# **The API**

| [**cozmo**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#module-cozmo) |  |
| --- | --- |
| [**cozmo.action**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#module-cozmo.action) | Actions encapsulate specific high-level tasks that the Cozmo robot can perform. |
| [**cozmo.anim**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#module-cozmo.anim) | Animation related classes, functions, events and values. |
| [**cozmo.annotate**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#module-cozmo.annotate) | Camera image annotation. |
| [**cozmo.audio**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.audio.html#module-cozmo.audio) | Audio related classes, functions, events and values. |
| [**cozmo.behavior**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#module-cozmo.behavior) | Behaviors represent a task that Cozmo may perform for an indefinite amount of time. |
| [**cozmo.camera**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#module-cozmo.camera) | Support for Cozmo’s camera. |
| [**cozmo.conn**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#module-cozmo.conn) | Engine connection. |
| [**cozmo.event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#module-cozmo.event) | Event dispatch system. |
| [**cozmo.exceptions**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#module-cozmo.exceptions) | SDK-specific exception classes. |
| [**cozmo.faces**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#module-cozmo.faces) | Face recognition and enrollment. |
| [**cozmo.lights**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#module-cozmo.lights) | Helper routines for dealing with Cozmo’s lights and colors. |
| [**cozmo.nav\_memory\_map**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#module-cozmo.nav_memory_map) | A 2D navigation memory map of the world around Cozmo. |
| [**cozmo.objects**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#module-cozmo.objects) | Object and Power Cube recognition. |
| [**cozmo.oled\_face**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.oled_face.html#module-cozmo.oled_face) | Cozmo’s OLED screen that displays his face - related functions and values. |
| [**cozmo.opengl**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#module-cozmo.opengl) | This module provides a 3D visualizer for Cozmo’s world state. |
| [**cozmo.pets**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#module-cozmo.pets) | Pet detection. |
| [**cozmo.robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#module-cozmo.robot) | Classes and functions relating to an individual Cozmo robot. |
| [**cozmo.robot\_alignment**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot_alignment.html#module-cozmo.robot_alignment) | RobotAlignment related classes, functions, events and values. |
| [**cozmo.run**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#module-cozmo.run) | The run module contains helper classes and functions for opening a connection to the engine. |
| [**cozmo.song**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html#module-cozmo.song) | Song related classes, functions, events and values. |
| [**cozmo.tkview**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.tkview.html#module-cozmo.tkview) | This module provides a simple GUI viewer for Cozmo’s camera. |
| [**cozmo.util**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#module-cozmo.util) | Utility classes and functions. |
| [**cozmo.world**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#module-cozmo.world) | The “world” represents the robot’s known view of its environment. |

# **cozmo**

**Functions**

| **verify\_min\_clad\_version**() |  |
| --- | --- |

**cozmo.logger*= <Logger cozmo.general (WARNING)>***

The general purpose logger logs high level information about Cozmo events.

**cozmo.logger\_protocol*= <Logger cozmo.protocol (WARNING)>***

The protocol logger logs low level messages that are sent back and forth to Cozmo.

***class*cozmo.DeviceConnector(*cozmo\_port=5106*, *enable\_env\_vars=True*)**

Base class for objects that setup the physical connection to a device.

**connect(*loop*, *protocol\_factory*, *conn\_check*)**

Connect attempts to open a connection transport to the Cozmo app on a device.

On opening a transport it will create a protocol from the supplied factory and connect it to the transport, returning a (transport, protocol) tuple. See **asyncio.BaseEventLoop.create\_connection()**

***class*cozmo.IOSConnector(*serial=None*, *\*\*kw*)**

Connects to an attached iOS device over USB.

Opens a connection to the first iOS device that’s found to be running the Cozmo app in SDK mode.

iTunes (or another service providing usbmuxd) must be installed in order for this connector to be able to open a connection to a device.

An instance of this class can be passed to the **connect\_** prefixed functions in this module.

| **Parameters:** | **serial** (*string*) – Serial number of the device to connect to. If None, then connect to the first available iOS device running the Cozmo app in SDK mode. |
| --- | --- |

**connect(*loop*, *protocol\_factory*, *conn\_check*)**

Connect attempts to open a connection transport to the Cozmo app on a device.

On opening a transport it will create a protocol from the supplied factory and connect it to the transport, returning a (transport, protocol) tuple. See **asyncio.BaseEventLoop.create\_connection()**

***class*cozmo.AndroidConnector(*adb\_cmd=None*, *serial=None*, *\*\*kw*)**

Connects to an attached Android device over USB.

This requires the Android Studio command line tools to be installed, specifically adb.

By default the connector will attempt to locate adb (or adb.exe on Windows) in common locations, but it may also be supplied by setting the **ANDROID\_ADB\_PATH** environment variable, or by passing it to the constructor.

An instance of this class can be passed to the **connect\_** prefixed functions in this module.

| **Parameters:** | **serial** (*string*) – Serial number of the device to connect to. If None, then connect to the first available Android device running the Cozmo app in SDK mode. |
| --- | --- |

**connect(*loop*, *protocol\_factory*, *conn\_check*)**

Connect attempts to open a connection transport to the Cozmo app on a device.

On opening a transport it will create a protocol from the supplied factory and connect it to the transport, returning a (transport, protocol) tuple. See **asyncio.BaseEventLoop.create\_connection()**

***class*cozmo.TCPConnector(*tcp\_port=None*, *ip\_addr='127.0.0.1'*, *\*\*kw*)**

Connects to the Cozmo app directly via TCP.

Generally only used for testing and debugging.

Requires that a SDK\_TCP\_PORT environment variable be set to the port number to connect to.

**connect(*loop*, *protocol\_factory*, *conn\_check*)**

Connect attempts to open a connection transport to the Cozmo app on a device.

On opening a transport it will create a protocol from the supplied factory and connect it to the transport, returning a (transport, protocol) tuple. See **asyncio.BaseEventLoop.create\_connection()**

**cozmo.connect(*f*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*)**

Connects to the Cozmo Engine on the mobile device and supplies the connection to a function.

Accepts a function, f, that is given a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) object as a parameter.

The supplied function may be either an asynchronous coroutine function (normally defined using **async def**) or a regular synchronous function.

If an asynchronous function is supplied it will be run on the same thread as the Cozmo event loop and must use the **await** keyword to yield control back to the loop.

If a synchronous function is supplied then it will run on the main thread and Cozmo’s event loop will run on a separate thread. Calls to asynchronous methods returned from CozmoConnection will automatically be translated to synchronous ones.

The connect function will return once the supplied function has completed, as which time it will terminate the connection to the robot.

| **Parameters:** | * **f** (*callable*) – The function to execute * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. |
| --- | --- |

**cozmo.connect\_with\_3dviewer(*f*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*, *enable\_camera\_view=False*, *show\_viewer\_controls=True*)**

Setup a connection to a device and run a user function while displaying Cozmo’s 3d world.

This displays an OpenGL window on the screen with a 3D view of Cozmo’s understanding of the world. Optionally, if use\_viewer is True, a 2nd OpenGL window will also display showing a view of Cozmo’s camera. It will return an error if the current system does not support PyOpenGL.

The function may be either synchronous or asynchronous (defined used **async def**).

The function must accept a **cozmo.CozmoConnection** object as its only argument. This call will block until the supplied function completes.

| **Parameters:** | * **f** (*callable*) – The function to execute * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. * **enable\_camera\_view** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to also open a 2D camera view in a second OpenGL window. * **show\_viewer\_controls** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to draw controls on the view. |
| --- | --- |

**cozmo.connect\_with\_tkviewer(*f*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*, *force\_on\_top=False*)**

Setup a connection to a device and run a user function while displaying Cozmo’s camera.

This displays a Tk window on the screen showing a view of Cozmo’s camera. It will return an error if the current system does not support Tk.

The function may be either synchronous or asynchronous (defined used **async def**).

The function must accept a **cozmo.CozmoConnection** object as its only argument. This call will block until the supplied function completes.

| **Parameters:** | * **f** (*callable*) – The function to execute * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. * **force\_on\_top** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether the window should be forced on top of all others |
| --- | --- |

**cozmo.connect\_on\_loop(*loop*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*)**

Uses the supplied event loop to connect to a device.

Will run the event loop in the current thread until the connection succeeds or fails.

If you do not want/need to manage your own loop, then use the [**connect()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.connect) function to handle setup/teardown and execute a user-supplied function.

| **Parameters:** | * **loop** ([**asyncio.BaseEventLoop**](https://docs.python.org/3.5/library/asyncio-eventloop.html#asyncio.BaseEventLoop)) – The event loop to use to connect to Cozmo. * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default, it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. |
| --- | --- |
| **Returns:** | A [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) instance. |

**cozmo.run\_program(*f*, *use\_viewer=False*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*, *force\_viewer\_on\_top=False*, *deprecated\_filter='default'*, *use\_3d\_viewer=False*, *show\_viewer\_controls=True*, *exit\_on\_connection\_error=True*)**

Connect to Cozmo and run the provided program/function f.

| **Parameters:** | * **f** (*callable*) – The function to execute, accepts a connected [**cozmo.robot.Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) as the parameter. * **use\_viewer** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to display a view of Cozmo’s camera in a window. * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. * **force\_viewer\_on\_top** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether the window should be forced on top of all others (only relevant if use\_viewer is True). Note that this is ignored if use\_3d\_viewer is True (as it’s not currently supported on that windowing system). * **deprecated\_filter** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – The filter for any DeprecationWarning messages. This is defaulted to “default” which shows the warning once per location. You can hide all deprecated warnings by passing in “ignore”, see <https://docs.python.org/3/library/warnings.html#warning-filter> for more information. * **use\_3d\_viewer** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to display a 3D view of Cozmo’s understanding of the world in a window. Note that if both this and use\_viewer are set then the 2D camera view will render in an OpenGL window instead of a TkView window. * **show\_viewer\_controls** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to draw controls on the view. * **exit\_on\_connection\_error** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specify whether the program should exit on connection error or should an error be raised. Default to true. |
| --- | --- |

**cozmo.setup\_basic\_logging(*general\_log\_level=None*, *protocol\_log\_level=None*, *protocol\_log\_messages='all'*, *target=<\_io.TextIOWrapper name='<stderr>' mode='w' encoding='UTF-8'>*, *deprecated\_filter='default'*)**

Helper to perform basic setup of the Python logging machinery.

The SDK defines two loggers:

* [**logger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.logger) (“cozmo.general”) - For general purpose information about events within the SDK; and
* [**logger\_protocol**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.logger_protocol) (“cozmo.protocol”) - For low level communication messages between the device and the SDK.

Generally only [**logger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.logger) is interesting.

| **Parameters:** | * **general\_log\_level** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – ‘DEBUG’, ‘INFO’, ‘WARN’, ‘ERROR’ or an equivalent constant from the [**logging**](https://docs.python.org/3.5/library/logging.html#module-logging) module. If None then a value will be read from the COZMO\_LOG\_LEVEL environment variable. * **protocol\_log\_level** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – as general\_log\_level. If None then a value will be read from the COZMO\_PROTOCOL\_LOG\_LEVEL environment variable. * **protocol\_log\_messages** ([*list*](https://docs.python.org/3.5/library/stdtypes.html#list)) – The low level messages that should be logged to the protocol log. Defaults to all. Will read from the COMZO\_PROTOCOL\_LOG\_MESSAGES if available which should be a comma separated list of message names (case sensitive). * **target** ([*object*](https://docs.python.org/3.5/library/functions.html#object)) – The stream to send the log data to; defaults to stderr * **deprecated\_filter** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – The filter for any DeprecationWarning messages. This is defaulted to “default” which shows the warning once per location. You can hide all deprecated warnings by passing in “ignore”, see <https://docs.python.org/3/library/warnings.html#warning-filter> for more information. |
| --- | --- |

***exception*cozmo.CozmoSDKException**

Base class of all Cozmo SDK exceptions.

***exception*cozmo.SDKShutdown**

Raised when the SDK is being shut down

***exception*cozmo.StopPropogation**

Raised by event handlers to prevent further handlers from being triggered.

***exception*cozmo.AnimationsNotLoaded**

Raised if an attempt is made to play a named animation before animations have been received.

***exception*cozmo.ActionError**

Base class for errors that occur with robot actions.

***exception*cozmo.ConnectionError**

Base class for errors regarding connection to the device.

***exception*cozmo.ConnectionAborted**

Raised if the connection to the device is unexpectedly lost.

***exception*cozmo.ConnectionCheckFailed**

Raised if the connection check has failed.

***exception*cozmo.NoDevicesFound**

Raised if no devices connected running Cozmo in SDK mode

***exception*cozmo.SDKVersionMismatch(*message*, *sdk\_version*, *sdk\_app\_version*, *app\_version*, *\*args*)**

Raised if the Cozmo SDK version is not compatible with the software running on the device.

**app\_version*= None***

The version of the App that was detected, and is incompatible, in Major.Minor.Patch format.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**sdk\_app\_version*= None***

The version of the App that this SDK is compatible with in Major.Minor.Patch format.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**sdk\_version*= None***

The SDK version number in Major.Minor.Patch format. See [SDK vs. App Versions](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/sdk-versions.html#sdk-versions) for which App version is compatible with each SDK version.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

***exception*cozmo.NotPickupable**

Raised if an attempt is made to pick up or place an object that can’t be picked up by Cozmo

***exception*cozmo.CannotPlaceObjectsOnThis**

Raised if an attempt is made to place an object on top of an invalid object

***exception*cozmo.RobotBusy**

Raised if an attempt is made to perform an action while another action is still running.

***exception*cozmo.InvalidOpenGLGlutImplementation**

Raised by opengl viewer if no valid GLUT implementation available.

# **cozmo.action**

Actions encapsulate specific high-level tasks that the Cozmo robot can perform. They have a definite beginning and end.

These tasks include picking up an object, rotating in place, saying text, etc.

Actions are usually triggered by a call to a method on the [**cozmo.robot.Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) class such as [**turn\_in\_place()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.turn_in_place)

The call will return an object that subclasses [**Action**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.Action) that can be used to cancel the action, or be observed to wait or be notified when the action completes (or fails) by calling its [**wait\_for()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.wait_for) or [**add\_event\_handler()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.add_event_handler) methods.

**Warning**

Unless you pass in\_parallel=True when starting the action, no other action can be active at the same time. Attempting to trigger a non-parallel action when another action is already in progress will result in a [**RobotBusy**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.RobotBusy) exception being raised.

When using in\_parallel=True you may see an action fail with the result [**ActionResults.TRACKS\_LOCKED**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.ActionResults.TRACKS_LOCKED) - this indicates that another in-progress action has already locked that movement track (e.g. two actions cannot move the head at the same time).

**Classes**

| [**Action**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.Action)(\*, conn, robot, \*\*kw) | An action holds the state of an in-progress robot action |
| --- | --- |
| [**ActionResults**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.ActionResults) | The possible result values for an Action. |
| [**EvtActionCompleted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.EvtActionCompleted)(\*\*kwargs) | Triggered when a robot action has completed or failed. |
| [**EvtActionStarted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.EvtActionStarted)(\*\*kwargs) | Triggered when a robot starts an action. |

**cozmo.action.ACTION\_IDLE*= 'action\_idle'***

Action idle state

| **Type:** | string |
| --- | --- |

**cozmo.action.ACTION\_RUNNING*= 'action\_running'***

Action running state

| **Type:** | string |
| --- | --- |

**cozmo.action.ACTION\_SUCCEEDED*= 'action\_succeeded'***

Action succeeded state

| **Type:** | string |
| --- | --- |

**cozmo.action.ACTION\_FAILED*= 'action\_failed'***

Action failed state

| **Type:** | string |
| --- | --- |

**cozmo.action.ACTION\_ABORTING*= 'action\_aborting'***

Action failed state

| **Type:** | string |
| --- | --- |

***class*cozmo.action.EvtActionStarted(*\*\*kwargs*)**

Triggered when a robot starts an action.

**action*= 'The action that started'***

***class*cozmo.action.EvtActionCompleted(*\*\*kwargs*)**

Triggered when a robot action has completed or failed.

**action*= 'The action that completed'***

**failure\_code*= 'A failure code such as "cancelled"'***

**failure\_reason*= 'A human-readable failure reason'***

**state*= 'The state of the action; either cozmo.action.ACTION\_SUCCEEDED or cozmo.action.ACTION\_FAILED'***

***class*cozmo.action.Action(*\**, *conn*, *robot*, *\*\*kw*)**

An action holds the state of an in-progress robot action

**abort(*log\_abort\_messages=False*)**

Trigger the robot to abort the running action.

| **Parameters:** | **log\_abort\_messages** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to log info on the action that is aborted. |
| --- | --- |
| **Raises:** | ValueError if the action is not currently being executed. |

**conn*= None***

The connection on which the action was sent.

| **Type:** | [**CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) |
| --- | --- |

**failure\_reason**

Both values will be None if no failure has occurred.

| **Type:** | tuple of (failure\_code, failure\_reason) |
| --- | --- |

**has\_failed**

True if the action has failed.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**has\_succeeded**

True if the action has succeeded.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_aborting**

True if the action is aborting (will soon be either succeeded or failed).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_completed**

True if the action has completed (either succeeded or failed).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_running**

True if the action is currently in progress.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**on\_completed(*handler*)**

Triggers a handler when the action completes.

| **Parameters:** | **handler** (*callable*) – An event handler which accepts arguments suited to the [**EvtActionCompleted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.EvtActionCompleted) event. See **cozmo.event.add\_event\_handler()** for more information. |
| --- | --- |

**result**

The result of running the action.

| **Type:** | An attribute of [**ActionResults**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.ActionResults) |
| --- | --- |

**robot*= None***

Th robot instance executing the action.

| **Type:** | [**Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) |
| --- | --- |

**state**

The current internal state of the action as a string.

Will match one of the constants: [**ACTION\_IDLE**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.ACTION_IDLE) [**ACTION\_RUNNING**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.ACTION_RUNNING) [**ACTION\_SUCCEEDED**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.ACTION_SUCCEEDED) [**ACTION\_FAILED**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.ACTION_FAILED) [**ACTION\_ABORTING**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.ACTION_ABORTING)

| **Type:** | string |
| --- | --- |

**wait\_for\_completed(*timeout=None*)**

Waits for the action to complete.

| **Parameters:** | **timeout** ([*int*](https://docs.python.org/3.5/library/functions.html#int) *or* [*None*](https://docs.python.org/3.5/library/constants.html#None)) – Maximum time in seconds to wait for the event. Pass None to wait indefinitely. |
| --- | --- |
| **Returns:** | The [**EvtActionCompleted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.EvtActionCompleted) event instance |
| **Raises:** | [**asyncio.TimeoutError**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.TimeoutError) |

***class*cozmo.action.ActionResults**

The possible result values for an Action.

An Action’s result is set when the action completes.

**ABORT*= \_ActionResult(name='ABORT', id=50331648)***

Action aborted itself (e.g. had invalid attributes, or a runtime failure).

**ANIM\_ABORTED*= \_ActionResult(name='ANIM\_ABORTED', id=50331649)***

Animation Action aborted itself (e.g. there was an error playing the animation).

**BAD\_MARKER*= \_ActionResult(name='BAD\_MARKER', id=50331650)***

There was an error related to vision markers.

**BAD\_OBJECT*= \_ActionResult(name='BAD\_OBJECT', id=50331652)***

There was a problem with the Object ID provided (e.g. there is no Object with that ID).

**BAD\_POSE*= \_ActionResult(name='BAD\_POSE', id=50331653)***

There was a problem with the Pose provided.

**CANCELLED\_WHILE\_RUNNING*= \_ActionResult(name='CANCELLED\_WHILE\_RUNNING', id=33554432)***

Action was cancelled (e.g. via [**abort\_all\_actions()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.abort_all_actions) or [**Action.abort()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.Action.abort)).

**DID\_NOT\_REACH\_PREACTION\_POSE*= \_ActionResult(name='DID\_NOT\_REACH\_PREACTION\_POSE', id=67108865)***

Failed to get into position.

**FAILED\_TRAVERSING\_PATH*= \_ActionResult(name='FAILED\_TRAVERSING\_PATH', id=67108866)***

Failed to follow the planned path.

**FOLLOWING\_PATH\_BUT\_NOT\_TRAVERSING*= \_ActionResult(name='FOLLOWING\_PATH\_BUT\_NOT\_TRAVERSING', id=50331656)***

There was an error following the planned path.

**INTERRUPTED*= \_ActionResult(name='INTERRUPTED', id=50331657)***

The action was interrupted by another Action or Behavior.

**INVALID\_OFF\_TREADS\_STATE*= \_ActionResult(name='INVALID\_OFF\_TREADS\_STATE', id=50331658)***

The robot ended up in an “off treads state” not valid for this action (e.g. the robot was placed on its back while executing a turn)

**LAST\_PICK\_AND\_PLACE\_FAILED*= \_ActionResult(name='LAST\_PICK\_AND\_PLACE\_FAILED', id=67108867)***

The previous attempt to pick and place an object failed.

