Streamlit - Cration - Core - API

Function	What it does	Example
`st.title(text)`	Top-level title (large bold).	`st.title("My App")`
`st.header(text)`	Section header (medium).	`st.header("Section 1")`
`st.subheader(text)`	Sub-section header (small).	`st.subheader("Details")`
`st.markdown(md, unsafe_allow_html=False)`	Render Markdown (or raw HTML if flag is True).	`st.markdown("**Bold** _italic_")`
`st.write(*args, **kwargs)`	Generic writer – auto-detects data type.	`st.write("Number:", 42)`
`st.text(text)`	Monospace text block.	`st.text("Plain text")`
`st.code(code, language=None)`	Render a code block with optional syntax highlighting.	`st.code("print('hello')", language='python')`
`st.latex(latex)`	Render LaTeX math.	`st.latex(r'\frac{1}{2}')`
`st.image(image, caption=None, width=None)`	Show an image (PIL, numpy array, URL).	`st.image("https://example.com/img.png", width=300)`
`st.video(video, format=None)`	Embed a video (URL or local file).	`st.video("https://youtu.be/xyz")`
`st.audio(audio, format=None)`	Embed an audio clip.	`st.audio("song.mp3")`
`st.map(df, latitude="lat", longitude="lon")`	Quick scatter map using pydeck.	`st.map(df)`
`st.pyplot(fig, use_container_width=False)`	Render a Matplotlib figure.	`st.pyplot(my_fig)`
`st.table(df)`	Render a static table (pandas).	`st.table(df.head())`
`st.dataframe(df, height=None)`	Render an interactive dataframe (scrollable).	`st.dataframe(df)`
`st.json(data)`	Pretty-print JSON.	`st.json({"a":1, "b":[2,3]})`
`st.progress(value)`	Show a progress bar (0–1).	`for i in range(100): st.progress(i/100)`
`st.spinner(text="Loading…")`	Show a spinner while code runs.	`with st.spinner("Working…"): time.sleep(2)`
`st.expander(label, expanded=False)`	Collapsible section.	`with st.expander("Details"): st.write("More info")`
`st.sidebar`	Reference to the sidebar container. Use same widgets inside it.	`st.sidebar.title("Sidebar")`
`st.beta_container()` (now just `st.container()`)	Group widgets into a single block.	`with st.container(): st.write("Block")`
`st.columns([width1, width2])`	Return a list of column objects.	`col1, col2 = st.columns([2, 1]); col1.write("Left"); col2.write("Right")`
`st.empty()`	Placeholder that can be replaced later.	`placeholder = st.empty(); placeholder.text("Loading…"); placeholder.success("Done")`
`st.markdown("", unsafe_allow_html=True)`	Inject raw HTML (use with care).	`st.markdown("<h1 style='color:red'>Hi</h1>", unsafe_allow_html=True)`

