.

In [ ]:
```
reset

variables = (
    mo.md(
        """
        Choose your variable values

        {x}

        {y}
        """
    )
    .batch(
        x=mo.ui.slider(start=1, stop=10, step=1, label="$x =$"),
        y=mo.ui.slider(start=1, stop=10, step=1, label="$y =$"),
    )
    .form(show_clear_button=True, bordered=False)
)

variables
```

<marimo-form data-initial-value='null' data-label='null' data-element-id='"bkHC-7"' data-loading='false' data-bordered='false' data-submit-button-label='"Submit"' data-submit-button-disabled='false' data-clear-on-submit='false' data-show-clear-button='true' data-clear-button-label='"Clear"' data-should-validate='false'><marimo-ui-element object-id='bkHC-7' random-id='44434ff4-ac87-b6c1-892f-7db94252dcc7'><marimo-dict data-initial-value='{"x":1,"y":1}' data-label='null' data-element-ids='{"bkHC-5":"x","bkHC-6":"y"}'>Choose your variable values <marimo-ui-element object-id='bkHC-5' random-id='d8908a25-52e7-99bc-ecbb-3e48d11b9b30'><marimo-slider data-initial-value='1' data-label='"<marimo-tex class=\"arithmatex\">||(x =||)</marimo-tex>"' data-start='1' data-stop='10' data-step='1' data-steps='[]' data-debounce='false' data-disabled='false' data-orientation='"horizontal"' data-show-value='false' data-include-input='false' data-full-width='false'></marimo-slider></marimo-ui-element> <marimo-ui-element object-id='bkHC-6' random-id='2842cd90-dd9b-089b-7b8c-a5ba90447c9d'><marimo-slider data-initial-value='1' data-label='"<marimo-tex class=\"arithmatex\">||(y =||)</marimo-tex>"' data-start='1' data-stop='10' data-step='1' data-steps='[]' data-debounce='false' data-disabled='false' data-orientation='"horizontal"' data-show-value='false' data-include-input='false' data-full-width='false'></marimo-slider></marimo-ui-element></marimo-dict></marimo-ui-element></marimo-form>

In [ ]:
```python
if variables.value is not None:
    submitted_values["x"].add(variables.value["x"])
    submitted_values["y"].add(variables.value["y"])

x = variables.value["x"] if variables.value else r"\ldots"
y = variables.value["y"] if variables.value else r"\ldots"


mo.md(
    f"""
    At the moment,
    $x = {x}$ and $y = {y}$

    All values ever assumed by $x$ and $y$ are
```

```
    {mo.hstack([mo.tree(submitted_values), reset], align="center", gap=4)}
    """
).callout()
```

In [ ]:
```
reset

submitted_values = {"x": set(), "y": set()}
```

In [ ]:
```
reset = mo.ui.button(label="reset history")
```