**MISMATCHED\_UP\_AXIS*= \_ActionResult(name='MISMATCHED\_UP\_AXIS', id=50331659)***

The Up Axis of a carried object doesn’t match the desired placement pose.

**MOTOR\_STOPPED\_MAKING\_PROGRESS*= \_ActionResult(name='MOTOR\_STOPPED\_MAKING\_PROGRESS', id=67108868)***

The required motor isn’t moving so the action cannot complete.

**NOT\_CARRYING\_OBJECT\_ABORT*= \_ActionResult(name='NOT\_CARRYING\_OBJECT\_ABORT', id=50331665)***

No object is being carried, but the action requires one.

**NOT\_CARRYING\_OBJECT\_RETRY*= \_ActionResult(name='NOT\_CARRYING\_OBJECT\_RETRY', id=67108869)***

Not carrying an object when it was expected, but may succeed if the action is retried.

**NOT\_ON\_CHARGER*= \_ActionResult(name='NOT\_ON\_CHARGER', id=67108870)***

Cozmo is expected to be on the charger, but is not.

**NOT\_STARTED*= \_ActionResult(name='NOT\_STARTED', id=33554433)***

Initial state of an Action to indicate it has not yet started.

**NO\_ANIM\_NAME*= \_ActionResult(name='NO\_ANIM\_NAME', id=50331660)***

No valid Animation name was found.

**NO\_DISTANCE\_SET*= \_ActionResult(name='NO\_DISTANCE\_SET', id=50331661)***

An invalid distance value was given.

**NO\_FACE*= \_ActionResult(name='NO\_FACE', id=50331662)***

There was a problem with the Face ID (e.g. Cozmo doesn’t no where it is).

**NO\_GOAL\_SET*= \_ActionResult(name='NO\_GOAL\_SET', id=50331663)***

No goal pose was set.

**NO\_PREACTION\_POSES*= \_ActionResult(name='NO\_PREACTION\_POSES', id=50331664)***

No pre-action poses were found (e.g. could not get into position).

**NULL\_SUBACTION*= \_ActionResult(name='NULL\_SUBACTION', id=50331666)***

No sub-action was provided.

**PATH\_PLANNING\_FAILED\_ABORT*= \_ActionResult(name='PATH\_PLANNING\_FAILED\_ABORT', id=50331667)***

Cozmo was unable to plan a path.

**PATH\_PLANNING\_FAILED\_RETRY*= \_ActionResult(name='PATH\_PLANNING\_FAILED\_RETRY', id=67108871)***

Cozmo was unable to plan a path, but may succeed if the action is retried.

**PICKUP\_OBJECT\_UNEXPECTEDLY\_MOVING*= \_ActionResult(name='PICKUP\_OBJECT\_UNEXPECTEDLY\_MOVING', id=50331668)***

The object that Cozmo is attempting to pickup is unexpectedly moving (e.g it is being moved by someone else).

**PICKUP\_OBJECT\_UNEXPECTEDLY\_NOT\_MOVING*= \_ActionResult(name='PICKUP\_OBJECT\_UNEXPECTEDLY\_NOT\_MOVING', id=50331669)***

The object that Cozmo thought he was lifting didn’t start moving, so he must have missed.

**PLACEMENT\_GOAL\_NOT\_FREE*= \_ActionResult(name='PLACEMENT\_GOAL\_NOT\_FREE', id=67108872)***

There is no room to place the object at the desired destination.

**RETRY*= \_ActionResult(name='RETRY', id=67108864)***

The Action failed, but may succeed if retried.

**RUNNING*= \_ActionResult(name='RUNNING', id=16777216)***

Action is still running.

**STILL\_CARRYING\_OBJECT*= \_ActionResult(name='STILL\_CARRYING\_OBJECT', id=50331671)***

Cozmo is unexpectedly still carrying an object.

**STILL\_ON\_CHARGER*= \_ActionResult(name='STILL\_ON\_CHARGER', id=67108873)***

Cozmo failed to drive off the charger.

**SUCCESS*= \_ActionResult(name='SUCCESS', id=0)***

Action completed successfully.

**TIMEOUT*= \_ActionResult(name='TIMEOUT', id=50331672)***

The Action timed out before completing correctly.

**TRACKS\_LOCKED*= \_ActionResult(name='TRACKS\_LOCKED', id=50331673)***

One or more animation tracks (Head, Lift, Body, Face, Backpack Lights, Audio) are already being used by another Action.

**UNEXPECTED\_DOCK\_ACTION*= \_ActionResult(name='UNEXPECTED\_DOCK\_ACTION', id=50331674)***

There was an internal error related to an unexpected type of dock action.

**UNEXPECTED\_PITCH\_ANGLE*= \_ActionResult(name='UNEXPECTED\_PITCH\_ANGLE', id=67108874)***

Cozmo’s pitch is at an unexpected angle for the Action.

**VISUAL\_OBSERVATION\_FAILED*= \_ActionResult(name='VISUAL\_OBSERVATION\_FAILED', id=50331677)***

Cozmo did not see the expected result (e.g. unable to see cubes in their expected position after a related action).

# **cozmo.anim**

Animation related classes, functions, events and values.

**Functions**

| [**animation\_completed\_filter**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.animation_completed_filter)() | Creates an [**cozmo.event.Filter**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Filter) to wait specifically for an animation completed event. |
| --- | --- |

**Classes**

| [**Animation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Animation)(anim\_name, loop\_count[, …]) | An Animation describes an actively-playing animation on a robot. |
| --- | --- |
| [**AnimationNames**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.AnimationNames)(conn, \*\*kw) | Holds the set of animation names (strings) returned from the Engine. |
| [**AnimationTrigger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.AnimationTrigger)(trigger, loop\_count, …) | An AnimationTrigger represents a playing animation trigger. |
| [**EvtAnimationCompleted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.EvtAnimationCompleted)(\*\*kwargs) | Triggered when an animation completes. |
| [**EvtAnimationsLoaded**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.EvtAnimationsLoaded)(\*\*kwargs) | Triggered when animations names have been received from the engine |
| [**Triggers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Triggers) | Playing an animation trigger causes the game engine play an animation of a particular type. |

***class*cozmo.anim.EvtAnimationsLoaded(*\*\*kwargs*)**

Triggered when animations names have been received from the engine

***class*cozmo.anim.EvtAnimationCompleted(*\*\*kwargs*)**

Triggered when an animation completes.

**animation\_name*= 'The name of the animation or trigger that completed'***

***class*cozmo.anim.Animation(*anim\_name*, *loop\_count*, *ignore\_body\_track=False*, *ignore\_head\_track=False*, *ignore\_lift\_track=False*, *\*\*kw*)**

An Animation describes an actively-playing animation on a robot.

**anim\_name*= None***

The name of the animation that was dispatched

**ignore\_body\_track*= None***

True to ignore the body track (i.e. the wheels / treads)

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**ignore\_head\_track*= None***

True to ignore the head track

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**ignore\_lift\_track*= None***

True to ignore the lift track

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**loop\_count*= None***

The number of iterations the animation was requested for

***class*cozmo.anim.AnimationTrigger(*trigger*, *loop\_count*, *use\_lift\_safe*, *ignore\_body\_track*, *ignore\_head\_track*, *ignore\_lift\_track*, *\*\*kw*)**

An AnimationTrigger represents a playing animation trigger.

Asking Cozmo to play an AnimationTrigger causes him to pick one of the animations represented by the group.

**ignore\_body\_track*= None***

True to ignore the body track (i.e. the wheels / treads)

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**ignore\_head\_track*= None***

True to ignore the head track

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**ignore\_lift\_track*= None***

True to ignore the lift track

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**loop\_count*= None***

The number of iterations the animation was requested for

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**trigger*= None***

The animation trigger dispatched.

| **Type:** | An attribute of [**cozmo.anim.Triggers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Triggers) |
| --- | --- |

**use\_lift\_safe*= None***

True to automatically ignore the lift track if Cozmo is carrying a cube.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

***class*cozmo.anim.AnimationNames(*conn*, *\*\*kw*)**

Holds the set of animation names (strings) returned from the Engine.

Animation names are dynamically retrieved from the engine when the SDK connects to it, unlike [**Triggers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Triggers) which are defined at runtime.

**is\_loaded**

True if the animation names have been received from the engine.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**refresh()**

Causes the list of animation names to be re-requested from the engine.

Attempting to play an animation while the list is refreshing will result in an AnimationsNotLoaded exception being raised.

Generates an EvtAnimationsLoaded event once completed.

**wait\_for\_loaded(*timeout=None*)**

Wait for the animation names to be loaded from the engine.

| **Returns:** | The [**EvtAnimationsLoaded**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.EvtAnimationsLoaded) instance once loaded |
| --- | --- |
| **Raises:** | [**asyncio.TimeoutError**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.TimeoutError) |

***class*cozmo.anim.Triggers**

Playing an animation trigger causes the game engine play an animation of a particular type.

The engine may pick one of a number of actual animations to play based on Cozmo’s mood or emotion, or with random weighting. Thus playing the same trigger twice may not result in the exact same underlying animation playing twice.

To play an exact animation, use play\_anim with a named animation.

This class holds the set of defined animations triggers to pass to play\_anim\_trigger.