Widgets – User Input

Widget	What it does	Example
`st.button(label, key=None)`	Clickable button. Returns True on click.	`if st.button("Run"): do_work()`
`st.download_button(label, data, file_name, mime)`	Trigger a download.	`st.download_button("Download CSV", df.to_csv(index=False), "data.csv")`
`st.checkbox(label, value=False, key=None)`	True/False toggle.	`if st.checkbox("Show plot"): plot()`
`st.radio(options, index=0, key=None)`	Single choice from a list.	`choice = st.radio("Pick one", ["A","B"])`
`st.selectbox(label, options, index=0, key=None)`	Dropdown menu.	`fruit = st.selectbox("Fruit", ["Apple","Banana"])`
`st.multiselect(label, options, default=None, key=None)`	Multiple selections.	`colors = st.multiselect("Colors", ["Red","Green"])`
`st.slider(label, min_value, max_value, value=None, step=1, key=None)`	Drag-slider.	`volume = st.slider("Volume", 0, 100, 50)`
`st.number_input(label, min_value=None, max_value=None, value=0, step=1, key=None)`	Numeric input.	`age = st.number_input("Age", min_value=0, max_value=120)`
`st.text_input(label, value="", key=None)`	Single line text.	`name = st.text_input("Name")`
`st.text_area(label, value="", height=None, key=None)`	Multi-line text.	`st.text_area("Comments")`
`st.date_input(label, value=None, min_value=None, max_value=None, key=None)`	Calendar picker.	`dob = st.date_input("DOB")`
`st.time_input(label, value=None, key=None)`	Time picker.	`start = st.time_input("Start time")`
`st.file_uploader(label, type=None, accept_multiple_files=False)`	Upload file(s).	`uploaded = st.file_uploader("Upload", type=["csv"])`
`st.color_picker(label, value="#000000")`	Pick a color.	`color = st.color_picker("Pick a color")`
`st.camera_input(label)`	Capture image from webcam (experimental).	`photo = st.camera_input("Take a photo")`
`st.audio(...)` (widget)	Upload or stream audio.	`audio_file = st.file_uploader("Audio", type=["mp3","wav"])`
`st.video(...)` (widget)	Upload or stream video.	`video_file = st.file_uploader("Video", type=["mp4"])`
`st.metric(label, value, delta=None)`	Show a KPI metric.	`st.metric("Temperature", "72 °F", delta="-2 °F")`

Layout & Styling

Function	What it does	Example
`st.set_page_config(page_title=None, page_icon=None, layout="centered", initial_sidebar_state="auto")`	Configure the app's appearance before any other calls.	`st.set_page_config(page_title="My App", layout="wide")`
`st.sidebar.title(...)`, `.write(...)`	Same as main, but in the sidebar.	`st.sidebar.header("Options")`
`st.columns([col1, col2])`	Return column objects for side-by-side layout.	`c1, c2 = st.columns([3, 1]); c1.write("Wide"); c2.write("Narrow")`
`st.container()`	Group widgets; can be nested.	`with st.container(): st.write("Block")`
`st.expander(label)`	Collapsible section.	`with st.expander("Details"): st.write("More")`
`st.beta_expander()` (now just expander)	Same as above.	`with st.expander("Help"): st.write("…")`
`st.empty()`	Placeholder for dynamic content.	`placeholder = st.empty(); placeholder.write("Loading…")`
`st.progress(value)`	Progress bar.	`st.progress(0.5)`
`st.spinner(text)`	Spinner during long operation.	`with st.spinner("Working…"): time.sleep(2)`
`st.toast(message, icon=None)` (since 1.18)	Small toast notification that disappears.	`st.toast("Saved!")`
`st.rerun()`	Programmatically rerun the app.	`if st.button("Rerun"): st.rerun()`

Session State – Persisting Data

Function	What it does	Example
`st.session_state`	Dictionary-like object that persists across reruns.	`if "counter" not in st.session_state: st.session_state.counter = 0`
`st.session_state["key"] = value`	Set a value.	`st.session_state.count += 1`
`st.session_state.get("key", default)`	Get with fallback.	`count = st.session_state.get("counter", 0)`
`st.experimental_set_query_params(**params)`	Update URL query string.	`st.experimental_set_query_params(page=2)`
`st.experimental_get_query_params()`	Read query string.	`params = st.experimental_get_query_params()`
`st.session_state.update(**kwargs)`	Bulk update.	`st.session_state.update(counter=0, mode="edit")`

