

New to Gradio? Start here: **Getting Started**

See the **Release History**

← Column

Group →

Tab

with gradio.Tab():

Description

Tab (or its alias TabItem) is a layout element. Components defined within the Tab will be visible when this tab is selected tab.

Example Usage

```
with gr.Blocks() as demo:
    with gr.Tab("Lion"):
        gr.Image("lion.jpg")
        gr.Button("New Lion")
    with gr.Tab("Tiger"):
        gr.Image("tiger.jpg")
        gr.Button("New Tiger")
```

Initialization

Parameter	Description
<div>label</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	The visual label for the tab
<div>visible</div> <div><i>bool</i></div> <div>default: True</div>	If False, Tab will be hidden.

meter	Description
<div>interactive</div> <div><i>bool</i></div> <div>default: True</div>	If False, Tab will not be clickable.
<div>id</div> <div><i>int str None</i></div> <div>default: None</div>	An optional identifier for the tab, required if you wish to control the selected tab from a predict function.
<div>elem_id</div> <div><i>str None</i></div> <div>default: None</div>	An optional string that is assigned as the id of the <div> containing the contents of the Tab layout. The same string followed by "-button" is attached to the Tab button. Can be used for targeting CSS styles.
<div>elem_classes</div> <div><i>list[str] str None</i></div> <div>default: None</div>	An optional string or list of strings that are assigned as the class of this component in the HTML DOM. Can be used for targeting CSS styles.
<div>render</div> <div><i>bool</i></div> <div>default: True</div>	If False, this layout will not be rendered in the Blocks context. Should be used if the intention is to assign event listeners now but render the component later.

Methods

select

```
gradio.Tab.select(...)
```

Description

Event listener for when the user selects or deselects the Tab. Uses event data `gradio.SelectData` to carry `value` referring to the label of the Tab, and `selected` to refer to state of the Tab. See `EventData` documentation on how to use this event data

Agruments



Parameter	Description
<div>fn</div> <div><i>Callable</i> <i>None</i> <i>Literal['decorator']</i></div> <div>default: "decorator"</div>	the function to call when this event is triggered. Often a machine learning model's prediction function. Each parameter of the function corresponds to one input component, and the function should return a single value or a tuple of values, with each element in the tuple corresponding to one output component.
<div>inputs</div> <div><i>Component</i> <i>list[Component]</i> <i>set[Component]</i> <i>None</i></div> <div>default: None</div>	List of gradio.components to use as inputs. If the function takes no inputs, this should be an empty list.
<div>outputs</div> <div><i>Component</i> <i>list[Component]</i> <i>None</i></div> <div>default: None</div>	List of gradio.components to use as outputs. If the function returns no outputs, this should be an empty list.
<div>api_name</div> <div><i>str</i> <i>None</i> <i>Literal[False]</i></div> <div>default: None</div>	defines how the endpoint appears in the API docs. Can be a string, None, or False. If set to a string, the endpoint will be exposed in the API docs with the given name. If None (default), the name of the function will be used as the API endpoint. If False, the endpoint will not be exposed in the API docs and downstream apps (including those that <code>gr.load</code> this app) will not be able to use this event.
<div>scroll_to_output</div> <div><i>bool</i></div> <div>default: False</div>	If True, will scroll to output component on completion
<div>show_progress</div> <div><i>Literal[('full', 'minimal', 'hidden')]</i></div> <div>default: "full"</div>	If True, will show progress animation while pending



Parameter	Description
<div>queue</div> <div><i>bool None</i></div> <div>default: None</div>	If True, will place the request on the queue, if the queue has been enabled. If False, will not put this event on the queue, even if the queue has been enabled. If None, will use the queue setting of the gradio app.
<div>batch</div> <div><i>bool</i></div> <div>default: False</div>	If True, then the function should process a batch of inputs, meaning that it should accept a list of input values for each parameter. The lists should be of equal length (and be up to length <code>max_batch_size</code>). The function is then <i>required</i> to return a tuple of lists (even if there is only 1 output component), with each list in the tuple corresponding to one output component.
<div>max_batch_size</div> <div><i>int</i></div> <div>default: 4</div>	Maximum number of inputs to batch together if this is called from the queue (only relevant if batch=True)
<div>preprocess</div> <div><i>bool</i></div> <div>default: True</div>	If False, will not run preprocessing of component data before running 'fn' (e.g. leaving it as a base64 string if this method is called with the <code>Image</code> component).
<div>postprocess</div> <div><i>bool</i></div> <div>default: True</div>	If False, will not run postprocessing of component data before returning 'fn' output to the browser.
<div>cancels</div> <div><i>dict[str, Any] list[dict[str, Any]] None</i></div> <div>default: None</div>	A list of other events to cancel when this listener is triggered. For example, setting cancels=[click_event] will cancel the click_event, where click_event is the return value of another components .click method. Functions that have not yet run (or generators that are iterating) will be cancelled, but functions that are currently running will be allowed to finish.



Parameter	Description
<div>every</div> <div><i>float</i> <i>None</i></div> <div>default: None</div>	Run this event 'every' number of seconds while the client connection is open. Interpreted in seconds. Queue must be enabled.
<div>trigger_mode</div> <div><i>Literal</i>[(<i>'once'</i>, <i>'multiple'</i>, <i>'always_last'</i>)] <i>None</i></div> <div>default: None</div>	If "once" (default for all events except <code>.change()</code>) would not allow any submissions while an event is pending. If set to "multiple", unlimited submissions are allowed while pending, and "always_last" (default for <code>.change()</code> event) would allow a second submission after the pending event is complete.
<div>js</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	Optional frontend js method to run before running 'fn'. Input arguments for js method are values of 'inputs' and 'outputs', return should be a list of values for output components.
<div>concurrency_limit</div> <div><i>int</i> <i>None</i> <i>Literal</i>['default']</div> <div>default: "default"</div>	If set, this is the maximum number of this event that can be running simultaneously. Can be set to None to mean no concurrency_limit (any number of this event can be running simultaneously). Set to "default" to use the default concurrency limit (defined by the <code>default_concurrency_limit</code> parameter in <code>Blocks.queue()</code> , which itself is 1 by default).
<div>concurrency_id</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	If set, this is the id of the concurrency group. Events with the same concurrency_id will be limited by the lowest set concurrency_limit.
<div>show_api</div> <div><i>bool</i></div> <div>default: True</div>	whether to show this event in the "view API" page of the Gradio app, or in the ".view_api()" method of the Gradio clients. Unlike setting api_name to False, setting show_api to False will still allow downstream apps to use this event. If fn is None, show_api will automatically be set to False.



← Column

Group →