**AcknowledgeFaceInitPause*= \_AnimTrigger(name='AcknowledgeFaceInitPause', id=0)***

**AcknowledgeFaceNamed*= \_AnimTrigger(name='AcknowledgeFaceNamed', id=1)***

**AcknowledgeFaceUnnamed*= \_AnimTrigger(name='AcknowledgeFaceUnnamed', id=2)***

**AcknowledgeObject*= \_AnimTrigger(name='AcknowledgeObject', id=3)***

**AskToBeRightedLeft*= \_AnimTrigger(name='AskToBeRightedLeft', id=4)***

**AskToBeRightedRight*= \_AnimTrigger(name='AskToBeRightedRight', id=5)***

**AudioTestAnim*= \_AnimTrigger(name='AudioTestAnim', id=6)***

**BlockReact*= \_AnimTrigger(name='BlockReact', id=7)***

**BouncerGetIn*= \_AnimTrigger(name='BouncerGetIn', id=8)***

**BouncerGetOut*= \_AnimTrigger(name='BouncerGetOut', id=9)***

**BouncerIdeaToPlay*= \_AnimTrigger(name='BouncerIdeaToPlay', id=10)***

**BouncerIntoScore1*= \_AnimTrigger(name='BouncerIntoScore1', id=11)***

**BouncerIntoScore2*= \_AnimTrigger(name='BouncerIntoScore2', id=12)***

**BouncerIntoScore3*= \_AnimTrigger(name='BouncerIntoScore3', id=13)***

**BouncerRequestToPlay*= \_AnimTrigger(name='BouncerRequestToPlay', id=14)***

**BouncerTimeout*= \_AnimTrigger(name='BouncerTimeout', id=15)***

**BouncerWait*= \_AnimTrigger(name='BouncerWait', id=16)***

**BuildPyramidFirstBlockOnSide*= \_AnimTrigger(name='BuildPyramidFirstBlockOnSide', id=17)***

**BuildPyramidFirstBlockUpright*= \_AnimTrigger(name='BuildPyramidFirstBlockUpright', id=18)***

**BuildPyramidLookForFace*= \_AnimTrigger(name='BuildPyramidLookForFace', id=19)***

**BuildPyramidReactToBase*= \_AnimTrigger(name='BuildPyramidReactToBase', id=20)***

**BuildPyramidSecondBlockOnSide*= \_AnimTrigger(name='BuildPyramidSecondBlockOnSide', id=21)***

**BuildPyramidSecondBlockUpright*= \_AnimTrigger(name='BuildPyramidSecondBlockUpright', id=22)***

**BuildPyramidSuccess*= \_AnimTrigger(name='BuildPyramidSuccess', id=23)***

**BuildPyramidThankUser*= \_AnimTrigger(name='BuildPyramidThankUser', id=24)***

**BuildPyramidThirdBlockOnSide*= \_AnimTrigger(name='BuildPyramidThirdBlockOnSide', id=25)***

**BuildPyramidThirdBlockUpright*= \_AnimTrigger(name='BuildPyramidThirdBlockUpright', id=26)***

**CantHandleTallStack*= \_AnimTrigger(name='CantHandleTallStack', id=27)***

**CodeLab123Go*= \_AnimTrigger(name='CodeLab123Go', id=30)***

**CodeLabAmazed*= \_AnimTrigger(name='CodeLabAmazed', id=31)***

**CodeLabBlink*= \_AnimTrigger(name='CodeLabBlink', id=32)***

**CodeLabBored*= \_AnimTrigger(name='CodeLabBored', id=33)***

**CodeLabCat*= \_AnimTrigger(name='CodeLabCat', id=34)***

**CodeLabCelebrate*= \_AnimTrigger(name='CodeLabCelebrate', id=35)***

**CodeLabChatty*= \_AnimTrigger(name='CodeLabChatty', id=36)***

**CodeLabChicken*= \_AnimTrigger(name='CodeLabChicken', id=37)***

**CodeLabConducting*= \_AnimTrigger(name='CodeLabConducting', id=38)***

**CodeLabCow*= \_AnimTrigger(name='CodeLabCow', id=39)***

**CodeLabCurious*= \_AnimTrigger(name='CodeLabCurious', id=40)***

**CodeLabDancingMambo*= \_AnimTrigger(name='CodeLabDancingMambo', id=41)***

**CodeLabDejected*= \_AnimTrigger(name='CodeLabDejected', id=42)***

**CodeLabDizzy*= \_AnimTrigger(name='CodeLabDizzy', id=43)***

**CodeLabDizzyEnd*= \_AnimTrigger(name='CodeLabDizzyEnd', id=44)***

**CodeLabDog*= \_AnimTrigger(name='CodeLabDog', id=45)***

**CodeLabDuck*= \_AnimTrigger(name='CodeLabDuck', id=46)***

**CodeLabElephant*= \_AnimTrigger(name='CodeLabElephant', id=47)***

**CodeLabEnergyEat*= \_AnimTrigger(name='CodeLabEnergyEat', id=48)***

**CodeLabEnter*= \_AnimTrigger(name='CodeLabEnter', id=49)***

**CodeLabExcited*= \_AnimTrigger(name='CodeLabExcited', id=50)***

**CodeLabExit*= \_AnimTrigger(name='CodeLabExit', id=51)***

**CodeLabFireTruck*= \_AnimTrigger(name='CodeLabFireTruck', id=52)***

**CodeLabFrog*= \_AnimTrigger(name='CodeLabFrog', id=53)***

**CodeLabFrustrated*= \_AnimTrigger(name='CodeLabFrustrated', id=54)***

**CodeLabGetInPos*= \_AnimTrigger(name='CodeLabGetInPos', id=55)***

**CodeLabGhoul*= \_AnimTrigger(name='CodeLabGhoul', id=56)***

**CodeLabHappy*= \_AnimTrigger(name='CodeLabHappy', id=57)***

**CodeLabHeadsUp*= \_AnimTrigger(name='CodeLabHeadsUp', id=58)***

**CodeLabHelium*= \_AnimTrigger(name='CodeLabHelium', id=59)***

**CodeLabHiccup*= \_AnimTrigger(name='CodeLabHiccup', id=60)***

**CodeLabIDK*= \_AnimTrigger(name='CodeLabIDK', id=61)***

**CodeLabIdle*= \_AnimTrigger(name='CodeLabIdle', id=62)***

**CodeLabLose*= \_AnimTrigger(name='CodeLabLose', id=63)***

**CodeLabNo*= \_AnimTrigger(name='CodeLabNo', id=64)***

**CodeLabPartyTime*= \_AnimTrigger(name='CodeLabPartyTime', id=65)***

**CodeLabRattleSnake*= \_AnimTrigger(name='CodeLabRattleSnake', id=66)***

**CodeLabReactHappy*= \_AnimTrigger(name='CodeLabReactHappy', id=67)***

**CodeLabRooster*= \_AnimTrigger(name='CodeLabRooster', id=68)***

**CodeLabScaredCozmo*= \_AnimTrigger(name='CodeLabScaredCozmo', id=69)***

**CodeLabScaryCozmo*= \_AnimTrigger(name='CodeLabScaryCozmo', id=70)***

**CodeLabSheep*= \_AnimTrigger(name='CodeLabSheep', id=71)***

**CodeLabSleep*= \_AnimTrigger(name='CodeLabSleep', id=72)***

**CodeLabSneeze*= \_AnimTrigger(name='CodeLabSneeze', id=73)***

**CodeLabSquint1*= \_AnimTrigger(name='CodeLabSquint1', id=74)***

**CodeLabSquint2*= \_AnimTrigger(name='CodeLabSquint2', id=75)***

**CodeLabStaring*= \_AnimTrigger(name='CodeLabStaring', id=76)***

**CodeLabSurprise*= \_AnimTrigger(name='CodeLabSurprise', id=77)***

**CodeLabTakaTaka*= \_AnimTrigger(name='CodeLabTakaTaka', id=78)***

**CodeLabTapCube*= \_AnimTrigger(name='CodeLabTapCube', id=79)***

**CodeLabThinking*= \_AnimTrigger(name='CodeLabThinking', id=80)***

**CodeLabTiger*= \_AnimTrigger(name='CodeLabTiger', id=81)***

**CodeLabTwitch*= \_AnimTrigger(name='CodeLabTwitch', id=82)***

**CodeLabUnhappy*= \_AnimTrigger(name='CodeLabUnhappy', id=83)***

**CodeLabVampire*= \_AnimTrigger(name='CodeLabVampire', id=84)***

**CodeLabVictory*= \_AnimTrigger(name='CodeLabVictory', id=85)***

**CodeLabWhee1*= \_AnimTrigger(name='CodeLabWhee1', id=86)***

**CodeLabWhee2*= \_AnimTrigger(name='CodeLabWhee2', id=87)***

**CodeLabWhee3*= \_AnimTrigger(name='CodeLabWhee3', id=88)***

**CodeLabWhee4*= \_AnimTrigger(name='CodeLabWhee4', id=89)***

**CodeLabWhew*= \_AnimTrigger(name='CodeLabWhew', id=90)***

**CodeLabWhoa*= \_AnimTrigger(name='CodeLabWhoa', id=91)***

**CodeLabWin*= \_AnimTrigger(name='CodeLabWin', id=92)***

**CodeLabWondering*= \_AnimTrigger(name='CodeLabWondering', id=93)***

**CodeLabYes*= \_AnimTrigger(name='CodeLabYes', id=94)***

**CodeLabYuck*= \_AnimTrigger(name='CodeLabYuck', id=95)***

**CodeLabZombie*= \_AnimTrigger(name='CodeLabZombie', id=96)***

**ComeHere\_AlreadyHere*= \_AnimTrigger(name='ComeHere\_AlreadyHere', id=100)***

**ComeHere\_SearchForFace*= \_AnimTrigger(name='ComeHere\_SearchForFace', id=101)***

**ComeHere\_SearchForFace\_FoundFace*= \_AnimTrigger(name='ComeHere\_SearchForFace\_FoundFace', id=102)***

**ConnectWakeUp*= \_AnimTrigger(name='ConnectWakeUp', id=97)***

**ConnectWakeUp\_SevereEnergy*= \_AnimTrigger(name='ConnectWakeUp\_SevereEnergy', id=98)***

**ConnectWakeUp\_SevereRepair*= \_AnimTrigger(name='ConnectWakeUp\_SevereRepair', id=99)***

**Count*= \_AnimTrigger(name='Count', id=575)***

**CozmoSaysBadWord*= \_AnimTrigger(name='CozmoSaysBadWord', id=112)***

**CozmoSaysGetIn*= \_AnimTrigger(name='CozmoSaysGetIn', id=103)***

**CozmoSaysGetOut*= \_AnimTrigger(name='CozmoSaysGetOut', id=104)***

**CozmoSaysIdle*= \_AnimTrigger(name='CozmoSaysIdle', id=113)***

**CozmoSaysSpeakGetInLong*= \_AnimTrigger(name='CozmoSaysSpeakGetInLong', id=107)***

**CozmoSaysSpeakGetInMedium*= \_AnimTrigger(name='CozmoSaysSpeakGetInMedium', id=106)***

**CozmoSaysSpeakGetInShort*= \_AnimTrigger(name='CozmoSaysSpeakGetInShort', id=105)***

**CozmoSaysSpeakGetOutLong*= \_AnimTrigger(name='CozmoSaysSpeakGetOutLong', id=110)***

**CozmoSaysSpeakGetOutMedium*= \_AnimTrigger(name='CozmoSaysSpeakGetOutMedium', id=109)***

**CozmoSaysSpeakGetOutShort*= \_AnimTrigger(name='CozmoSaysSpeakGetOutShort', id=108)***

**CozmoSaysSpeakLoop*= \_AnimTrigger(name='CozmoSaysSpeakLoop', id=111)***

**CubeMovedSense*= \_AnimTrigger(name='CubeMovedSense', id=114)***

**CubeMovedUpset*= \_AnimTrigger(name='CubeMovedUpset', id=115)***

**CubePounceFake*= \_AnimTrigger(name='CubePounceFake', id=116)***

**CubePounceGetIn*= \_AnimTrigger(name='CubePounceGetIn', id=117)***

**CubePounceGetOut*= \_AnimTrigger(name='CubePounceGetOut', id=118)***

**CubePounceGetReady*= \_AnimTrigger(name='CubePounceGetReady', id=119)***

**CubePounceGetUnready*= \_AnimTrigger(name='CubePounceGetUnready', id=120)***

**CubePounceIdleLiftDown*= \_AnimTrigger(name='CubePounceIdleLiftDown', id=121)***

**CubePounceIdleLiftUp*= \_AnimTrigger(name='CubePounceIdleLiftUp', id=122)***

**CubePounceLoseHand*= \_AnimTrigger(name='CubePounceLoseHand', id=123)***

**CubePounceLoseRound*= \_AnimTrigger(name='CubePounceLoseRound', id=124)***

**CubePounceLoseSession*= \_AnimTrigger(name='CubePounceLoseSession', id=125)***

**CubePouncePounceClose*= \_AnimTrigger(name='CubePouncePounceClose', id=126)***

**CubePouncePounceNormal*= \_AnimTrigger(name='CubePouncePounceNormal', id=127)***

**CubePounceWinHand*= \_AnimTrigger(name='CubePounceWinHand', id=128)***

**CubePounceWinRound*= \_AnimTrigger(name='CubePounceWinRound', id=129)***

**CubePounceWinSession*= \_AnimTrigger(name='CubePounceWinSession', id=130)***

**DanceMambo*= \_AnimTrigger(name='DanceMambo', id=131)***

**DemoSpeedTapCozmoLose*= \_AnimTrigger(name='DemoSpeedTapCozmoLose', id=133)***

**DemoSpeedTapCozmoWin*= \_AnimTrigger(name='DemoSpeedTapCozmoWin', id=132)***

**DizzyReactionHard*= \_AnimTrigger(name='DizzyReactionHard', id=134)***

**DizzyReactionMedium*= \_AnimTrigger(name='DizzyReactionMedium', id=135)***

**DizzyReactionSoft*= \_AnimTrigger(name='DizzyReactionSoft', id=136)***

**DizzyShakeLoop*= \_AnimTrigger(name='DizzyShakeLoop', id=137)***

**DizzyShakeStop*= \_AnimTrigger(name='DizzyShakeStop', id=138)***

**DizzyStillPickedUp*= \_AnimTrigger(name='DizzyStillPickedUp', id=139)***

**DriveEndAngry*= \_AnimTrigger(name='DriveEndAngry', id=140)***

**DriveEndDefault*= \_AnimTrigger(name='DriveEndDefault', id=141)***

**DriveEndHappy*= \_AnimTrigger(name='DriveEndHappy', id=142)***

**DriveEndLaunch*= \_AnimTrigger(name='DriveEndLaunch', id=143)***

**DriveLoopAngry*= \_AnimTrigger(name='DriveLoopAngry', id=144)***

**DriveLoopDefault*= \_AnimTrigger(name='DriveLoopDefault', id=145)***

**DriveLoopHappy*= \_AnimTrigger(name='DriveLoopHappy', id=146)***

**DriveLoopLaunch*= \_AnimTrigger(name='DriveLoopLaunch', id=147)***

**DriveStartAngry*= \_AnimTrigger(name='DriveStartAngry', id=148)***

**DriveStartDefault*= \_AnimTrigger(name='DriveStartDefault', id=149)***

**DriveStartHappy*= \_AnimTrigger(name='DriveStartHappy', id=150)***

**DriveStartLaunch*= \_AnimTrigger(name='DriveStartLaunch', id=151)***

**DroneModeBackwardDrivingEnd*= \_AnimTrigger(name='DroneModeBackwardDrivingEnd', id=152)***

**DroneModeBackwardDrivingLoop*= \_AnimTrigger(name='DroneModeBackwardDrivingLoop', id=153)***

**DroneModeBackwardDrivingStart*= \_AnimTrigger(name='DroneModeBackwardDrivingStart', id=154)***

**DroneModeCliffEvent*= \_AnimTrigger(name='DroneModeCliffEvent', id=155)***

**DroneModeForwardDrivingEnd*= \_AnimTrigger(name='DroneModeForwardDrivingEnd', id=156)***

**DroneModeForwardDrivingLoop*= \_AnimTrigger(name='DroneModeForwardDrivingLoop', id=157)***

**DroneModeForwardDrivingStart*= \_AnimTrigger(name='DroneModeForwardDrivingStart', id=158)***

**DroneModeGetIn*= \_AnimTrigger(name='DroneModeGetIn', id=159)***

**DroneModeGetOut*= \_AnimTrigger(name='DroneModeGetOut', id=160)***

**DroneModeIdle*= \_AnimTrigger(name='DroneModeIdle', id=161)***

**DroneModeKeepAlive*= \_AnimTrigger(name='DroneModeKeepAlive', id=162)***

**DroneModeTurboDrivingStart*= \_AnimTrigger(name='DroneModeTurboDrivingStart', id=163)***

**EarnedSparks*= \_AnimTrigger(name='EarnedSparks', id=164)***

**FacePlantRoll*= \_AnimTrigger(name='FacePlantRoll', id=165)***

**FacePlantRollArmUp*= \_AnimTrigger(name='FacePlantRollArmUp', id=166)***

**FailedToRightFromFace*= \_AnimTrigger(name='FailedToRightFromFace', id=167)***

**FeedingAteFullEnough\_Normal*= \_AnimTrigger(name='FeedingAteFullEnough\_Normal', id=168)***

**FeedingAteFullEnough\_Severe*= \_AnimTrigger(name='FeedingAteFullEnough\_Severe', id=169)***

**FeedingAteNotFullEnough\_Normal*= \_AnimTrigger(name='FeedingAteNotFullEnough\_Normal', id=170)***

**FeedingAteNotFullEnough\_Severe*= \_AnimTrigger(name='FeedingAteNotFullEnough\_Severe', id=171)***

**FeedingDrivingGetIn\_Severe*= \_AnimTrigger(name='FeedingDrivingGetIn\_Severe', id=172)***

**FeedingDrivingGetOut\_Severe*= \_AnimTrigger(name='FeedingDrivingGetOut\_Severe', id=173)***

**FeedingDrivingLoop\_Severe*= \_AnimTrigger(name='FeedingDrivingLoop\_Severe', id=174)***

**FeedingIdleSearchForFaces\_Normal*= \_AnimTrigger(name='FeedingIdleSearchForFaces\_Normal', id=179)***

**FeedingIdleSearchForFaces\_Severe*= \_AnimTrigger(name='FeedingIdleSearchForFaces\_Severe', id=180)***

**FeedingIdleSearch\_Normal*= \_AnimTrigger(name='FeedingIdleSearch\_Normal', id=177)***

**FeedingIdleSearch\_Severe*= \_AnimTrigger(name='FeedingIdleSearch\_Severe', id=178)***

**FeedingIdleWaitForFullCube\_Normal*= \_AnimTrigger(name='FeedingIdleWaitForFullCube\_Normal', id=181)***

**FeedingIdleWaitForFullCube\_Severe*= \_AnimTrigger(name='FeedingIdleWaitForFullCube\_Severe', id=182)***

**FeedingIdleWaitForShakeNoHead\_Severe*= \_AnimTrigger(name='FeedingIdleWaitForShakeNoHead\_Severe', id=183)***

**FeedingIdleWaitForShake\_Normal*= \_AnimTrigger(name='FeedingIdleWaitForShake\_Normal', id=184)***

**FeedingIdleWaitForShake\_Severe*= \_AnimTrigger(name='FeedingIdleWaitForShake\_Severe', id=185)***

**FeedingInterrupted*= \_AnimTrigger(name='FeedingInterrupted', id=186)***

**FeedingInterrupted\_Severe*= \_AnimTrigger(name='FeedingInterrupted\_Severe', id=187)***

**FeedingPlaceLiftOnCube\_Normal*= \_AnimTrigger(name='FeedingPlaceLiftOnCube\_Normal', id=175)***

**FeedingPlaceLiftOnCube\_Severe*= \_AnimTrigger(name='FeedingPlaceLiftOnCube\_Severe', id=176)***

**FeedingReactToFullCube\_Normal*= \_AnimTrigger(name='FeedingReactToFullCube\_Normal', id=192)***

**FeedingReactToFullCube\_Severe*= \_AnimTrigger(name='FeedingReactToFullCube\_Severe', id=193)***

**FeedingReactToSeeCube\_Normal*= \_AnimTrigger(name='FeedingReactToSeeCube\_Normal', id=194)***

**FeedingReactToSeeCube\_Severe*= \_AnimTrigger(name='FeedingReactToSeeCube\_Severe', id=195)***

**FeedingReactToShake\_Normal*= \_AnimTrigger(name='FeedingReactToShake\_Normal', id=196)***

**FeedingReactToShake\_Severe*= \_AnimTrigger(name='FeedingReactToShake\_Severe', id=197)***

**FeedingSearchFailure*= \_AnimTrigger(name='FeedingSearchFailure', id=190)***

**FeedingSearchFailure\_Severe*= \_AnimTrigger(name='FeedingSearchFailure\_Severe', id=191)***

**FeedingSearchRequest*= \_AnimTrigger(name='FeedingSearchRequest', id=188)***

**FeedingSearchRequest\_Severe*= \_AnimTrigger(name='FeedingSearchRequest\_Severe', id=189)***

**FistBumpIdle*= \_AnimTrigger(name='FistBumpIdle', id=198)***

**FistBumpLeftHanging*= \_AnimTrigger(name='FistBumpLeftHanging', id=202)***

**FistBumpRequestOnce*= \_AnimTrigger(name='FistBumpRequestOnce', id=199)***

**FistBumpRequestRetry*= \_AnimTrigger(name='FistBumpRequestRetry', id=200)***

**FistBumpSuccess*= \_AnimTrigger(name='FistBumpSuccess', id=201)***

**FlipDownFromBack*= \_AnimTrigger(name='FlipDownFromBack', id=203)***

**FrustratedByFailure*= \_AnimTrigger(name='FrustratedByFailure', id=204)***

**FrustratedByFailureMajor*= \_AnimTrigger(name='FrustratedByFailureMajor', id=205)***

**GameSetupGetIn*= \_AnimTrigger(name='GameSetupGetIn', id=206)***

**GameSetupGetOut*= \_AnimTrigger(name='GameSetupGetOut', id=207)***

**GameSetupIdle*= \_AnimTrigger(name='GameSetupIdle', id=208)***

**GameSetupReaction*= \_AnimTrigger(name='GameSetupReaction', id=209)***

**GoToSleepGetIn*= \_AnimTrigger(name='GoToSleepGetIn', id=210)***

**GoToSleepGetOut*= \_AnimTrigger(name='GoToSleepGetOut', id=211)***

**GoToSleepOff*= \_AnimTrigger(name='GoToSleepOff', id=212)***

**GoToSleepSleeping*= \_AnimTrigger(name='GoToSleepSleeping', id=213)***

**GuardDogBusted*= \_AnimTrigger(name='GuardDogBusted', id=214)***

**GuardDogCubeDisconnect*= \_AnimTrigger(name='GuardDogCubeDisconnect', id=215)***

**GuardDogFakeout*= \_AnimTrigger(name='GuardDogFakeout', id=216)***

**GuardDogInterruption*= \_AnimTrigger(name='GuardDogInterruption', id=217)***

**GuardDogPlayerSuccess*= \_AnimTrigger(name='GuardDogPlayerSuccess', id=218)***

**GuardDogPulse*= \_AnimTrigger(name='GuardDogPulse', id=219)***

**GuardDogSettle*= \_AnimTrigger(name='GuardDogSettle', id=220)***

**GuardDogSleepLoop*= \_AnimTrigger(name='GuardDogSleepLoop', id=221)***

**GuardDogTimeout*= \_AnimTrigger(name='GuardDogTimeout', id=222)***

**GuardDogTimeoutCubesTouched*= \_AnimTrigger(name='GuardDogTimeoutCubesTouched', id=223)***

**GuardDogTimeoutCubesUntouched*= \_AnimTrigger(name='GuardDogTimeoutCubesUntouched', id=224)***

**Hiccup*= \_AnimTrigger(name='Hiccup', id=225)***

**HiccupGetIn*= \_AnimTrigger(name='HiccupGetIn', id=226)***

**HiccupPlayerCure*= \_AnimTrigger(name='HiccupPlayerCure', id=227)***

**HiccupRobotOnBack*= \_AnimTrigger(name='HiccupRobotOnBack', id=228)***

**HiccupRobotOnFace*= \_AnimTrigger(name='HiccupRobotOnFace', id=229)***

**HiccupRobotPickedUp*= \_AnimTrigger(name='HiccupRobotPickedUp', id=230)***

**HiccupSelfCure*= \_AnimTrigger(name='HiccupSelfCure', id=231)***

**HikingDrivingEnd*= \_AnimTrigger(name='HikingDrivingEnd', id=232)***

**HikingDrivingLoop*= \_AnimTrigger(name='HikingDrivingLoop', id=233)***

**HikingDrivingStart*= \_AnimTrigger(name='HikingDrivingStart', id=234)***

**HikingInterestingEdgeThought*= \_AnimTrigger(name='HikingInterestingEdgeThought', id=235)***

**HikingIntro*= \_AnimTrigger(name='HikingIntro', id=236)***

**HikingLookAround*= \_AnimTrigger(name='HikingLookAround', id=237)***

**HikingObserve*= \_AnimTrigger(name='HikingObserve', id=238)***

**HikingReactToEdge*= \_AnimTrigger(name='HikingReactToEdge', id=239)***

**HikingReactToNewArea*= \_AnimTrigger(name='HikingReactToNewArea', id=240)***

**HikingReactToPossibleMarker*= \_AnimTrigger(name='HikingReactToPossibleMarker', id=241)***

**HikingSquintEnd*= \_AnimTrigger(name='HikingSquintEnd', id=242)***

**HikingSquintLoop*= \_AnimTrigger(name='HikingSquintLoop', id=243)***

**HikingSquintStart*= \_AnimTrigger(name='HikingSquintStart', id=244)***

**HikingWakeUpOffCharger*= \_AnimTrigger(name='HikingWakeUpOffCharger', id=245)***

**IdleOnCharger*= \_AnimTrigger(name='IdleOnCharger', id=246)***

**InteractWithFaceTrackingIdle*= \_AnimTrigger(name='InteractWithFaceTrackingIdle', id=247)***

**InteractWithFacesInitialNamed*= \_AnimTrigger(name='InteractWithFacesInitialNamed', id=248)***

**InteractWithFacesInitialUnnamed*= \_AnimTrigger(name='InteractWithFacesInitialUnnamed', id=249)***

**KnockOverEyes*= \_AnimTrigger(name='KnockOverEyes', id=250)***

**KnockOverFailure*= \_AnimTrigger(name='KnockOverFailure', id=251)***

**KnockOverGrabAttempt*= \_AnimTrigger(name='KnockOverGrabAttempt', id=252)***

**KnockOverPreActionNamedFace*= \_AnimTrigger(name='KnockOverPreActionNamedFace', id=253)***

**KnockOverPreActionUnnamedFace*= \_AnimTrigger(name='KnockOverPreActionUnnamedFace', id=254)***

**KnockOverSuccess*= \_AnimTrigger(name='KnockOverSuccess', id=255)***

**LaserAcknowledge*= \_AnimTrigger(name='LaserAcknowledge', id=256)***