Callbacks & Events

Function	What it does	Example
`st.button(..., on_click=callback, args=None, kwargs=None)`	Register a callback that runs when the button is clicked.	`def greet(): st.write("Hi!"); st.session_state.name = "Alice"; st.rerun(); st.button("Say hi", on_click=greet)`
`st.text_input(..., key=..., on_change=callback)`	Run callback when the text changes.	`def update(): st.session_state.name = st.session_state.my_name; st.rerun(); st.text_input("Name", key="my_name", on_change=update)`
`st.selectbox(..., key=..., index=0, on_change=callback)`	Same for select boxes.	`def choose(): st.session_state.choice = st.session_state.my_choice; st.rerun(); st.selectbox("Pick", ["A","B"], key="my_choice", on_change=choose)`
`st.slider(..., key=..., on_change=callback)`	Same for sliders.	`def adjust(): st.session_state.volume = st.session_state.my_vol; st.rerun(); st.slider("Vol", 0,100, key="my_vol", on_change=adjust)`

File I-O & Data

Function	What it does	Example
`st.file_uploader(...)`	Upload a file.	`uploaded = st.file_uploader("CSV", type="csv")`
`st.download_button(...)`	Download data.	`st.download_button("Download", df.to_csv(index=False), "data.csv")`
`pd.read_csv(file)`	Read CSV from uploaded file.	`if uploaded: df = pd.read_csv(uploaded); st.dataframe(df)`
`st.session_state["file"] = file`	Persist uploaded file for later use.	`if uploaded: st.session_state.file = u`

Plotting & Visualization

Function	What it does	Example
`st.pyplot(fig)`	Render Matplotlib figure.	`fig, ax = plt.subplots(); ax.plot([1,2]); st.pyplot(fig)`
`st.altair_chart(chart, use_container_width=True)`	Render Altair chart.	`chart = alt.Chart(df).mark_line().encode(x='x', y='y'); st.altair_chart(chart)`
`st.vega_lite_chart(spec, use_container_width=True)`	Render Vega-Lite JSON spec.	`spec = {"mark":"bar","encoding":{...}}; st.vega_lite_chart(spec)`
`st.plotly_chart(fig, use_container_width=True)`	Render Plotly figure.	`fig = px.line(df, x='x', y='y'); st.plotly_chart(fig)`
`st.bokeh_chart(fig, use_container_width=True)`	Render Bokeh figure.	`p = figure(); p.line(x, y); st.bokeh_chart(p)`

Media & Animation

Function	What it does	Example
`st.image(...)` (widget)	Upload or stream image.	`img = st.file_uploader("Image", type=["png","jpg"])`
`st.video(...)` (widget)	Upload or stream video.	`vid = st.file_uploader("Video", type=["mp4"])`
`st.audio(...)` (widget)	Upload or stream audio.	`aud = st.file_uploader("Audio", type=["mp3","wav"])`
`st.camera_input(label)`	Capture photo from webcam.	`photo = st.camera_input("Take a selfie")`
`st.lottie(...)` (via `streamlit_lottie`)	Render Lottie animations.	`st.lottie(url_or_dict)`

Other - Commands

Function	What it does	Example
`st.experimental_memo(func, ttl=None)`	Cache a function's return value (by default for the whole session).	`@st.experimental_memo; def load_data(): return pd.read_csv("big.csv")`
`st.experimental_singleton(func)`	Cache a singleton (one per app).	`@st.experimental_singleton; def get_db(): return sqlite3.connect("db.sqlite")`
`st.experimental_set_query_params(**params)`	Update URL query string (see above).	`st.experimental_set_query_params(page="2")`
`st.experimental_get_query_params()`	Read URL query string.	`params = st.experimental_get_query_params(); page = params.get("page", ["1"])[0]`
`st.set_option(key, value)`	Set Streamlit config options at runtime.	`st.set_option("deprecation.showPyplotGlobalUse", False)`
`st.toggle(key, value=False)`	*Deprecated* – use `checkbox`.	`if st.checkbox("Toggle"): ...`
`st.echo(code)`	Show the code that generated a widget.	`with st.echo(): st.button("Click")`
`st.experimental_show()`	Show the widget in a new window (debug).	`st.experimental_show("widget")`
`st.components.v1.html(html, height=400)`	Render arbitrary HTML/JS (e.g., Google Maps).	`st.components.v1.html("<iframe src='...'></iframe>")`