**LaserDriveEnd*= \_AnimTrigger(name='LaserDriveEnd', id=257)***

**LaserDriveLoop*= \_AnimTrigger(name='LaserDriveLoop', id=258)***

**LaserDriveStart*= \_AnimTrigger(name='LaserDriveStart', id=259)***

**LaserFace*= \_AnimTrigger(name='LaserFace', id=260)***

**LaserGetOut*= \_AnimTrigger(name='LaserGetOut', id=261)***

**LaserPounce*= \_AnimTrigger(name='LaserPounce', id=262)***

**LookInPlaceForFacesBodyPause*= \_AnimTrigger(name='LookInPlaceForFacesBodyPause', id=263)***

**LookInPlaceForFacesHeadMovePause*= \_AnimTrigger(name='LookInPlaceForFacesHeadMovePause', id=264)***

**MajorFail*= \_AnimTrigger(name='MajorFail', id=265)***

**MajorWin*= \_AnimTrigger(name='MajorWin', id=266)***

**MeetCozmoFirstEnrollmentCelebration*= \_AnimTrigger(name='MeetCozmoFirstEnrollmentCelebration', id=273)***

**MeetCozmoFirstEnrollmentRepeatName*= \_AnimTrigger(name='MeetCozmoFirstEnrollmentRepeatName', id=272)***

**MeetCozmoFirstEnrollmentSayName*= \_AnimTrigger(name='MeetCozmoFirstEnrollmentSayName', id=271)***

**MeetCozmoGetIn*= \_AnimTrigger(name='MeetCozmoGetIn', id=267)***

**MeetCozmoLookFaceGetIn*= \_AnimTrigger(name='MeetCozmoLookFaceGetIn', id=268)***

**MeetCozmoLookFaceGetOut*= \_AnimTrigger(name='MeetCozmoLookFaceGetOut', id=269)***

**MeetCozmoLookFaceInterrupt*= \_AnimTrigger(name='MeetCozmoLookFaceInterrupt', id=276)***

**MeetCozmoReEnrollmentSayName*= \_AnimTrigger(name='MeetCozmoReEnrollmentSayName', id=274)***

**MeetCozmoRenameFaceSayName*= \_AnimTrigger(name='MeetCozmoRenameFaceSayName', id=275)***

**MeetCozmoScanningIdle*= \_AnimTrigger(name='MeetCozmoScanningIdle', id=270)***

**MemoryMatchCozmoFollowTapsSoundOnly*= \_AnimTrigger(name='MemoryMatchCozmoFollowTapsSoundOnly', id=300)***

**MemoryMatchCozmoGetOut*= \_AnimTrigger(name='MemoryMatchCozmoGetOut', id=299)***

**MemoryMatchCozmoLoseHand*= \_AnimTrigger(name='MemoryMatchCozmoLoseHand', id=283)***

**MemoryMatchCozmoWinGame*= \_AnimTrigger(name='MemoryMatchCozmoWinGame', id=284)***

**MemoryMatchCozmoWinHand*= \_AnimTrigger(name='MemoryMatchCozmoWinHand', id=282)***

**MemoryMatchPlayerLoseHand*= \_AnimTrigger(name='MemoryMatchPlayerLoseHand', id=280)***

**MemoryMatchPlayerLoseHandSolo*= \_AnimTrigger(name='MemoryMatchPlayerLoseHandSolo', id=281)***

**MemoryMatchPlayerWinGame*= \_AnimTrigger(name='MemoryMatchPlayerWinGame', id=285)***

**MemoryMatchPlayerWinHand*= \_AnimTrigger(name='MemoryMatchPlayerWinHand', id=277)***

**MemoryMatchPlayerWinHandLong*= \_AnimTrigger(name='MemoryMatchPlayerWinHandLong', id=278)***

**MemoryMatchPlayerWinHandSolo*= \_AnimTrigger(name='MemoryMatchPlayerWinHandSolo', id=279)***

**MemoryMatchPointCenter*= \_AnimTrigger(name='MemoryMatchPointCenter', id=289)***

**MemoryMatchPointCenterFast*= \_AnimTrigger(name='MemoryMatchPointCenterFast', id=294)***

**MemoryMatchPointLeftBig*= \_AnimTrigger(name='MemoryMatchPointLeftBig', id=290)***

**MemoryMatchPointLeftBigFast*= \_AnimTrigger(name='MemoryMatchPointLeftBigFast', id=295)***

**MemoryMatchPointLeftSmall*= \_AnimTrigger(name='MemoryMatchPointLeftSmall', id=291)***

**MemoryMatchPointLeftSmallFast*= \_AnimTrigger(name='MemoryMatchPointLeftSmallFast', id=296)***

**MemoryMatchPointRightBig*= \_AnimTrigger(name='MemoryMatchPointRightBig', id=292)***

**MemoryMatchPointRightBigFast*= \_AnimTrigger(name='MemoryMatchPointRightBigFast', id=297)***

**MemoryMatchPointRightSmall*= \_AnimTrigger(name='MemoryMatchPointRightSmall', id=293)***

**MemoryMatchPointRightSmallFast*= \_AnimTrigger(name='MemoryMatchPointRightSmallFast', id=298)***

**MemoryMatchReactToPattern*= \_AnimTrigger(name='MemoryMatchReactToPattern', id=287)***

**MemoryMatchReactToPatternSolo*= \_AnimTrigger(name='MemoryMatchReactToPatternSolo', id=288)***

**MemoryMatchSoloGameOver*= \_AnimTrigger(name='MemoryMatchSoloGameOver', id=286)***

**NamedFaceInitialGreeting*= \_AnimTrigger(name='NamedFaceInitialGreeting', id=301)***

**NeedsMildLowEnergyRequest*= \_AnimTrigger(name='NeedsMildLowEnergyRequest', id=302)***

**NeedsMildLowPlayRequest*= \_AnimTrigger(name='NeedsMildLowPlayRequest', id=303)***

**NeedsMildLowRepairRequest*= \_AnimTrigger(name='NeedsMildLowRepairRequest', id=304)***

**NeedsSevereLowEnergyCliffReact*= \_AnimTrigger(name='NeedsSevereLowEnergyCliffReact', id=305)***

**NeedsSevereLowEnergyDrivingEnd*= \_AnimTrigger(name='NeedsSevereLowEnergyDrivingEnd', id=306)***

**NeedsSevereLowEnergyDrivingLoop*= \_AnimTrigger(name='NeedsSevereLowEnergyDrivingLoop', id=307)***

**NeedsSevereLowEnergyDrivingStart*= \_AnimTrigger(name='NeedsSevereLowEnergyDrivingStart', id=308)***

**NeedsSevereLowEnergyForceGetOut*= \_AnimTrigger(name='NeedsSevereLowEnergyForceGetOut', id=309)***

**NeedsSevereLowEnergyGetIn*= \_AnimTrigger(name='NeedsSevereLowEnergyGetIn', id=310)***

**NeedsSevereLowEnergyIdle*= \_AnimTrigger(name='NeedsSevereLowEnergyIdle', id=311)***

**NeedsSevereLowEnergyRequest*= \_AnimTrigger(name='NeedsSevereLowEnergyRequest', id=312)***

**NeedsSevereLowEnergySlopeReact*= \_AnimTrigger(name='NeedsSevereLowEnergySlopeReact', id=313)***

**NeedsSevereLowPlayForceGetOut*= \_AnimTrigger(name='NeedsSevereLowPlayForceGetOut', id=314)***

**NeedsSevereLowPlayGetIn*= \_AnimTrigger(name='NeedsSevereLowPlayGetIn', id=315)***

**NeedsSevereLowPlayRequest*= \_AnimTrigger(name='NeedsSevereLowPlayRequest', id=316)***

**NeedsSevereLowRepairCliffReact*= \_AnimTrigger(name='NeedsSevereLowRepairCliffReact', id=317)***

**NeedsSevereLowRepairDrivingEnd*= \_AnimTrigger(name='NeedsSevereLowRepairDrivingEnd', id=318)***

**NeedsSevereLowRepairDrivingLoop*= \_AnimTrigger(name='NeedsSevereLowRepairDrivingLoop', id=319)***

**NeedsSevereLowRepairDrivingStart*= \_AnimTrigger(name='NeedsSevereLowRepairDrivingStart', id=320)***

**NeedsSevereLowRepairForceGetOut*= \_AnimTrigger(name='NeedsSevereLowRepairForceGetOut', id=321)***

**NeedsSevereLowRepairGetIn*= \_AnimTrigger(name='NeedsSevereLowRepairGetIn', id=322)***

**NeedsSevereLowRepairIdle*= \_AnimTrigger(name='NeedsSevereLowRepairIdle', id=323)***

**NeedsSevereLowRepairRequest*= \_AnimTrigger(name='NeedsSevereLowRepairRequest', id=324)***

**NeedsSevereLowRepairSlopeReact*= \_AnimTrigger(name='NeedsSevereLowRepairSlopeReact', id=325)***

**NeutralFace*= \_AnimTrigger(name='NeutralFace', id=326)***

**NothingToDoBoredEvent*= \_AnimTrigger(name='NothingToDoBoredEvent', id=327)***

**NothingToDoBoredIdle*= \_AnimTrigger(name='NothingToDoBoredIdle', id=328)***

**NothingToDoBoredIntro*= \_AnimTrigger(name='NothingToDoBoredIntro', id=329)***

**NothingToDoBoredOutro*= \_AnimTrigger(name='NothingToDoBoredOutro', id=330)***

**OnLearnedPlayerName*= \_AnimTrigger(name='OnLearnedPlayerName', id=331)***

**OnSawNewNamedFace*= \_AnimTrigger(name='OnSawNewNamedFace', id=332)***

**OnSawNewUnnamedFace*= \_AnimTrigger(name='OnSawNewUnnamedFace', id=333)***

**OnSawOldNamedFace*= \_AnimTrigger(name='OnSawOldNamedFace', id=334)***

**OnSawOldUnnamedFace*= \_AnimTrigger(name='OnSawOldUnnamedFace', id=335)***

**OnSpeedtapCozmoConfirm*= \_AnimTrigger(name='OnSpeedtapCozmoConfirm', id=336)***

**OnSpeedtapFakeout*= \_AnimTrigger(name='OnSpeedtapFakeout', id=337)***

**OnSpeedtapGameCozmoWinHighIntensity*= \_AnimTrigger(name='OnSpeedtapGameCozmoWinHighIntensity', id=338)***

**OnSpeedtapGameCozmoWinLowIntensity*= \_AnimTrigger(name='OnSpeedtapGameCozmoWinLowIntensity', id=339)***

**OnSpeedtapGamePlayerWinHighIntensity*= \_AnimTrigger(name='OnSpeedtapGamePlayerWinHighIntensity', id=340)***

**OnSpeedtapGamePlayerWinLowIntensity*= \_AnimTrigger(name='OnSpeedtapGamePlayerWinLowIntensity', id=341)***

**OnSpeedtapGetOut*= \_AnimTrigger(name='OnSpeedtapGetOut', id=342)***

**OnSpeedtapHandCozmoWin*= \_AnimTrigger(name='OnSpeedtapHandCozmoWin', id=343)***

**OnSpeedtapHandPlayerWin*= \_AnimTrigger(name='OnSpeedtapHandPlayerWin', id=344)***

**OnSpeedtapIdle*= \_AnimTrigger(name='OnSpeedtapIdle', id=345)***

**OnSpeedtapRoundCozmoWinHighIntensity*= \_AnimTrigger(name='OnSpeedtapRoundCozmoWinHighIntensity', id=346)***

**OnSpeedtapRoundCozmoWinLowIntensity*= \_AnimTrigger(name='OnSpeedtapRoundCozmoWinLowIntensity', id=347)***

**OnSpeedtapRoundPlayerWinHighIntensity*= \_AnimTrigger(name='OnSpeedtapRoundPlayerWinHighIntensity', id=348)***

**OnSpeedtapRoundPlayerWinLowIntensity*= \_AnimTrigger(name='OnSpeedtapRoundPlayerWinLowIntensity', id=349)***

**OnSpeedtapTap*= \_AnimTrigger(name='OnSpeedtapTap', id=350)***

**OnWaitForCubesMinigameSetup*= \_AnimTrigger(name='OnWaitForCubesMinigameSetup', id=351)***

**OnWiggle*= \_AnimTrigger(name='OnWiggle', id=352)***

**OnboardingBirth*= \_AnimTrigger(name='OnboardingBirth', id=353)***

**OnboardingCubeDockFail*= \_AnimTrigger(name='OnboardingCubeDockFail', id=354)***

**OnboardingDiscoverCube*= \_AnimTrigger(name='OnboardingDiscoverCube', id=355)***

**OnboardingDriveEnd*= \_AnimTrigger(name='OnboardingDriveEnd', id=356)***

**OnboardingDriveLoop*= \_AnimTrigger(name='OnboardingDriveLoop', id=357)***

**OnboardingDriveStart*= \_AnimTrigger(name='OnboardingDriveStart', id=358)***

**OnboardingEyesOn*= \_AnimTrigger(name='OnboardingEyesOn', id=359)***

**OnboardingGetOut*= \_AnimTrigger(name='OnboardingGetOut', id=373)***

**OnboardingHelloPlayer*= \_AnimTrigger(name='OnboardingHelloPlayer', id=360)***

**OnboardingHelloWorld*= \_AnimTrigger(name='OnboardingHelloWorld', id=361)***

**OnboardingIdle*= \_AnimTrigger(name='OnboardingIdle', id=362)***

**OnboardingIdleEnergy*= \_AnimTrigger(name='OnboardingIdleEnergy', id=363)***

**OnboardingIdlePlay*= \_AnimTrigger(name='OnboardingIdlePlay', id=364)***

**OnboardingIdleRepair*= \_AnimTrigger(name='OnboardingIdleRepair', id=365)***

**OnboardingInteractWithCube*= \_AnimTrigger(name='OnboardingInteractWithCube', id=366)***

**OnboardingPreBirth*= \_AnimTrigger(name='OnboardingPreBirth', id=367)***

**OnboardingReactToCube*= \_AnimTrigger(name='OnboardingReactToCube', id=368)***

**OnboardingReactToCubePutDown*= \_AnimTrigger(name='OnboardingReactToCubePutDown', id=369)***

**OnboardingReactToFace*= \_AnimTrigger(name='OnboardingReactToFace', id=370)***

**OnboardingSoundOnlyLiftEffortPickup*= \_AnimTrigger(name='OnboardingSoundOnlyLiftEffortPickup', id=371)***

**OnboardingSoundOnlyLiftEffortPlaceLow*= \_AnimTrigger(name='OnboardingSoundOnlyLiftEffortPlaceLow', id=372)***

**OnboardingWakeUpDriveOffCharger*= \_AnimTrigger(name='OnboardingWakeUpDriveOffCharger', id=374)***

**PatternGuessNewIdea*= \_AnimTrigger(name='PatternGuessNewIdea', id=28)***

**PatternGuessThinking*= \_AnimTrigger(name='PatternGuessThinking', id=29)***

**PeekABooGetIn*= \_AnimTrigger(name='PeekABooGetIn', id=375)***

**PeekABooGetOutHappy*= \_AnimTrigger(name='PeekABooGetOutHappy', id=376)***

**PeekABooGetOutSad*= \_AnimTrigger(name='PeekABooGetOutSad', id=377)***

**PeekABooHighIntensity*= \_AnimTrigger(name='PeekABooHighIntensity', id=378)***

**PeekABooIdle*= \_AnimTrigger(name='PeekABooIdle', id=379)***

**PeekABooLowIntensity*= \_AnimTrigger(name='PeekABooLowIntensity', id=380)***

**PeekABooMedIntensity*= \_AnimTrigger(name='PeekABooMedIntensity', id=381)***

**PeekABooNoUserInteraction*= \_AnimTrigger(name='PeekABooNoUserInteraction', id=382)***

**PeekABooShort*= \_AnimTrigger(name='PeekABooShort', id=383)***

**PeekABooSurprised*= \_AnimTrigger(name='PeekABooSurprised', id=384)***

**PetDetectionCat*= \_AnimTrigger(name='PetDetectionCat', id=385)***

**PetDetectionDog*= \_AnimTrigger(name='PetDetectionDog', id=386)***

**PetDetectionShort*= \_AnimTrigger(name='PetDetectionShort', id=387)***

**PetDetectionShort\_Cat*= \_AnimTrigger(name='PetDetectionShort\_Cat', id=388)***

**PetDetectionShort\_Dog*= \_AnimTrigger(name='PetDetectionShort\_Dog', id=389)***

**PetDetectionSneeze*= \_AnimTrigger(name='PetDetectionSneeze', id=390)***

**PickupHelperPreActionNamedFace*= \_AnimTrigger(name='PickupHelperPreActionNamedFace', id=391)***

**PickupHelperPreActionUnnamedFace*= \_AnimTrigger(name='PickupHelperPreActionUnnamedFace', id=392)***

**PlacedOnCharger*= \_AnimTrigger(name='PlacedOnCharger', id=393)***

**PopAWheelieInitial*= \_AnimTrigger(name='PopAWheelieInitial', id=394)***

**PopAWheeliePreActionNamedFace*= \_AnimTrigger(name='PopAWheeliePreActionNamedFace', id=395)***

**PopAWheeliePreActionUnnamedFace*= \_AnimTrigger(name='PopAWheeliePreActionUnnamedFace', id=396)***

**PopAWheelieRealign*= \_AnimTrigger(name='PopAWheelieRealign', id=397)***

**PopAWheelieRetry*= \_AnimTrigger(name='PopAWheelieRetry', id=398)***

**PounceDriveEnd*= \_AnimTrigger(name='PounceDriveEnd', id=399)***

**PounceDriveLoop*= \_AnimTrigger(name='PounceDriveLoop', id=400)***

**PounceDriveStart*= \_AnimTrigger(name='PounceDriveStart', id=401)***

**PounceFace*= \_AnimTrigger(name='PounceFace', id=402)***

**PounceFail*= \_AnimTrigger(name='PounceFail', id=403)***

**PounceGetOut*= \_AnimTrigger(name='PounceGetOut', id=404)***

**PounceInitial*= \_AnimTrigger(name='PounceInitial', id=405)***

**PouncePounce*= \_AnimTrigger(name='PouncePounce', id=406)***

**PounceSuccess*= \_AnimTrigger(name='PounceSuccess', id=407)***

**ProceduralLive*= \_AnimTrigger(name='ProceduralLive', id=408)***

**PutDownBlockKeepAlive*= \_AnimTrigger(name='PutDownBlockKeepAlive', id=409)***

**PutDownBlockPutDown*= \_AnimTrigger(name='PutDownBlockPutDown', id=410)***

**ReactToBlockPickupSuccess*= \_AnimTrigger(name='ReactToBlockPickupSuccess', id=411)***

**ReactToBlockRetryPickup*= \_AnimTrigger(name='ReactToBlockRetryPickup', id=412)***

**ReactToCliff*= \_AnimTrigger(name='ReactToCliff', id=413)***

**ReactToCliffDetectorStop*= \_AnimTrigger(name='ReactToCliffDetectorStop', id=414)***

**ReactToFalling*= \_AnimTrigger(name='ReactToFalling', id=415)***

**ReactToImpact*= \_AnimTrigger(name='ReactToImpact', id=416)***

**ReactToMotorCalibration*= \_AnimTrigger(name='ReactToMotorCalibration', id=418)***

**ReactToNewBlockAsk*= \_AnimTrigger(name='ReactToNewBlockAsk', id=419)***

**ReactToNewBlockBig*= \_AnimTrigger(name='ReactToNewBlockBig', id=420)***

**ReactToNewBlockSmall*= \_AnimTrigger(name='ReactToNewBlockSmall', id=421)***

**ReactToObstacle*= \_AnimTrigger(name='ReactToObstacle', id=417)***

**ReactToOnLeftSide*= \_AnimTrigger(name='ReactToOnLeftSide', id=422)***

**ReactToOnRightSide*= \_AnimTrigger(name='ReactToOnRightSide', id=423)***

**ReactToPerchedOnBlock*= \_AnimTrigger(name='ReactToPerchedOnBlock', id=424)***

**ReactToPickup*= \_AnimTrigger(name='ReactToPickup', id=425)***

**ReactToPokeReaction*= \_AnimTrigger(name='ReactToPokeReaction', id=426)***

**ReactToPokeStartled*= \_AnimTrigger(name='ReactToPokeStartled', id=427)***

**ReactToUnexpectedMovement*= \_AnimTrigger(name='ReactToUnexpectedMovement', id=428)***

**ReactToUnexpectedMovement\_Severe\_Energy*= \_AnimTrigger(name='ReactToUnexpectedMovement\_Severe\_Energy', id=429)***

**ReactToUnexpectedMovement\_Severe\_Repair*= \_AnimTrigger(name='ReactToUnexpectedMovement\_Severe\_Repair', id=430)***

**RepairFailMild*= \_AnimTrigger(name='RepairFailMild', id=431)***

**RepairFailSevere*= \_AnimTrigger(name='RepairFailSevere', id=432)***

**RepairFixMildGetIn*= \_AnimTrigger(name='RepairFixMildGetIn', id=433)***

**RepairFixMildGetOut*= \_AnimTrigger(name='RepairFixMildGetOut', id=434)***

**RepairFixMildGetReady*= \_AnimTrigger(name='RepairFixMildGetReady', id=435)***

**RepairFixMildHeadDown*= \_AnimTrigger(name='RepairFixMildHeadDown', id=436)***

**RepairFixMildHeadUp*= \_AnimTrigger(name='RepairFixMildHeadUp', id=437)***

**RepairFixMildIdle*= \_AnimTrigger(name='RepairFixMildIdle', id=438)***

**RepairFixMildLiftDown*= \_AnimTrigger(name='RepairFixMildLiftDown', id=439)***

**RepairFixMildLiftUp*= \_AnimTrigger(name='RepairFixMildLiftUp', id=440)***

**RepairFixMildLowerLift*= \_AnimTrigger(name='RepairFixMildLowerLift', id=441)***

**RepairFixMildRaiseLift*= \_AnimTrigger(name='RepairFixMildRaiseLift', id=442)***

**RepairFixMildRoundReact*= \_AnimTrigger(name='RepairFixMildRoundReact', id=443)***

**RepairFixMildWheelsBack*= \_AnimTrigger(name='RepairFixMildWheelsBack', id=444)***

**RepairFixMildWheelsForward*= \_AnimTrigger(name='RepairFixMildWheelsForward', id=445)***

**RepairFixSevereGetIn*= \_AnimTrigger(name='RepairFixSevereGetIn', id=446)***

**RepairFixSevereGetOut*= \_AnimTrigger(name='RepairFixSevereGetOut', id=447)***

**RepairFixSevereGetReady*= \_AnimTrigger(name='RepairFixSevereGetReady', id=448)***

**RepairFixSevereHeadDown*= \_AnimTrigger(name='RepairFixSevereHeadDown', id=449)***

**RepairFixSevereHeadUp*= \_AnimTrigger(name='RepairFixSevereHeadUp', id=450)***

**RepairFixSevereIdle*= \_AnimTrigger(name='RepairFixSevereIdle', id=451)***

**RepairFixSevereLiftDown*= \_AnimTrigger(name='RepairFixSevereLiftDown', id=452)***

**RepairFixSevereLiftUp*= \_AnimTrigger(name='RepairFixSevereLiftUp', id=453)***

**RepairFixSevereLowerLift*= \_AnimTrigger(name='RepairFixSevereLowerLift', id=454)***

**RepairFixSevereRaiseLift*= \_AnimTrigger(name='RepairFixSevereRaiseLift', id=455)***

**RepairFixSevereRoundReact*= \_AnimTrigger(name='RepairFixSevereRoundReact', id=456)***

**RepairFixSevereWheelsBack*= \_AnimTrigger(name='RepairFixSevereWheelsBack', id=457)***

**RepairFixSevereWheelsForward*= \_AnimTrigger(name='RepairFixSevereWheelsForward', id=458)***

**RepairIdleFullyRepaired*= \_AnimTrigger(name='RepairIdleFullyRepaired', id=459)***

**RepairPartRepaired\_Head\_Mild*= \_AnimTrigger(name='RepairPartRepaired\_Head\_Mild', id=460)***

**RepairPartRepaired\_Lift\_Mild*= \_AnimTrigger(name='RepairPartRepaired\_Lift\_Mild', id=461)***

**RepairPartRepaired\_Tread\_Mild*= \_AnimTrigger(name='RepairPartRepaired\_Tread\_Mild', id=462)***

**RequestGameInterrupt*= \_AnimTrigger(name='RequestGameInterrupt', id=463)***

**RequestGameKeepAwayAccept0*= \_AnimTrigger(name='RequestGameKeepAwayAccept0', id=464)***

**RequestGameKeepAwayAccept1*= \_AnimTrigger(name='RequestGameKeepAwayAccept1', id=465)***

**RequestGameKeepAwayDeny0*= \_AnimTrigger(name='RequestGameKeepAwayDeny0', id=466)***

**RequestGameKeepAwayDeny1*= \_AnimTrigger(name='RequestGameKeepAwayDeny1', id=467)***

**RequestGameKeepAwayIdle0*= \_AnimTrigger(name='RequestGameKeepAwayIdle0', id=468)***

**RequestGameKeepAwayIdle1*= \_AnimTrigger(name='RequestGameKeepAwayIdle1', id=469)***

**RequestGameKeepAwayInitial0*= \_AnimTrigger(name='RequestGameKeepAwayInitial0', id=470)***

**RequestGameKeepAwayInitial1*= \_AnimTrigger(name='RequestGameKeepAwayInitial1', id=471)***

**RequestGameKeepAwayPreDrive0*= \_AnimTrigger(name='RequestGameKeepAwayPreDrive0', id=472)***

**RequestGameKeepAwayPreDrive1*= \_AnimTrigger(name='RequestGameKeepAwayPreDrive1', id=473)***

**RequestGameKeepAwayRequest0*= \_AnimTrigger(name='RequestGameKeepAwayRequest0', id=474)***

**RequestGameKeepAwayRequest1*= \_AnimTrigger(name='RequestGameKeepAwayRequest1', id=475)***

**RequestGameMemoryMatchAccept0*= \_AnimTrigger(name='RequestGameMemoryMatchAccept0', id=476)***

**RequestGameMemoryMatchAccept1*= \_AnimTrigger(name='RequestGameMemoryMatchAccept1', id=477)***

**RequestGameMemoryMatchDeny0*= \_AnimTrigger(name='RequestGameMemoryMatchDeny0', id=478)***

**RequestGameMemoryMatchDeny1*= \_AnimTrigger(name='RequestGameMemoryMatchDeny1', id=479)***

**RequestGameMemoryMatchIdle0*= \_AnimTrigger(name='RequestGameMemoryMatchIdle0', id=480)***

**RequestGameMemoryMatchIdle1*= \_AnimTrigger(name='RequestGameMemoryMatchIdle1', id=481)***

**RequestGameMemoryMatchInitial0*= \_AnimTrigger(name='RequestGameMemoryMatchInitial0', id=482)***

**RequestGameMemoryMatchInitial1*= \_AnimTrigger(name='RequestGameMemoryMatchInitial1', id=483)***

**RequestGameMemoryMatchPreDrive0*= \_AnimTrigger(name='RequestGameMemoryMatchPreDrive0', id=484)***

**RequestGameMemoryMatchPreDrive1*= \_AnimTrigger(name='RequestGameMemoryMatchPreDrive1', id=485)***

**RequestGameMemoryMatchRequest0*= \_AnimTrigger(name='RequestGameMemoryMatchRequest0', id=486)***

**RequestGameMemoryMatchRequest1*= \_AnimTrigger(name='RequestGameMemoryMatchRequest1', id=487)***

**RequestGamePickupFail*= \_AnimTrigger(name='RequestGamePickupFail', id=488)***

**RequestGameSpeedTapAccept0*= \_AnimTrigger(name='RequestGameSpeedTapAccept0', id=489)***

**RequestGameSpeedTapAccept1*= \_AnimTrigger(name='RequestGameSpeedTapAccept1', id=490)***

**RequestGameSpeedTapDeny0*= \_AnimTrigger(name='RequestGameSpeedTapDeny0', id=491)***

**RequestGameSpeedTapDeny1*= \_AnimTrigger(name='RequestGameSpeedTapDeny1', id=492)***

**RequestGameSpeedTapIdle0*= \_AnimTrigger(name='RequestGameSpeedTapIdle0', id=493)***

**RequestGameSpeedTapIdle1*= \_AnimTrigger(name='RequestGameSpeedTapIdle1', id=494)***

**RequestGameSpeedTapInitial0*= \_AnimTrigger(name='RequestGameSpeedTapInitial0', id=495)***

**RequestGameSpeedTapInitial1*= \_AnimTrigger(name='RequestGameSpeedTapInitial1', id=496)***

**RequestGameSpeedTapPreDrive0*= \_AnimTrigger(name='RequestGameSpeedTapPreDrive0', id=497)***

**RequestGameSpeedTapPreDrive1*= \_AnimTrigger(name='RequestGameSpeedTapPreDrive1', id=498)***

**RequestGameSpeedTapRequest0*= \_AnimTrigger(name='RequestGameSpeedTapRequest0', id=499)***

**RequestGameSpeedTapRequest1*= \_AnimTrigger(name='RequestGameSpeedTapRequest1', id=500)***

**RollBlockInitial*= \_AnimTrigger(name='RollBlockInitial', id=501)***

**RollBlockPreActionNamedFace*= \_AnimTrigger(name='RollBlockPreActionNamedFace', id=502)***

**RollBlockPreActionUnnamedFace*= \_AnimTrigger(name='RollBlockPreActionUnnamedFace', id=503)***

**RollBlockPutDown*= \_AnimTrigger(name='RollBlockPutDown', id=504)***

**RollBlockRealign*= \_AnimTrigger(name='RollBlockRealign', id=505)***

**RollBlockRetry*= \_AnimTrigger(name='RollBlockRetry', id=506)***

**RollBlockSuccess*= \_AnimTrigger(name='RollBlockSuccess', id=507)***

**SdkTextToSpeech*= \_AnimTrigger(name='SdkTextToSpeech', id=508)***

**Shiver*= \_AnimTrigger(name='Shiver', id=509)***

**Shocked*= \_AnimTrigger(name='Shocked', id=510)***

**Singing\_100bpm*= \_AnimTrigger(name='Singing\_100bpm', id=512)***

**Singing\_120bpm*= \_AnimTrigger(name='Singing\_120bpm', id=513)***

**Singing\_80bpm*= \_AnimTrigger(name='Singing\_80bpm', id=511)***

**Singing\_GetIn*= \_AnimTrigger(name='Singing\_GetIn', id=514)***

**Singing\_GetOut*= \_AnimTrigger(name='Singing\_GetOut', id=515)***

**Sleeping*= \_AnimTrigger(name='Sleeping', id=516)***

**SoftSparkUpgradeLift*= \_AnimTrigger(name='SoftSparkUpgradeLift', id=517)***

**SoftSparkUpgradeTracks*= \_AnimTrigger(name='SoftSparkUpgradeTracks', id=518)***

**SoundOnlyLiftEffortPickup*= \_AnimTrigger(name='SoundOnlyLiftEffortPickup', id=519)***

**SoundOnlyLiftEffortPlaceHigh*= \_AnimTrigger(name='SoundOnlyLiftEffortPlaceHigh', id=520)***

**SoundOnlyLiftEffortPlaceLow*= \_AnimTrigger(name='SoundOnlyLiftEffortPlaceLow', id=521)***

**SoundOnlyLiftEffortPlaceRoll*= \_AnimTrigger(name='SoundOnlyLiftEffortPlaceRoll', id=522)***

**SoundOnlyRamIntoBlock*= \_AnimTrigger(name='SoundOnlyRamIntoBlock', id=523)***

**SoundOnlyTurnSmall*= \_AnimTrigger(name='SoundOnlyTurnSmall', id=524)***

**SparkDrivingLoop*= \_AnimTrigger(name='SparkDrivingLoop', id=525)***

**SparkDrivingStart*= \_AnimTrigger(name='SparkDrivingStart', id=526)***

**SparkDrivingStop*= \_AnimTrigger(name='SparkDrivingStop', id=527)***

**SparkFailure*= \_AnimTrigger(name='SparkFailure', id=528)***

**SparkGetIn*= \_AnimTrigger(name='SparkGetIn', id=529)***

**SparkGetOut*= \_AnimTrigger(name='SparkGetOut', id=530)***

**SparkIdle*= \_AnimTrigger(name='SparkIdle', id=531)***

**SparkPickupFinalCubeReaction*= \_AnimTrigger(name='SparkPickupFinalCubeReaction', id=532)***

**SparkPickupInitialCubeReaction*= \_AnimTrigger(name='SparkPickupInitialCubeReaction', id=533)***

**SparkSuccess*= \_AnimTrigger(name='SparkSuccess', id=534)***

**SpeedTapDrivingEnd*= \_AnimTrigger(name='SpeedTapDrivingEnd', id=535)***

**SpeedTapDrivingLoop*= \_AnimTrigger(name='SpeedTapDrivingLoop', id=536)***

**SpeedTapDrivingStart*= \_AnimTrigger(name='SpeedTapDrivingStart', id=537)***

**StackBlocksSuccess*= \_AnimTrigger(name='StackBlocksSuccess', id=538)***

**StartSleeping*= \_AnimTrigger(name='StartSleeping', id=539)***

**SuccessfulWheelie*= \_AnimTrigger(name='SuccessfulWheelie', id=540)***

**Surprise*= \_AnimTrigger(name='Surprise', id=541)***

**TurtleRoll*= \_AnimTrigger(name='TurtleRoll', id=542)***

**UnitTestAnim*= \_AnimTrigger(name='UnitTestAnim', id=543)***

**VC\_Alrighty*= \_AnimTrigger(name='VC\_Alrighty', id=544)***

**VC\_HowAreYouDoing\_AllGood*= \_AnimTrigger(name='VC\_HowAreYouDoing\_AllGood', id=545)***

**VC\_Listening*= \_AnimTrigger(name='VC\_Listening', id=546)***

**VC\_LookDownForLaser*= \_AnimTrigger(name='VC\_LookDownForLaser', id=547)***

**VC\_LookDownNoLaser*= \_AnimTrigger(name='VC\_LookDownNoLaser', id=548)***

**VC\_NoFollowupCommand\_NoFace*= \_AnimTrigger(name='VC\_NoFollowupCommand\_NoFace', id=549)***

**VC\_NoFollowupCommand\_WithFace*= \_AnimTrigger(name='VC\_NoFollowupCommand\_WithFace', id=550)***

**VC\_Refuse\_energy*= \_AnimTrigger(name='VC\_Refuse\_energy', id=551)***

**VC\_Refuse\_repair*= \_AnimTrigger(name='VC\_Refuse\_repair', id=552)***

**VC\_Refuse\_sparks*= \_AnimTrigger(name='VC\_Refuse\_sparks', id=553)***

**VC\_StartledWakeup*= \_AnimTrigger(name='VC\_StartledWakeup', id=554)***

**WaitOnSideLoop*= \_AnimTrigger(name='WaitOnSideLoop', id=555)***

**WorkoutPostLift\_highEnergy*= \_AnimTrigger(name='WorkoutPostLift\_highEnergy', id=556)***

**WorkoutPostLift\_lowEnergy*= \_AnimTrigger(name='WorkoutPostLift\_lowEnergy', id=557)***

**WorkoutPostLift\_mediumEnergy*= \_AnimTrigger(name='WorkoutPostLift\_mediumEnergy', id=558)***

**WorkoutPreLift\_highEnergy*= \_AnimTrigger(name='WorkoutPreLift\_highEnergy', id=559)***

**WorkoutPreLift\_lowEnergy*= \_AnimTrigger(name='WorkoutPreLift\_lowEnergy', id=560)***

**WorkoutPreLift\_mediumEnergy*= \_AnimTrigger(name='WorkoutPreLift\_mediumEnergy', id=561)***

**WorkoutPutDown\_highEnergy*= \_AnimTrigger(name='WorkoutPutDown\_highEnergy', id=562)***

**WorkoutPutDown\_lowEnergy*= \_AnimTrigger(name='WorkoutPutDown\_lowEnergy', id=563)***

**WorkoutPutDown\_lowEnergy\_simple*= \_AnimTrigger(name='WorkoutPutDown\_lowEnergy\_simple', id=564)***

**WorkoutPutDown\_mediumEnergy*= \_AnimTrigger(name='WorkoutPutDown\_mediumEnergy', id=565)***

**WorkoutStrongLift\_highEnergy*= \_AnimTrigger(name='WorkoutStrongLift\_highEnergy', id=566)***

**WorkoutStrongLift\_lowEnergy*= \_AnimTrigger(name='WorkoutStrongLift\_lowEnergy', id=567)***

**WorkoutStrongLift\_mediumEnergy*= \_AnimTrigger(name='WorkoutStrongLift\_mediumEnergy', id=568)***

**WorkoutTransition\_highEnergy*= \_AnimTrigger(name='WorkoutTransition\_highEnergy', id=569)***

**WorkoutTransition\_lowEnergy*= \_AnimTrigger(name='WorkoutTransition\_lowEnergy', id=570)***

**WorkoutTransition\_mediumEnergy*= \_AnimTrigger(name='WorkoutTransition\_mediumEnergy', id=571)***

**WorkoutWeakLift\_highEnergy*= \_AnimTrigger(name='WorkoutWeakLift\_highEnergy', id=572)***

**WorkoutWeakLift\_lowEnergy*= \_AnimTrigger(name='WorkoutWeakLift\_lowEnergy', id=573)***

**WorkoutWeakLift\_mediumEnergy*= \_AnimTrigger(name='WorkoutWeakLift\_mediumEnergy', id=574)***

**cozmo.anim.animation\_completed\_filter()**

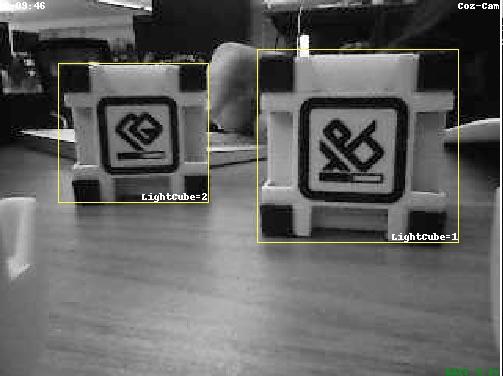
Creates an [**cozmo.event.Filter**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Filter) to wait specifically for an animation completed event.

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html)

# **cozmo.annotate**

Camera image annotation.



This module defines an [**ImageAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator) class used by [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) to add annotations to camera images received by Cozmo.

This can include the location of cubes, faces and pets that Cozmo currently sees, along with user-defined custom annotations.

The ImageAnnotator instance can be accessed as [**cozmo.world.World.image\_annotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.image_annotator).

**Functions**

| [**add\_img\_box\_to\_image**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.add_img_box_to_image)(image, box, color[, text]) | Draw a box on an image and optionally add text. |
| --- | --- |
| [**add\_polygon\_to\_image**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.add_polygon_to_image)(image, poly\_points, …) | Draw a polygon on an image |
| [**annotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.annotator)(f) | A decorator for converting a regular function/method into an Annotator. |

**Classes**

| [**Annotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.Annotator)(img\_annotator[, priority]) | Annotation base class |
| --- | --- |
| [**FaceAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.FaceAnnotator)(img\_annotator[, box\_color]) | Adds annotations of currently detected faces to a camera image. |
| [**ImageAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator)(world, \*\*kw) | ImageAnnotator applies annotations to the camera image received from the robot. |
| [**ImageText**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageText)(text[, position, align, color, …]) | ImageText represents some text that can be applied to an image. |
| [**ObjectAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ObjectAnnotator)(img\_annotator[, object\_colors]) | Adds object annotations to an Image. |
| [**PetAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.PetAnnotator)(img\_annotator[, box\_color]) | Adds annotations of currently detected pets to a camera image. |
| [**TextAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.TextAnnotator)(img\_annotator, text) | Adds simple text annotations to a camera image. |

**cozmo.annotate.TOP\_LEFT*= 5***

Top left position

**cozmo.annotate.TOP\_RIGHT*= 6***

Top right position

**cozmo.annotate.BOTTOM\_LEFT*= 9***

Bottom left position

**cozmo.annotate.BOTTOM\_RIGHT*= 10***

Bottom right position

**cozmo.annotate.RESAMPLE\_MODE\_NEAREST*= 0***

Fastest resampling mode, use nearest pixel

**cozmo.annotate.RESAMPLE\_MODE\_BILINEAR*= 2***

Slower, but smoother, resampling mode - linear interpolation from 2x2 grid of pixels

***class*cozmo.annotate.ImageText(*text*, *position=10*, *align='left'*, *color='white'*, *font=None*, *line\_spacing=3*, *outline\_color=None*, *full\_outline=True*)**

ImageText represents some text that can be applied to an image.

The class allows the text to be placed at various positions inside a bounding box within the image itself.

| **Parameters:** | * **text** (*string*) – The text to display; may contain newlines * **position** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Where on the screen to render the text - A constant such at TOP\_LEFT or BOTTOM\_RIGHT * **align** (*string*) – Text alignment for multi-line strings * **color** (*string*) – Color to use for the text - see [**PIL.ImageColor**](https://pillow.readthedocs.io/en/latest/reference/ImageColor.html#module-PIL.ImageColor) * **font** ([**PIL.ImageFont**](https://pillow.readthedocs.io/en/latest/reference/ImageFont.html#module-PIL.ImageFont)) – Font to use (None for a default font) * **line\_spacing** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The vertical spacing for multi-line strings * **outline\_color** (*string*) – Color to use for the outline - see [**PIL.ImageColor**](https://pillow.readthedocs.io/en/latest/reference/ImageColor.html#module-PIL.ImageColor) - use None for no outline. * **full\_outline** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True if the outline should surround the text, otherwise a cheaper drop-shadow is displayed. Only relevant if outline\_color is specified. |
| --- | --- |

**render(*draw*, *bounds*)**

Renders the text onto an image within the specified bounding box.

| **Parameters:** | * **draw** (**PIL.ImageDraw.ImageDraw**) – The drawable surface to write on * **bounds** (*tuple of int*) – (top\_left\_x, top\_left\_y, bottom\_right\_x, bottom\_right\_y): bounding box |
| --- | --- |
| **Returns:** | The same **PIL.ImageDraw.ImageDraw** object as was passed-in with text applied. |

***class*cozmo.annotate.Annotator(*img\_annotator*, *priority=None*)**

Annotation base class

Subclasses of Annotator handle applying a single annotation to an image.

**apply(*image*, *scale*)**

Applies the annotation to the image.

**enabled*= None***

Set enabled to false to prevent the annotator being called

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**img\_annotator*= None***

The object managing camera annotations

| **Type:** | [**ImageAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator) |
| --- | --- |

**priority*= 100***

The priority of the annotator - Annotators with higher numbered priorities are applied first.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**world*= None***

The world object for the robot who owns the camera

| **Type:** | [**World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) |
| --- | --- |

***class*cozmo.annotate.ObjectAnnotator(*img\_annotator*, *object\_colors=None*)**

Adds object annotations to an Image.

This handles [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube) objects as well as custom objects.

**apply(*image*, *scale*)**

Applies the annotation to the image.

**label\_for\_obj(*obj*)**

Fetch a label to display for the object.

Override or replace to customize.

***class*cozmo.annotate.FaceAnnotator(*img\_annotator*, *box\_color=None*)**

Adds annotations of currently detected faces to a camera image.

This handles the display of [**cozmo.faces.Face**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face) objects.

**apply(*image*, *scale*)**

Applies the annotation to the image.

**label\_for\_face(*obj*)**

Fetch a label to display for the face.

Override or replace to customize.

***class*cozmo.annotate.PetAnnotator(*img\_annotator*, *box\_color=None*)**

Adds annotations of currently detected pets to a camera image.

This handles the display of [**cozmo.pets.Pet**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#cozmo.pets.Pet) objects.

**apply(*image*, *scale*)**

Applies the annotation to the image.

**label\_for\_pet(*obj*)**

Fetch a label to display for the pet.

Override or replace to customize.

***class*cozmo.annotate.TextAnnotator(*img\_annotator*, *text*)**

Adds simple text annotations to a camera image.

**apply(*image*, *scale*)**

Applies the annotation to the image.

***class*cozmo.annotate.ImageAnnotator(*world*, *\*\*kw*)**

ImageAnnotator applies annotations to the camera image received from the robot.

This is instantiated by [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) and is accessible as [**cozmo.world.World.image\_annotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.image_annotator).

By default it defines three active annotators named **objects**, **faces** and **pets**.

The **objects** annotator adds a box around each object (such as light cubes) that Cozmo can see. The **faces** annotator adds a box around each person’s face that Cozmo can recognize. The **pets** annotator adds a box around each pet face that Cozmo can recognize.

Custom annotations can be defined by calling [**add\_annotator()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator.add_annotator) with a name of your choosing and an instance of a [**Annotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.Annotator) subclass, or use a regular function wrapped with the [**annotator()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.annotator) decorator.

Individual annotations can be disabled and re-enabled using the [**disable\_annotator()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator.disable_annotator) and [**enable\_annotator()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator.enable_annotator) methods.

All annotations can be disabled by setting the [**annotation\_enabled**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator.annotation_enabled) property to False.

E.g. to disable face annotations, call **coz.world.image\_annotator.disable\_annotator('faces')**

Annotators each have a priority number associated with them. Annotators with a larger priority number are rendered first and may be overdrawn by those with a lower/smaller priority number.

**add\_annotator(*name*, *annotator*)**

Adds a new annotator for display.

Annotators are enabled by default.

| **Parameters:** | * **name** (*string*) – An arbitrary name for the annotator; must not already be defined * **annotator** ([**Annotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.Annotator) or callable) – The annotator to add may either by an instance of Annotator, or a factory callable that will return an instance of Annotator. The callable will be called with an ImageAnnotator instance as its first argument. |
| --- | --- |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) if the annotator is already defined. |

**add\_static\_text(*name*, *text*, *color='white'*, *position=5*)**

Add some static text to annotated images.

This is a convenience method to create a **TextAnnnotator** and add it to the image.

| **Parameters:** | * **name** (*string*) – An arbitrary name for the annotator; must not already be defined * **text** (str or [**ImageText**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageText) instance) – The text to display may be a plain string, or an ImageText instance * **color** (*string*) – Used if text is a string; defaults to white * **position** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Used if text is a string; defaults to TOP\_LEFT |
| --- | --- |

**annotate\_image(*image*, *scale=None*, *fit\_size=None*, *resample\_mode=0*)**

Called by [**World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) to annotate camera images.

| **Parameters:** | * **image** ([**PIL.Image.Image**](https://pillow.readthedocs.io/en/latest/reference/Image.html#PIL.Image.Image)) – The image to annotate * **scale** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – If set then the base image will be scaled by the supplied multiplier. Cannot be combined with fit\_size * **fit\_size** (*tuple of int*) – If set, then scale the image to fit inside the supplied (width, height) dimensions. The original aspect ratio will be preserved. Cannot be combined with scale. * **resample\_mode** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The resampling mode to use when scaling the image. Should be either [**RESAMPLE\_MODE\_NEAREST**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.RESAMPLE_MODE_NEAREST) (fast) or [**RESAMPLE\_MODE\_BILINEAR**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.RESAMPLE_MODE_BILINEAR) (slower, but smoother). |
| --- | --- |
| **Returns:** | [**PIL.Image.Image**](https://pillow.readthedocs.io/en/latest/reference/Image.html#PIL.Image.Image) |

**annotation\_enabled*= None***

If this attribute is set to false, the [**annotate\_image()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator.annotate_image) method will continue to provide a scaled image, but will not apply any annotations.

**disable\_annotator(*name*)**

Disable a named annotator.

Leaves the annotator as registered, but does not include its output in the annotated image.

| **Parameters:** | **name** (*string*) – The name of the annotator to disable |
| --- | --- |

**enable\_annotator(*name*)**

Enabled a named annotator.

(re)enable an annotator if it was previously disabled.

| **Parameters:** | **name** (*string*) – The name of the annotator to enable |
| --- | --- |

**get\_annotator(*name*)**

Return a named annotator.

| **Parameters:** | **name** (*string*) – The name of the annotator to return |
| --- | --- |
| **Raises:** | KeyError if the annotator isn’t registered |

**remove\_annotator(*name*)**

Remove an annotator.

| **Parameters:** | **name** (*string*) – The name of the annotator to remove as passed to [**add\_annotator()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator.add_annotator). |
| --- | --- |
| **Raises:** | KeyError if the annotator isn’t registered |

**world*= None***

World object that created the annotator.

| **Type:** | [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) |
| --- | --- |

**cozmo.annotate.add\_img\_box\_to\_image(*image*, *box*, *color*, *text=None*)**

Draw a box on an image and optionally add text.

This will draw the outline of a rectangle to the passed in image in the specified color and optionally add one or more pieces of text along the inside edge of the rectangle.

| **Parameters:** | * **image** ([**PIL.Image.Image**](https://pillow.readthedocs.io/en/latest/reference/Image.html#PIL.Image.Image)) – The image to draw on * **box** ([**cozmo.util.ImageBox**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.ImageBox)) – The ImageBox defining the rectangle to draw * **color** (*string*) – A color string suitable for use with PIL - see [**PIL.ImageColor**](https://pillow.readthedocs.io/en/latest/reference/ImageColor.html#module-PIL.ImageColor) * **text** (instance or iterable of [**ImageText**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageText)) – The text to display - may be a single ImageText instance, or any iterable (eg a list of ImageText instances) to display multiple pieces of text. |
| --- | --- |

**cozmo.annotate.add\_polygon\_to\_image(*image*, *poly\_points*, *scale*, *line\_color*, *fill\_color=None*)**

Draw a polygon on an image

This will draw a polygon on the passed-in image in the specified colors and scale.

| **Parameters:** | * **image** ([**PIL.Image.Image**](https://pillow.readthedocs.io/en/latest/reference/Image.html#PIL.Image.Image)) – The image to draw on * **poly\_points** – A sequence of points representing the polygon, where each point has float members (x, y) * **scale** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Scale to multiply each point to match the image scaling * **line\_color** (*string*) – The color for the outline of the polygon. The string value must be a color string suitable for use with PIL - see [**PIL.ImageColor**](https://pillow.readthedocs.io/en/latest/reference/ImageColor.html#module-PIL.ImageColor) * **fill\_color** (*string*) – The color for the inside of the polygon. The string value must be a color string suitable for use with PIL - see [**PIL.ImageColor**](https://pillow.readthedocs.io/en/latest/reference/ImageColor.html#module-PIL.ImageColor) |
| --- | --- |

**cozmo.annotate.annotator(*f*)**

A decorator for converting a regular function/method into an Annotator.

The wrapped function should have a signature of **(image, scale, img\_annotator=None, world=None, \*\*kw)**

# **cozmo.audio**

Audio related classes, functions, events and values.

**Classes**

| [**AudioEvents**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.audio.html#cozmo.audio.AudioEvents) | The possible values for an AudioEvent. |
| --- | --- |

***class*cozmo.audio.AudioEvents**

The possible values for an AudioEvent. Pass one of these event objects to robot.play\_audio() to play the corresponding sound clip. Example: **robot.play\_audio(cozmo.audio.AudioEvents.MusicFunLoop)**

**Invalid*= \_AudioEvent(name='Invalid', id=0)***

Reserved Id for invalid sound events

**MusicBackgroundSilenceOff*= \_AudioEvent(name='MusicBackgroundSilenceOff', id=3328628191)***

Unmute cozmo background music

**MusicBackgroundSilenceOn*= \_AudioEvent(name='MusicBackgroundSilenceOn', id=4207184651)***

Mute cozmo background music

**MusicCubeWhack*= \_AudioEvent(name='MusicCubeWhack', id=3324716189)***

Plays the cube whack music

**MusicFunLoop*= \_AudioEvent(name='Music\_Fun\_Loop', id=1284343309)***

Plays a fun music sound (that loops indefinitely).

**MusicFunLoopStop*= \_AudioEvent(name='Music\_Fun\_Loop\_Stop', id=2123694542)***

Stops all active plays of the fun music sound.

**MusicGlobalStop*= \_AudioEvent(name='MusicGlobalStop', id=2512535841)***

Stop all playing music

**MusicHotPotatoLevel1Loop*= \_AudioEvent(name='MusicHotPotatoLevel1Loop', id=716489468)***

Plays the level 1 hot potato music (Does not repeat)

**MusicHotPotatoLevel1LoopStop*= \_AudioEvent(name='MusicHotPotatoLevel1LoopStop', id=2689778401)***

Stops active plays of the level 1 hot potato music

**MusicHotPotatoLevel2Loop*= \_AudioEvent(name='MusicHotPotatoLevel2Loop', id=2357357213)***

Plays the level 2 hot potato music (Does not repeat)

**MusicHotPotatoLevel2LoopStop*= \_AudioEvent(name='MusicHotPotatoLevel2LoopStop', id=3058241090)***

Stops active plays of the level 2 hot potato music

**MusicHotPotatoLevel3Loop*= \_AudioEvent(name='MusicHotPotatoLevel3Loop', id=2340195146)***

Plays the level 3 hot potato music (Does not repeat)

**MusicHotPotatoLevel3LoopStop*= \_AudioEvent(name='MusicHotPotatoLevel3LoopStop', id=2301371667)***

Stops active plays of the level 3 hot potato music

**MusicHotPotatoLevel4Loop*= \_AudioEvent(name='MusicHotPotatoLevel4Loop', id=4097176099)***

Plays the level 4 hot potato music (Does not repeat)

**MusicHotPotatoLevel4LoopStop*= \_AudioEvent(name='MusicHotPotatoLevel4LoopStop', id=3129138908)***

Stops active plays of the level 4 hot potato music

**MusicMagic8RevealStinger*= \_AudioEvent(name='MusicMagic8RevealStinger', id=1998467689)***

Plays the magic fortune teller reveal music

**MusicMagic8RevealStingerStop*= \_AudioEvent(name='MusicMagic8RevealStingerStop', id=16687378)***

Stops active plays of the magic fortune teller reveal music

**MusicStyle80S1159BpmLoop*= \_AudioEvent(name='MusicStyle80S1159BpmLoop', id=1808088864)***

Plays 80s style music (Does not repeat)

**MusicStyle80S1159BpmLoopStop*= \_AudioEvent(name='MusicStyle80S1159BpmLoopStop', id=1451276801)***

Stops active plays of 80s style music

**MusicStyleDisco1135BpmLoop*= \_AudioEvent(name='MusicStyleDisco1135BpmLoop', id=2717741607)***

Plays disco style music (Does not repeat)

**MusicStyleDisco1135BpmLoopStop*= \_AudioEvent(name='MusicStyleDisco1135BpmLoopStop', id=3069781320)***

Stops active plays of disco style music

**MusicStyleMambo1183BpmLoop*= \_AudioEvent(name='MusicStyleMambo1183BpmLoop', id=1905159696)***

Plays mambo style music (Does not repeat)

**MusicStyleMambo1183BpmLoopStop*= \_AudioEvent(name='MusicStyleMambo1183BpmLoopStop', id=1709157641)***

Stops active plays of mambo style music

**MusicTinyOrchestraBass01Loop*= \_AudioEvent(name='MusicTinyOrchestraBass01Loop', id=1352943452)***

Plays the first tiny orchestra bass track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraBass01LoopStop*= \_AudioEvent(name='MusicTinyOrchestraBass01LoopStop', id=1353021637)***

Stops active plays of the first tiny orchestra bass track

**MusicTinyOrchestraBass02Loop*= \_AudioEvent(name='MusicTinyOrchestraBass02Loop', id=3270107901)***

Plays the second tiny orchestra bass track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraBass02LoopStop*= \_AudioEvent(name='MusicTinyOrchestraBass02LoopStop', id=3636979286)***

Stops active plays of the second tiny orchestra bass track

**MusicTinyOrchestraBass03Loop*= \_AudioEvent(name='MusicTinyOrchestraBass03Loop', id=3677506218)***

Plays the third tiny orchestra bass track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraBass03LoopStop*= \_AudioEvent(name='MusicTinyOrchestraBass03LoopStop', id=1838210855)***

Stops active plays of the third tiny orchestra bass track

**MusicTinyOrchestraBassMode1*= \_AudioEvent(name='MusicTinyOrchestraBassMode1', id=2111780371)***

Turn on the first mode of the synchronized tiny orchestra bass channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraBassMode1Stop*= \_AudioEvent(name='MusicTinyOrchestraBassMode1Stop', id=222737888)***

Turn off the first mode of the synchronized tiny orchestra bass channel

**MusicTinyOrchestraBassMode2*= \_AudioEvent(name='MusicTinyOrchestraBassMode2', id=2111780368)***

Turn on the second mode of the synchronized tiny orchestra bass channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraBassMode2Stop*= \_AudioEvent(name='MusicTinyOrchestraBassMode2Stop', id=1587327405)***

Turn off the second mode of the synchronized tiny orchestra bass channel

**MusicTinyOrchestraBassMode3*= \_AudioEvent(name='MusicTinyOrchestraBassMode3', id=2111780369)***

Turn on the third mode of the synchronized tiny orchestra bass channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraBassMode3Stop*= \_AudioEvent(name='MusicTinyOrchestraBassMode3Stop', id=1731643034)***

Turn off the third mode of the synchronized tiny orchestra bass channel

**MusicTinyOrchestraBassStop*= \_AudioEvent(name='MusicTinyOrchestraBassStop', id=1011028768)***

Turn off all synchronized tiny orchestra bass channels

**MusicTinyOrchestraGlockPluck01Loop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluck01Loop', id=2091022501)***

Plays the first tiny orchestra glock pluck track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraGlockPluck01LoopStop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluck01LoopStop', id=1371551766)***

Stops active plays of the first tiny orchestra glock pluck track

**MusicTinyOrchestraGlockPluck02Loop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluck02Loop', id=1416718788)***

Plays the second tiny orchestra glock pluck track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraGlockPluck02LoopStop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluck02LoopStop', id=1925511301)***

Stops active plays of the second tiny orchestra glock pluck track

**MusicTinyOrchestraGlockPluck03Loop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluck03Loop', id=668702647)***

Plays the third tiny orchestra glock pluck track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraGlockPluck03LoopStop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluck03LoopStop', id=2578133564)***

Stops active plays of the third tiny orchestra glock pluck track

**MusicTinyOrchestraGlockPluckMode1*= \_AudioEvent(name='MusicTinyOrchestraGlockPluckMode1', id=1774266660)***

Turn on the first mode of the synchronized tiny orchestra glock pluck channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraGlockPluckMode1Stop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluckMode1Stop', id=1662844729)***

Turn off the first mode of the synchronized tiny orchestra glock pluck channel

**MusicTinyOrchestraGlockPluckMode2*= \_AudioEvent(name='MusicTinyOrchestraGlockPluckMode2', id=1774266663)***

Turn on the second mode of the synchronized tiny orchestra glock pluck channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraGlockPluckMode2Stop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluckMode2Stop', id=2287415996)***

Turn off the second mode of the synchronized tiny orchestra glock pluck channel

**MusicTinyOrchestraGlockPluckMode3*= \_AudioEvent(name='MusicTinyOrchestraGlockPluckMode3', id=1774266662)***

Turn on the third mode of the synchronized tiny orchestra glock pluck channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraGlockPluckMode3Stop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluckMode3Stop', id=3420658727)***

Turn off the third mode of the synchronized tiny orchestra glock pluck channel

**MusicTinyOrchestraGlockPluckStop*= \_AudioEvent(name='MusicTinyOrchestraGlockPluckStop', id=3644877187)***

Turn off all synchronized tiny orchestra glock pluck channels

**MusicTinyOrchestraInit*= \_AudioEvent(name='MusicTinyOrchestraInit', id=4134268600)***

Initialize the synchronized tiny orchestra system (Will not produce any sound on its own, one of the modes must be triggered)

**MusicTinyOrchestraStop*= \_AudioEvent(name='MusicTinyOrchestraStop', id=4261649076)***

Turn off the synchronized tiny orchestra system

**MusicTinyOrchestraStrings01Loop*= \_AudioEvent(name='MusicTinyOrchestraStrings01Loop', id=846761783)***

Plays the first tiny orchestra string track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraStrings01LoopStop*= \_AudioEvent(name='MusicTinyOrchestraStrings01LoopStop', id=3051334784)***

Stops active plays of the first tiny orchestra strings track

**MusicTinyOrchestraStrings02Loop*= \_AudioEvent(name='MusicTinyOrchestraStrings02Loop', id=363664914)***

Plays the second tiny orchestra string track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraStrings02LoopStop*= \_AudioEvent(name='MusicTinyOrchestraStrings02LoopStop', id=3105309531)***

Stops active plays of the second tiny orchestra strings track

**MusicTinyOrchestraStrings03Loop*= \_AudioEvent(name='MusicTinyOrchestraStrings03Loop', id=2269081637)***

Plays the third tiny orchestra string track (Does not repeat. Does not interact with the synchronized tiny orchestra system)

**MusicTinyOrchestraStrings03LoopStop*= \_AudioEvent(name='MusicTinyOrchestraStrings03LoopStop', id=4031668042)***

Stops active plays of the third tiny orchestra strings track

**MusicTinyOrchestraStringsMode1*= \_AudioEvent(name='MusicTinyOrchestraStringsMode1', id=3175211494)***

Turn on the first mode of the synchronized tiny orchestra strings channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraStringsMode1Stop*= \_AudioEvent(name='MusicTinyOrchestraStringsMode1Stop', id=2836526843)***

Turn off the first mode of the synchronized tiny orchestra strings channel

**MusicTinyOrchestraStringsMode2*= \_AudioEvent(name='MusicTinyOrchestraStringsMode2', id=3175211493)***

Turn on the second mode of the synchronized tiny orchestra strings channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraStringsMode2Stop*= \_AudioEvent(name='MusicTinyOrchestraStringsMode2Stop', id=3381154170)***

Turn off the second mode of the synchronized tiny orchestra strings channel

**MusicTinyOrchestraStringsMode3*= \_AudioEvent(name='MusicTinyOrchestraStringsMode3', id=3175211492)***

Turn on the third mode of the synchronized tiny orchestra strings channel (Requires the tiny orchestra system be initialized, and will loop until the system is turned off)

**MusicTinyOrchestraStringsMode3Stop*= \_AudioEvent(name='MusicTinyOrchestraStringsMode3Stop', id=2812380813)***

Turn off the third mode of the synchronized tiny orchestra strings channel

**MusicTinyOrchestraStringsStop*= \_AudioEvent(name='MusicTinyOrchestraStringsStop', id=3805843413)***

Turn off all synchronized tiny orchestra strings channels

**SfxCubeLight*= \_AudioEvent(name='SfxCubeLight', id=1237348851)***

Plays cube light sound

**SfxCubeLightStop*= \_AudioEvent(name='SfxCubeLightStop', id=2096539976)***

Stops active plays of cube light sound

**SfxFiretruckTimerEnd*= \_AudioEvent(name='SfxFiretruckTimerEnd', id=2490749974)***

Plays firetruck timer end sound

**SfxFiretruckTimerEndStop*= \_AudioEvent(name='SfxFiretruckTimerEndStop', id=421166967)***

Stops active plays of firetruck timer end sound

**SfxFiretruckTimerStart*= \_AudioEvent(name='SfxFiretruckTimerStart', id=2969457917)***

Plays firetruck timer start sound

**SfxFiretruckTimerStartStop*= \_AudioEvent(name='SfxFiretruckTimerStartStop', id=494556302)***

Stops active plays of firetruck timer start sound

**SfxGameLose*= \_AudioEvent(name='SfxGameLose', id=1551222965)***

Plays game lose sound

**SfxGameLoseStop*= \_AudioEvent(name='SfxGameLoseStop', id=347471374)***

Stops active plays of game lose sound

**SfxGameWin*= \_AudioEvent(name='SfxGameWin', id=2440620400)***

Plays game win sound

**SfxGameWinStop*= \_AudioEvent(name='SfxGameWinStop', id=1156077641)***

Stops active plays of game win sound

**SfxGlobalStop*= \_AudioEvent(name='SfxGlobalStop', id=2451909313)***

Stops all playing sound effects

**SfxHotPotatoCubeCharge*= \_AudioEvent(name='SfxHotPotatoCubeCharge', id=1053690875)***

Plays hot potato cube charge sound

**SfxHotPotatoCubeChargeStop*= \_AudioEvent(name='SfxHotPotatoCubeChargeStop', id=3280818964)***

Stops active plays of hot potato cube charge sound

**SfxHotPotatoCubeReady*= \_AudioEvent(name='SfxHotPotatoCubeReady', id=1379908684)***

Plays hot potato cube ready sound

**SfxHotPotatoCubeReadyStop*= \_AudioEvent(name='SfxHotPotatoCubeReadyStop', id=3293236253)***

Stops active plays of hot potato cube ready sound

**SfxHotPotatoPass*= \_AudioEvent(name='SfxHotPotatoPass', id=17262752)***

Plays hot potato pass sound

**SfxHotPotatoPassStop*= \_AudioEvent(name='SfxHotPotatoPassStop', id=2579746409)***

Stops active plays of hot potato pass sound

**SfxHotPotatoTimerEnd*= \_AudioEvent(name='SfxHotPotatoTimerEnd', id=1167243172)***

Plays hot potato timer end sound

**SfxHotPotatoTimerEndStop*= \_AudioEvent(name='SfxHotPotatoTimerEndStop', id=1930994181)***

Stops active plays of hot potato timer end sound

**SfxMagic8MessageReveal*= \_AudioEvent(name='SfxMagic8MessageReveal', id=4071915446)***

Plays magic fortune teller message reveal sound

**SfxMagic8MessageRevealStop*= \_AudioEvent(name='SfxMagic8MessageRevealStop', id=3951709903)***

Stops active plays of magic fortune teller message reveal sound

**SfxMagnetAttract*= \_AudioEvent(name='SfxMagnetAttract', id=3253652261)***

Plays magnet attract sound

**SfxMagnetAttractStop*= \_AudioEvent(name='SfxMagnetAttractStop', id=1573765534)***

Stops active plays of magnet attrack sound

**SfxMagnetRepel*= \_AudioEvent(name='SfxMagnetRepel', id=131125164)***

Plays magnet repel sound

**SfxMagnetRepelStop*= \_AudioEvent(name='SfxMagnetRepelStop', id=827939469)***

Stops active plays of magnet repel sound

**SfxPuttHoleSuccess*= \_AudioEvent(name='Sfx\_Putt\_Hole\_Success', id=3020894301)***

Plays the putt-hole-success sound.

**SfxPuttHoleSuccessStop*= \_AudioEvent(name='Sfx\_Putt\_Hole\_Success\_Stop', id=3048074246)***

Stops all active plays of the putt-hole-success sound.

**SfxSharedCountdown*= \_AudioEvent(name='SfxSharedCountdown', id=3959673678)***

Plays countdown sound

**SfxSharedCountdownStop*= \_AudioEvent(name='SfxSharedCountdownStop', id=769770611)***

Stops active plays of countdown sound

**SfxSharedCubeLightOn*= \_AudioEvent(name='SfxSharedCubeLightOn', id=2279105405)***

Plays cube light on sound

**SfxSharedCubeLightOnStop*= \_AudioEvent(name='SfxSharedCubeLightOnStop', id=3298580386)***

Stops active plays of cube light on sound

**SfxSharedError*= \_AudioEvent(name='SfxSharedError', id=2368558265)***

Plays error sound

**SfxSharedErrorStop*= \_AudioEvent(name='SfxSharedErrorStop', id=63429898)***

Stops active plays of error sound

**SfxSharedSuccess*= \_AudioEvent(name='SfxSharedSuccess', id=211381434)***

Plays success sound

**SfxSharedSuccessStop*= \_AudioEvent(name='SfxSharedSuccessStop', id=2158984211)***

Stops active plays of success sound

**SfxSharedTimerClick*= \_AudioEvent(name='SfxSharedTimerClick', id=3038792837)***

Plays timer click sound

**SfxSharedTimerClickStop*= \_AudioEvent(name='SfxSharedTimerClickStop', id=165689370)***

Stops active plays of timer click sound

**SfxSharedTimerEnd*= \_AudioEvent(name='SfxSharedTimerEnd', id=343938494)***

Plays timer end sound

**SfxSharedTimerEndStop*= \_AudioEvent(name='SfxSharedTimerEndStop', id=1219615815)***

Stops active plays of timer end sound

**SfxSharedTimerWarning*= \_AudioEvent(name='SfxSharedTimerWarning', id=1420860677)***

Plays timer warning sound

**SfxSharedTimerWarningStop*= \_AudioEvent(name='SfxSharedTimerWarningStop', id=3016301418)***

Stop all active plays of timer warning sound

**Sfx\_Alien\_Invasion\_Ufo*= \_AudioEvent(name='Sfx\_Alien\_Invasion\_Ufo', id=72626837)***

Plays alien invasion sound.

**Sfx\_Alien\_Invasion\_Ufo\_Stop*= \_AudioEvent(name='Sfx\_Alien\_Invasion\_Ufo\_Stop', id=2912905250)***

Stops all active plays of the alien invasion sound.

**Sfx\_Brick\_Bash*= \_AudioEvent(name='Sfx\_Brick\_Bash', id=773532685)***

Plays brick bash sound.

**Sfx\_Brick\_Bash\_Stop*= \_AudioEvent(name='Sfx\_Brick\_Bash\_Stop', id=1455749214)***

Stops all active plays of the brick bash sound.

**Sfx\_Constellation\_Star*= \_AudioEvent(name='Sfx\_Constellation\_Star', id=1881919109)***

Plays constellation star sound.

**Sfx\_Constellation\_Star\_Stop*= \_AudioEvent(name='Sfx\_Constellation\_Star\_Stop', id=579345806)***

Stops all active plays of the constellation star sound.

**Sfx\_Egg\_Decorating\_Crack*= \_AudioEvent(name='Sfx\_Egg\_Decorating\_Crack', id=1294616912)***

Plays egg cracking sound.

**Sfx\_Egg\_Decorating\_Crack\_Stop*= \_AudioEvent(name='Sfx\_Egg\_Decorating\_Crack\_Stop', id=3818701437)***

Stops all active plays of the egg cracking sound.

**Sfx\_Fidget\_Spinner\_Loop\_Play*= \_AudioEvent(name='Sfx\_Fidget\_Spinner\_Loop\_Play', id=711055678)***

Plays fidget spinner loop.

**Sfx\_Fidget\_Spinner\_Loop\_Stop*= \_AudioEvent(name='Sfx\_Fidget\_Spinner\_Loop\_Stop', id=2909245734)***

Stops all the fidget spinned looping sound.

**Sfx\_Fidget\_Spinner\_Start*= \_AudioEvent(name='Sfx\_Fidget\_Spinner\_Start', id=2368469177)***

Plays fidget spinner sound.

**Sfx\_Fidget\_Spinner\_Start\_Stop*= \_AudioEvent(name='Sfx\_Fidget\_Spinner\_Start\_Stop', id=2997419718)***

Stops all active plays of the fidget spinner sound.

**Sfx\_Flappy\_Increase*= \_AudioEvent(name='Sfx\_Flappy\_Increase', id=3457937792)***

Plays flappy sound.

**Sfx\_Flappy\_Increase\_Stop*= \_AudioEvent(name='Sfx\_Flappy\_Increase\_Stop', id=437871341)***

Stops all active plays of the flappy sound.

**Sfx\_Morse\_Code\_Dash*= \_AudioEvent(name='Sfx\_Morse\_Code\_Dash', id=3631133922)***

Plays morse code dash sound.

**Sfx\_Morse\_Code\_Dash\_Stop*= \_AudioEvent(name='Sfx\_Morse\_Code\_Dash\_Stop', id=4156175391)***

Stops all active plays of the morse code dash sound.

**Sfx\_Morse\_Code\_Dot*= \_AudioEvent(name='Sfx\_Morse\_Code\_Dot', id=4212384135)***

Plays morse code dot sound.

**Sfx\_Morse\_Code\_Dot\_Stop*= \_AudioEvent(name='Sfx\_Morse\_Code\_Dot\_Stop', id=850860868)***

Stops all active plays of the morse code dot sound.

**Sfx\_Morse\_Code\_Silent*= \_AudioEvent(name='Sfx\_Morse\_Code\_Silent', id=1002265509)***

Plays morse code silent sound.

**Sfx\_Morse\_Code\_Silent\_Stop*= \_AudioEvent(name='Sfx\_Morse\_Code\_Silent\_Stop', id=620507378)***

Stops all active plays of the morse code silent sound.

**Sfx\_Paddle\_Ball\_Bounce*= \_AudioEvent(name='Sfx\_Paddle\_Ball\_Bounce', id=1845849602)***

Plays paddle ball bounce sound.

**Sfx\_Paddle\_Ball\_Bounce\_Stop*= \_AudioEvent(name='Sfx\_Paddle\_Ball\_Bounce\_Stop', id=3192082383)***

Stops all active plays of the paddle ball bounce sound.

**Sfx\_Pot\_O\_Gold\_Blip\_Level1*= \_AudioEvent(name='Sfx\_Pot\_O\_Gold\_Blip\_Level1', id=3533103863)***

Plays the first pot of gold sound sound.

**Sfx\_Pot\_O\_Gold\_Blip\_Level1\_Stop*= \_AudioEvent(name='Sfx\_Pot\_O\_Gold\_Blip\_Level1\_Stop', id=970660780)***

Stops all active plays of the first pot of gold blip sound.

**Sfx\_Pot\_O\_Gold\_Blip\_Level2*= \_AudioEvent(name='Sfx\_Pot\_O\_Gold\_Blip\_Level2', id=3533103860)***

Plays the second pot of gold sound sound.

**Sfx\_Pot\_O\_Gold\_Blip\_Level2\_Stop*= \_AudioEvent(name='Sfx\_Pot\_O\_Gold\_Blip\_Level2\_Stop', id=3985940521)***

Stops all active plays of the second pot of gold blip sound.

**Sfx\_Pot\_O\_Gold\_Blip\_Level3*= \_AudioEvent(name='Sfx\_Pot\_O\_Gold\_Blip\_Level3', id=3533103861)***

Plays the third pot of gold sound sound.

**Sfx\_Pot\_O\_Gold\_Blip\_Level3\_Stop*= \_AudioEvent(name='Sfx\_Pot\_O\_Gold\_Blip\_Level3\_Stop', id=2482812534)***

Stops all active plays of the third pot of gold blip sound.

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html)

# **cozmo.behavior**

Behaviors represent a task that Cozmo may perform for an indefinite amount of time.

For example, the “LookAroundInPlace” behavior causes Cozmo to start looking around him (without driving), which will cause events such as [**cozmo.objects.EvtObjectObserved**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectObserved) to be generated as he comes across objects.

Behaviors must be explicitly stopped before having the robot do something else (for example, pick up the object he just observed).

Behaviors are started by a call to [**cozmo.robot.Robot.start\_behavior()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.start_behavior), which returns a [**Behavior**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.Behavior) object. Calling the [**stop()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.Behavior.stop) method on that object terminate the behavior.

The [**BehaviorTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.BehaviorTypes) class in this module holds a list of all available behaviors.

**Classes**

| [**Behavior**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.Behavior)(robot, behavior\_type[, is\_active]) | A Behavior instance describes a behavior the robot is currently performing. |
| --- | --- |
| [**BehaviorTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.BehaviorTypes) | Defines all executable robot behaviors. |
| [**EvtBehaviorRequested**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.EvtBehaviorRequested)(\*\*kwargs) | Triggered when a behavior is requested to start. |
| [**EvtBehaviorStarted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.EvtBehaviorStarted)(\*\*kwargs) | Triggered when a behavior starts running on the robot. |
| [**EvtBehaviorStopped**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.EvtBehaviorStopped)(\*\*kwargs) | Triggered when a behavior stops. |

**cozmo.behavior.BEHAVIOR\_IDLE*= 'behavior\_idle'***

Behavior idle state (not requested to run)

| **Type:** | string |
| --- | --- |

**cozmo.behavior.BEHAVIOR\_REQUESTED*= 'behavior\_requested'***

Behavior requested state (waiting for engine to start it)

| **Type:** | string |
| --- | --- |

**cozmo.behavior.BEHAVIOR\_RUNNING*= 'behavior\_running'***

Behavior running state

| **Type:** | string |
| --- | --- |

**cozmo.behavior.BEHAVIOR\_STOPPED*= 'behavior\_stopped'***

Behavior stopped state

| **Type:** | string |
| --- | --- |

***class*cozmo.behavior.EvtBehaviorRequested(*\*\*kwargs*)**

Triggered when a behavior is requested to start.

**behavior*= 'The Behavior object'***

**behavior\_type\_name*= 'The behavior type name - equivalent to behavior.type.name'***

***class*cozmo.behavior.EvtBehaviorStarted(*\*\*kwargs*)**

Triggered when a behavior starts running on the robot.

**behavior*= 'The Behavior object'***

**behavior\_type\_name*= 'The behavior type name - equivalent to behavior.type.name'***

***class*cozmo.behavior.EvtBehaviorStopped(*\*\*kwargs*)**

Triggered when a behavior stops.

**behavior*= 'The behavior type object'***

**behavior\_type\_name*= 'The behavior type name - equivalent to behavior.type.name'***

***class*cozmo.behavior.Behavior(*robot*, *behavior\_type*, *is\_active=False*, *\*\*kw*)**

A Behavior instance describes a behavior the robot is currently performing.

Returned by [**cozmo.robot.Robot.start\_behavior()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.start_behavior).

**is\_active**

True if the behavior is currently active and may run on the robot.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_running**

True if the behavior is currently running on the robot.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**stop()**

Requests that the robot stop performing the behavior.

Has no effect if the behavior is not presently active.

**wait\_for\_completed(*timeout=None*)**

Waits for the behavior to complete.

| **Parameters:** | **timeout** ([*int*](https://docs.python.org/3.5/library/functions.html#int) *or* [*None*](https://docs.python.org/3.5/library/constants.html#None)) – Maximum time in seconds to wait for the event. Pass None to wait indefinitely. |
| --- | --- |
| **Raises:** | [**asyncio.TimeoutError**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.TimeoutError) |

**wait\_for\_started(*timeout=5*)**

Waits for the behavior to start.

| **Parameters:** | **timeout** ([*int*](https://docs.python.org/3.5/library/functions.html#int) *or* [*None*](https://docs.python.org/3.5/library/constants.html#None)) – Maximum time in seconds to wait for the event. Pass None to wait indefinitely. If a behavior can run it should usually start within ~0.2 seconds. |
| --- | --- |
| **Raises:** | [**asyncio.TimeoutError**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.TimeoutError) |

***class*cozmo.behavior.BehaviorTypes**

Defines all executable robot behaviors.

For use with [**cozmo.robot.Robot.start\_behavior()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.start_behavior).

**FindFaces*= \_BehaviorType(name='FindFaces', id=1)***

Turn and move head, but don’t drive, with Cozmo’s head angled upwards where faces are likely to be.

**KnockOverCubes*= \_BehaviorType(name='KnockOverCubes', id=4)***

Knock over a stack of cubes.

**LookAroundInPlace*= \_BehaviorType(name='LookAroundInPlace', id=6)***

Turn and move head, but don’t drive, to see what is around Cozmo.

**PounceOnMotion*= \_BehaviorType(name='PounceOnMotion', id=7)***

Tries to “pounce” (drive forward and lower lift) when it detects nearby motion on the ground plane.

**RollBlock*= \_BehaviorType(name='RollBlock', id=8)***

Roll a block, regardless of orientation.

**StackBlocks*= \_BehaviorType(name='StackBlocks', id=9)***

Pickup one block, and stack it onto another block.

**\_EnrollFace*= \_BehaviorType(name='EnrollFace', id=0)***

# **cozmo.camera**

Support for Cozmo’s camera.

Cozmo has a built-in camera which he uses to observe the world around him.

The [**Camera**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera) class defined in this module is made available as **cozmo.world.World.camera** and can be used to enable/disable image sending, enable/disable color images, modify various camera settings, read the robot’s unique camera calibration settings, as well as observe raw unprocessed images being sent by the robot.

Generally, however, it is more useful to observe [**cozmo.world.EvtNewCameraImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.EvtNewCameraImage) events, which include the raw camera images along with annotated images, which can illustrate objects the robot has identified.

**Classes**

| [**Camera**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera)(robot, \*\*kw) | Represents Cozmo’s camera. |
| --- | --- |
| [**CameraConfig**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.CameraConfig)(focal\_length\_x, focal\_length\_y, …) | The fixed properties for Cozmo’s Camera |
| [**EvtNewRawCameraImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.EvtNewRawCameraImage)(\*\*kwargs) | Dispatched when a new raw image is received from the robot’s camera. |
| [**EvtRobotObservedMotion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.EvtRobotObservedMotion)(\*\*kwargs) | Generated when the robot observes motion. |

***class*cozmo.camera.EvtNewRawCameraImage(*\*\*kwargs*)**

Dispatched when a new raw image is received from the robot’s camera.

See also [**EvtNewCameraImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.EvtNewCameraImage) which provides access to both the raw image and a scaled and annotated version.

**image*= 'A PIL.Image.Image object'***

***class*cozmo.camera.EvtRobotObservedMotion(*\*\*kwargs*)**

Generated when the robot observes motion.

**ground\_area*= 'Area of the supporting region for the point, as a fraction of the ground ROI'***

**ground\_pos*= 'Approximate coordinates of observed motion on the ground, relative to robot, in mm'***

**has\_left\_movement*= "Movement detected near the left edge of the robot's view"***

**has\_right\_movement*= "Movement detected near the right edge of the robot's view"***

**has\_top\_movement*= "Movement detected near the top of the robot's view"***

**img\_area*= 'Area of the supporting region for the point, as a fraction of the image'***

**img\_pos*= 'Centroid of observed motion, relative to top-left corner'***

**left\_img\_pos*= 'Coordinates of the centroid of observed motion, relative to top-left corner'***

**right\_img\_pos*= 'Coordinates of the centroid of observed motion, relative to top-left corner'***

**timestamp*= 'Robot timestamp for when movement was observed'***

**top\_img\_pos*= 'Coordinates of the centroid of observed motion, relative to top-left corner'***

***class*cozmo.camera.CameraConfig(*focal\_length\_x: float*, *focal\_length\_y: float*, *center\_x: float*, *center\_y: float*, *fov\_x\_degrees: float*, *fov\_y\_degrees: float*, *min\_exposure\_time\_ms: int*, *max\_exposure\_time\_ms: int*, *min\_gain: float*, *max\_gain: float*)**

The fixed properties for Cozmo’s Camera

A full 3x3 calibration matrix for doing 3D reasoning based on the camera images would look like:

| focal\_length.x | 0 | center.x |
| --- | --- | --- |
| 0 | focal\_length.y | center.y |
| 0 | 0 | 1 |

**center**

The focal center of the camera.

This is the position of the optical center of projection within the image. It will be close to the center of the image, but adjusted based on the calibration of the lens at the factory. It is in floating point pixel values e.g. <155.11, 111.40>.

| **Type:** | [**cozmo.util.Vector2**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector2) |
| --- | --- |

**focal\_length**

The focal length of the camera.

This is focal length combined with pixel skew (as the pixels aren’t perfectly square), so there are subtly different values for x and y. It is in floating point pixel values e.g. <288.87, 288.36>.

| **Type:** | [**cozmo.util.Vector2**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector2) |
| --- | --- |

**fov\_x**

The x (horizontal) field of view.

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**fov\_y**

The y (vertical) field of view.

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**max\_exposure\_time\_ms**

The maximum supported exposure time in milliseconds.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**max\_gain**

The maximum supported camera gain.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**min\_exposure\_time\_ms**

The minimum supported exposure time in milliseconds.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**min\_gain**

The minimum supported camera gain.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.camera.Camera(*robot*, *\*\*kw*)**

Represents Cozmo’s camera.

The Camera object receives images from Cozmo’s camera and emits EvtNewRawCameraImage events.

The [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) instance observes the camera and provides more useful methods for accessing the camera images.

**Important**

The camera will not receive any image data unless you explicitly enable it by setting [**Camera.image\_stream\_enabled**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera.image_stream_enabled) to **True**

**color\_image\_enabled**

Set to true to receive color images from the robot.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**config**

The read-only config/calibration for the camera

| **Type:** | [**cozmo.camera.CameraConfig**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.CameraConfig) |
| --- | --- |

**enable\_auto\_exposure(*enable\_auto\_exposure=True*)**

Enable auto exposure on Cozmo’s Camera.

Enable auto exposure on Cozmo’s camera to constantly update the exposure time and gain values based on the recent images. This is the default mode when any SDK program starts.

| **Parameters:** | **enable\_auto\_exposure** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – whether the camera should automcatically adjust exposure |
| --- | --- |

**exposure\_ms**

The current camera exposure setting in milliseconds.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**gain**

The current camera gain setting.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**image\_stream\_enabled**

Set to true to receive camera images from the robot.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_auto\_exposure\_enabled**

True if auto exposure is currently enabled

If auto exposure is enabled the gain and exposure\_ms values will constantly be updated by Cozmo.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**set\_manual\_exposure(*exposure\_ms*, *gain*)**

Set manual exposure values for Cozmo’s Camera.

Disable auto exposure on Cozmo’s camera and force the specified exposure time and gain values.

| **Parameters:** | * **exposure\_ms** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The desired exposure time in milliseconds. Must be within the robot’s [**config**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera.config) exposure range from [**min\_exposure\_time\_ms**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.CameraConfig.min_exposure_time_ms) to [**max\_exposure\_time\_ms**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.CameraConfig.max_exposure_time_ms) * **gain** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The desired gain value. Must be within the robot’s **camera\_config** gain range from [**min\_gain**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.CameraConfig.min_gain) to [**max\_gain**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.CameraConfig.max_gain) |
| --- | --- |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) if supplied an out-of-range exposure or gain. |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html)

# **cozmo.conn**

Engine connection.

The SDK operates by connecting to the Cozmo “engine” - typically the Cozmo app that runs on an iOS or Android device.

The engine is responsible for much of the work that Cozmo does, including image recognition, path planning, behaviors and animation handling, etc.

The [**cozmo.run**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#module-cozmo.run) module takes care of opening a connection over a USB connection to a device, but the [**CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) class defined in this module does the work of relaying messages to and from the engine and dispatching them to the [**cozmo.robot.Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) instance.

**Classes**

| [**CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection)(\*a, \*\*kw) | Manages the connection to the Cozmo app to communicate with the core engine. |
| --- | --- |
| **EvtConnected**(\*\*kwargs) | Triggered when the initial connection to the device has been established. |
| **EvtConnectionClosed**(\*\*kwargs) | Triggered when the connection to the controlling device is closed. |
| [**EvtRobotFound**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.EvtRobotFound)(\*\*kwargs) | Triggered when a Cozmo robot is detected, but before he’s initialized. |

***class*cozmo.conn.EvtRobotFound(*\*\*kwargs*)**

Triggered when a Cozmo robot is detected, but before he’s initialized.

[**cozmo.robot.EvtRobotReady**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.EvtRobotReady) is dispatched when the robot is fully initialized.

**robot*= 'The Cozmo object for the robot'***

***class*cozmo.conn.CozmoConnection(*\*a*, *\*\*kw*)**

Manages the connection to the Cozmo app to communicate with the core engine.

An instance of this class is passed to functions used with [**cozmo.run.connect()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect). At the point the function is executed, the connection is already established and verified, and the **EvtConnected** has already been sent.

However, after the initial connection is established, programs will usually want to call [**wait\_for\_robot()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection.wait_for_robot) to wait for an actual Cozmo robot to be detected and initialized before doing useful work.

**abort(*exc*)**

Abort the connection to the device.

**anim\_names*= None***

An [**cozmo.anim.AnimationNames**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.AnimationNames) object that references all available animation names

**anim\_names\_factory*= functools.partial(<class 'cozmo.anim.AnimationNames'>, loop=None)***

The factory function that returns an [**cozmo.anim.AnimationNames**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.AnimationNames) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**clad\_decode\_union**

alias of **cozmoclad.clad.externalInterface.messageEngineToGame.MessageEngineToGame**

**clad\_encode\_union**

alias of **cozmoclad.clad.externalInterface.messageGameToEngine.MessageGameToEngine**

**connection\_lost(*exc*)**

Called when the connection is lost or closed.

The argument is an exception object or None (the latter meaning a regular EOF is received or the connection was aborted or closed).

**connection\_made(*transport*)**

Called when a connection is made.

The argument is the transport representing the pipe connection. To receive data, wait for data\_received() calls. When the connection is closed, connection\_lost() is called.

**device\_info*= None***

A dict containing information about the device the connection is using.

**is\_connected**

True if currently connected to the remote engine.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**msg\_received(*msg*)**

Receives low level communication messages from the engine.

**robot\_factory*= functools.partial(<class 'cozmo.robot.Robot'>, loop=None)***

The factory function that returns a [**cozmo.robot.Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**shutdown()**

Close the connection to the device.

**wait\_for\_robot(*timeout=5*)**

Wait for a Cozmo robot to connect and complete initialization.

| **Parameters:** | **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Maximum length of time to wait for a robot to be ready in seconds. |
| --- | --- |
| **Returns:** | A [**cozmo.robot.Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) instance that’s ready to use. |
| **Raises:** | [**asyncio.TimeoutError**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.TimeoutError) if there’s no response from the robot. |

# **cozmo.event**

Event dispatch system.

The SDK is based around the dispatch and observation of events. Objects inheriting from the [**Dispatcher**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher) generate and dispatch events as the state of the robot and its world are updated.

For example the [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube) class generates an [**EvtObjectTapped**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectTapped) event anytime the cube the object represents is tapped.

The event can be observed in a number of different ways:

1. By calling the [**wait\_for()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.wait_for) method on the object to observe. This will wait until the specific event has been sent to that object and return the generated event.
2. By calling [**add\_event\_handler()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.add_event_handler) on the object to observe, which will cause the supplied function to be called every time the specified event occurs (use the [**oneshot()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.oneshot) decorator to only have the handler called once)
3. By sub-classing a type and implementing a receiver method. For example, subclass the [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube) type and implement evt\_object\_tapped. Note that the factory attribute would need to be updated on the generating class for your type to be used by the SDK. For example, [**light\_cube\_factory**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.light_cube_factory) in this example.
4. By subclassing a type and implementing a default receiver method. Events not dispatched to an explicit receiver method are dispatched to recv\_default\_handler.

Events are dispatched to a target object (by calling **dispatch\_event()** on the receiving object). In line with the above, upon receiving an event, the object will:

1. Dispatch the event to any handlers which have explicitly registered interest in the event (or a superclass of the event) via [**add\_event\_handler()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.add_event_handler) or via [**Dispatcher.wait\_for()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.wait_for)
2. Dispatch the event to any “children” of the object (see below)
3. Dispatch the event to method handlers on the receiving object, or the recv\_default\_handler if it has no matching handler
4. Dispatch the event to the parent of the object (if any), and in turn onto the parent’s parents.

Any handler may raise a [**StopPropogation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.StopPropogation) exception to prevent the event reaching any subsequent handlers (but generally should have no need to do so).

Child objects receive all events that are sent to the originating object (which may have multiple children).

Originating objects may have one parent object, which receives all events sent to its child.

For example, [**cozmo.robot.Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) creates a [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) object and sets itself as a parent and the World as the child; both receive events sent to the other.

The World class creates individual [**cozmo.objects.ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject) objects as they are discovered and makes itself a parent, so as to receive all events sent to the child. However, it does not make those ObservableObject objects children for the sake of message dispatch as they only need to receive a small subset of messages the World object receives.

**Functions**

| [**filter\_handler**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.filter_handler)(event, \*\*filters) | Decorates a handler function or Future to only be called if a filter is matched. |
| --- | --- |
| [**oneshot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.oneshot)(f) | Event handler decorator; causes the handler to only be dispatched to once. |
| [**wait\_for\_first**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.wait_for_first)(\*futures[, …]) | Wait the first of a set of futures to complete. |

**Classes**

| [**Dispatcher**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher)(\*a[, dispatch\_parent, loop]) | Mixin to provide event dispatch handling. |
| --- | --- |
| [**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event)(\*\*kwargs) | An event representing an action that has occurred. |
| [**Filter**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Filter)(event, \*\*filters) | Provides fine-grain filtering of events for dispatch. |
| [**Handler**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Handler) | A Handler is returned by [**Dispatcher.add\_event\_handler()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.add_event_handler) |
| **NullHandler** |  |
| **docstr** |  |

***class*cozmo.event.Event(*\*\*kwargs*)**

An event representing an action that has occurred.

Instances of an Event have attributes set to values passed to the event.

For example, [**cozmo.objects.EvtObjectTapped**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectTapped) defines obj and tap\_count parameters which can be accessed as **evt.obj** and **evt.tap\_count**.

***class*cozmo.event.Dispatcher(*\*a*, *dispatch\_parent=None*, *loop=None*, *\*\*kw*)**

Mixin to provide event dispatch handling.

**add\_event\_handler(*event*, *f*)**

Register an event handler to be notified when this object receives a type of Event.

Expects a subclass of Event as the first argument. If the class has subclasses then the handler will be notified for events of that subclass too. For example, adding a handler for [**EvtActionCompleted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.action.html#cozmo.action.EvtActionCompleted) will cause the handler to also be notified for [**EvtAnimationCompleted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.EvtAnimationCompleted) as it’s a subclass.

Callable handlers (e.g. functions) are called with a first argument containing an Event instance and the remaining keyword arguments set as the event parameters.

For example, **def my\_ontap\_handler(evt, \*, obj, tap\_count, \*\*kwargs)** or **def my\_ontap\_handler(evt, obj=None, tap\_count=None, \*\*kwargs)**

It’s recommended that a **\*\*kwargs** parameter be included in the definition so that future expansion of event parameters do not cause the handler to fail.

Callable handlers may raise an events.StopProgation exception to prevent other handlers listening to the same event from being triggered.

[**asyncio.Future**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.Future) handlers are called with a result set to the event.

| **Parameters:** | * **event** ([**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event)) – A subclass of [**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event) (not an instance of that class) * **f** (*callable*) – A callable or [**asyncio.Future**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.Future) to execute when the event is received |
| --- | --- |
| **Raises:** | [**TypeError**](https://docs.python.org/3.5/library/exceptions.html#TypeError) – An invalid event type was supplied |

**dispatch\_event(*event*, *\*\*kw*)**

Dispatches a single event to registered handlers.

Not generally called from user-facing code.

| **Parameters:** | * **event** ([**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event)) – An class or instance of [**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event) * **kw** ([*dict*](https://docs.python.org/3.5/library/stdtypes.html#dict)) – If a class is passed to event, then the remaining keywords are passed to it to create an instance of the event. |
| --- | --- |
| **Returns:** | A [**asyncio.Task**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.Task) or [**asyncio.Future**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.Future) that will complete once all event handlers have been called. |
| **Raises:** | [**TypeError**](https://docs.python.org/3.5/library/exceptions.html#TypeError) if an invalid event is supplied. |

**remove\_event\_handler(*event*, *f*)**

Remove an event handler for this object.

| **Parameters:** | * **event** ([**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event)) – The event class, or an instance thereof, used with register\_event\_handler. * **f** (callable or [**Handler**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Handler)) – The callable object that was passed as a handler to [**add\_event\_handler()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.add_event_handler), or a [**Handler**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Handler) instance that was returned by [**add\_event\_handler()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.add_event_handler). |
| --- | --- |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) – No matching handler found. |

**wait\_for(*event\_or\_filter*, *timeout=30*)**

Waits for the specified event to be sent to the current object.

| **Parameters:** | * **event\_or\_filter** ([**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event)) – Either a [**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event) class or a [**Filter**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Filter) instance to wait to trigger * **timeout** – Maximum time to wait for the event. Pass None to wait indefinitely. |
| --- | --- |
| **Returns:** | The [**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event) instance that was dispatched |
| **Raises:** | [**asyncio.TimeoutError**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.TimeoutError) |

***class*cozmo.event.Filter(*event*, *\*\*filters*)**

Provides fine-grain filtering of events for dispatch.

See the ::func::filter\_handler method for further details.

***class*cozmo.event.Handler**

A Handler is returned by [**Dispatcher.add\_event\_handler()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.add_event_handler)

The handler can be disabled at any time by calling its [**disable()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Handler.disable) method.

**disable()**

Removes the handler from the object it was originally registered with.

**oneshot**

True if the wrapped handler function will only be called once.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**cozmo.event.oneshot(*f*)**

Event handler decorator; causes the handler to only be dispatched to once.

**cozmo.event.filter\_handler(*event*, *\*\*filters*)**

Decorates a handler function or Future to only be called if a filter is matched.

A handler may apply multiple separate filters; the handlers will be called if any of those filters matches.

For example:

*# Handle only if the anim\_majorwin animation completed*

**@filter\_handler**(cozmo.anim.EvtAnimationCompleted, animation\_name="anim\_majorwin")

*# Handle only when the observed object is a LightCube*

**@filter\_handler**(cozmo.objects.EvtObjectObserved, obj=**lambda** obj: isinstance(cozmo.objects.LightCube))

| **Parameters:** | * **event** ([**Event**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Event)) – The event class to match on * **filters** ([*dict*](https://docs.python.org/3.5/library/stdtypes.html#dict)) – Zero or more event parameters to filter on. Values may be either strings for exact matches, or functions which accept the value as the first argument and return a bool indicating whether the value passes the filter. |
| --- | --- |

**cozmo.event.wait\_for\_first(*\*futures*, *discard\_remaining=True*, *loop=None*)**

Wait the first of a set of futures to complete.

Eg:

event = cozmo.event.wait\_for\_first(

coz.world.wait\_for\_new\_cube(),

playing\_anim.wait\_for(cozmo.anim.EvtAnimationCompleted)

)

If more than one completes during a single event loop run, then if any of those results are not exception, one of them will be selected (at random, as determined by **set.pop**) to be returned, else one of the result exceptions will be raised instead.

| **Parameters:** | * **futures** (list of [**asyncio.Future**](https://docs.python.org/3.5/library/asyncio-task.html#asyncio.Future)) – The futures or coroutines to wait on. * **discard\_remaining** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Cancel or discard the results of the futures that did not return first. * **loop** ([**asyncio.BaseEventLoop**](https://docs.python.org/3.5/library/asyncio-eventloop.html#asyncio.BaseEventLoop)) – The event loop to wait on. |
| --- | --- |
| **Returns:** | The first result, or raised exception |

# **cozmo.exceptions**

SDK-specific exception classes.

**Exceptions**

| [**ActionError**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.ActionError) | Base class for errors that occur with robot actions. |
| --- | --- |
| [**AnimationsNotLoaded**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.AnimationsNotLoaded) | Raised if an attempt is made to play a named animation before animations have been received. |
| [**CannotPlaceObjectsOnThis**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.CannotPlaceObjectsOnThis) | Raised if an attempt is made to place an object on top of an invalid object |
| [**ConnectionAborted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.ConnectionAborted) | Raised if the connection to the device is unexpectedly lost. |
| [**ConnectionCheckFailed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.ConnectionCheckFailed) | Raised if the connection check has failed. |
| [**ConnectionError**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.ConnectionError) | Base class for errors regarding connection to the device. |
| [**CozmoSDKException**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.CozmoSDKException) | Base class of all Cozmo SDK exceptions. |
| [**InvalidOpenGLGlutImplementation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.InvalidOpenGLGlutImplementation) | Raised by opengl viewer if no valid GLUT implementation available. |
| [**NoDevicesFound**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.NoDevicesFound) | Raised if no devices connected running Cozmo in SDK mode |
| [**NotPickupable**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.NotPickupable) | Raised if an attempt is made to pick up or place an object that can’t be picked up by Cozmo |
| [**RobotBusy**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.RobotBusy) | Raised if an attempt is made to perform an action while another action is still running. |
| [**SDKShutdown**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.SDKShutdown) | Raised when the SDK is being shut down |
| [**SDKVersionMismatch**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.SDKVersionMismatch)(message, sdk\_version, …) | Raised if the Cozmo SDK version is not compatible with the software running on the device. |
| [**StopPropogation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.StopPropogation) | Raised by event handlers to prevent further handlers from being triggered. |

***exception*cozmo.exceptions.CozmoSDKException**

Base class of all Cozmo SDK exceptions.

***exception*cozmo.exceptions.SDKShutdown**

Raised when the SDK is being shut down

***exception*cozmo.exceptions.StopPropogation**

Raised by event handlers to prevent further handlers from being triggered.

***exception*cozmo.exceptions.AnimationsNotLoaded**

Raised if an attempt is made to play a named animation before animations have been received.

***exception*cozmo.exceptions.ActionError**

Base class for errors that occur with robot actions.

***exception*cozmo.exceptions.ConnectionError**

Base class for errors regarding connection to the device.

***exception*cozmo.exceptions.ConnectionAborted**

Raised if the connection to the device is unexpectedly lost.

***exception*cozmo.exceptions.ConnectionCheckFailed**

Raised if the connection check has failed.

***exception*cozmo.exceptions.NoDevicesFound**

Raised if no devices connected running Cozmo in SDK mode

***exception*cozmo.exceptions.SDKVersionMismatch(*message*, *sdk\_version*, *sdk\_app\_version*, *app\_version*, *\*args*)**

Raised if the Cozmo SDK version is not compatible with the software running on the device.

**app\_version*= None***

The version of the App that was detected, and is incompatible, in Major.Minor.Patch format.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**sdk\_app\_version*= None***

The version of the App that this SDK is compatible with in Major.Minor.Patch format.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**sdk\_version*= None***

The SDK version number in Major.Minor.Patch format. See [SDK vs. App Versions](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/sdk-versions.html#sdk-versions) for which App version is compatible with each SDK version.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

***exception*cozmo.exceptions.NotPickupable**

Raised if an attempt is made to pick up or place an object that can’t be picked up by Cozmo

***exception*cozmo.exceptions.CannotPlaceObjectsOnThis**

Raised if an attempt is made to place an object on top of an invalid object

***exception*cozmo.exceptions.RobotBusy**

Raised if an attempt is made to perform an action while another action is still running.

***exception*cozmo.exceptions.InvalidOpenGLGlutImplementation**

Raised by opengl viewer if no valid GLUT implementation available.

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html)

# **cozmo.faces**

Face recognition and enrollment.

Cozmo is capable of recognizing human faces, tracking their position and rotation (“pose”) and assigning names to them via an enrollment process.

The [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) object keeps track of faces the robot currently knows about, along with those that are currently visible to the camera.

Each face is assigned a [**Face**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face) object, which generates a number of observable events whenever the face is observed, has its ID updated, is renamed, etc.

Note that these face-specific events are also passed up to the [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) object, so events for all known faces can be observed by adding handlers there.

**Functions**

| [**erase\_all\_enrolled\_faces**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.erase_all_enrolled_faces)(conn) | Erase the enrollment (name) records for all faces. |
| --- | --- |
| [**erase\_enrolled\_face\_by\_id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.erase_enrolled_face_by_id)(conn, face\_id) | Erase the enrollment (name) record for the face with this ID. |
| [**update\_enrolled\_face\_by\_id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.update_enrolled_face_by_id)(conn, face\_id, …) | Update the name enrolled for a given face. |

**Classes**

| [**EvtErasedEnrolledFace**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.EvtErasedEnrolledFace)(\*\*kwargs) | Triggered when a face enrollment is removed (via erase\_enrolled\_face\_by\_id) |
| --- | --- |
| [**EvtFaceAppeared**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.EvtFaceAppeared)(\*\*kwargs) | Triggered whenever a face is first visually identified by a robot. |
| [**EvtFaceDisappeared**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.EvtFaceDisappeared)(\*\*kwargs) | Triggered whenever a face that was previously being observed is no longer visible. |
| [**EvtFaceIdChanged**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.EvtFaceIdChanged)(\*\*kwargs) | Triggered whenever a face has its ID updated in engine. |
| [**EvtFaceObserved**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.EvtFaceObserved)(\*\*kwargs) | Triggered whenever a face is visually identified by the robot. |
| [**EvtFaceRenamed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.EvtFaceRenamed)(\*\*kwargs) | Triggered whenever a face is renamed (via RobotRenamedEnrolledFace) |
| [**Face**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face)(conn, world, robot[, face\_id]) | A single face that Cozmo has detected. |

**cozmo.faces.FACE\_VISIBILITY\_TIMEOUT*= 0.4***

Length of time in seconds to go without receiving an observed event before assuming that Cozmo can no longer see a face.

**cozmo.faces.FACIAL\_EXPRESSION\_UNKNOWN*= 'unknown'***

Facial expression not recognized. Call [**cozmo.robot.Robot.enable\_facial\_expression\_estimation()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.enable_facial_expression_estimation) to enable recognition.

**cozmo.faces.FACIAL\_EXPRESSION\_NEUTRAL*= 'neutral'***

Facial expression neutral

**cozmo.faces.FACIAL\_EXPRESSION\_HAPPY*= 'happy'***

Facial expression happy

**cozmo.faces.FACIAL\_EXPRESSION\_SURPRISED*= 'surprised'***

Facial expression surprised

**cozmo.faces.FACIAL\_EXPRESSION\_ANGRY*= 'angry'***

Facial expression angry

**cozmo.faces.FACIAL\_EXPRESSION\_SAD*= 'sad'***

Facial expression sad

***class*cozmo.faces.EvtErasedEnrolledFace(*\*\*kwargs*)**

Triggered when a face enrollment is removed (via erase\_enrolled\_face\_by\_id)

**face*= 'The Face instance that the enrollment is being erased for'***

**old\_name*= 'The name previously used for this face'***

***class*cozmo.faces.EvtFaceAppeared(*\*\*kwargs*)**

Triggered whenever a face is first visually identified by a robot.

This differs from EvtFaceObserved in that it’s only triggered when a face initially becomes visible. If it disappears for more than FACE\_VISIBILITY\_TIMEOUT seconds and then is seen again, a EvtFaceDisappeared will be dispatched, followed by another EvtFaceAppeared event.

For continuous tracking information about a visible face, see EvtFaceObserved.

**face*= 'The Face instance that was observed'***

**image\_box*= "A comzo.util.ImageBox defining where the face is within Cozmo's camera view"***

**name*= 'The name associated with the face that was observed'***

**pose*= 'The cozmo.util.Pose defining the position and rotation of the face.'***

**updated*= 'A set of field names that have changed'***

***class*cozmo.faces.EvtFaceDisappeared(*\*\*kwargs*)**

Triggered whenever a face that was previously being observed is no longer visible.

**face*= 'The Face instance that is no longer being observed'***

***class*cozmo.faces.EvtFaceIdChanged(*\*\*kwargs*)**

Triggered whenever a face has its ID updated in engine.

Generally occurs when: 1) A tracked but unrecognized face (negative ID) is recognized and receives a positive ID or 2) Face records get merged (on realization that 2 faces are actually the same)

**face*= 'The Face instance that is being given a new id'***

**new\_id*= 'The new ID that will be used for this face'***

**old\_id*= 'The ID previously used for this face'***

***class*cozmo.faces.EvtFaceObserved(*\*\*kwargs*)**

Triggered whenever a face is visually identified by the robot.

A stream of these events are produced while a face is visible to the robot. Each event has an updated image\_box field.

See EvtFaceAppeared if you only want to know when a face first becomes visible.

**face*= 'The Face instance that was observed'***

**image\_box*= "A comzo.util.ImageBox defining where the face is within Cozmo's camera view"***

**name*= 'The name associated with the face that was observed'***

**pose*= 'The cozmo.util.Pose defining the position and rotation of the face.'***

**updated*= 'A set of field names that have changed'***

***class*cozmo.faces.EvtFaceRenamed(*\*\*kwargs*)**

Triggered whenever a face is renamed (via RobotRenamedEnrolledFace)

**face*= 'The Face instance that is being given a new name'***

**new\_name*= 'The new name that will be used for this face'***

**old\_name*= 'The name previously used for this face'***

***class*cozmo.faces.Face(*conn*, *world*, *robot*, *face\_id=None*, *\*\*kw*)**

A single face that Cozmo has detected.

May represent a face that has previously been enrolled, in which case [**name**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face.name) will hold the name that it was enrolled with.

Each Face instance has a [**face\_id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face.face_id) integer - This may change if Cozmo later gets an improved view and makes a different prediction about which face it is looking at.

See parent class [**ObservableElement**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableElement) for additional properties and methods.

**erase\_enrolled\_face()**

Remove the name associated with this face.

Cozmo will no longer remember the name associated with this face between SDK runs.

**expression**

The facial expression Cozmo has recognized on the face.

Will be [**FACIAL\_EXPRESSION\_UNKNOWN**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.FACIAL_EXPRESSION_UNKNOWN) by default if you haven’t called [**cozmo.robot.Robot.enable\_facial\_expression\_estimation()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.enable_facial_expression_estimation) to enable the facial expression estimation. Otherwise it will be equal to one of: [**FACIAL\_EXPRESSION\_NEUTRAL**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.FACIAL_EXPRESSION_NEUTRAL), [**FACIAL\_EXPRESSION\_HAPPY**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.FACIAL_EXPRESSION_HAPPY), [**FACIAL\_EXPRESSION\_SURPRISED**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.FACIAL_EXPRESSION_SURPRISED), [**FACIAL\_EXPRESSION\_ANGRY**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.FACIAL_EXPRESSION_ANGRY), or [**FACIAL\_EXPRESSION\_SAD**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.FACIAL_EXPRESSION_SAD).

| **Type:** | string |
| --- | --- |

**expression\_score**

The score/confidence that [**expression**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face.expression) was correct.

Will be 0 if expression is [**FACIAL\_EXPRESSION\_UNKNOWN**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.FACIAL_EXPRESSION_UNKNOWN) (e.g. if [**cozmo.robot.Robot.enable\_facial\_expression\_estimation()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.enable_facial_expression_estimation) wasn’t called yet). The maximum possible score is 100.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**face\_id**

The internal ID assigned to the face.

This value can only be assigned once as it is static in the engine.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**has\_updated\_face\_id**

True if this face been updated / superseded by a face with a new ID

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**known\_expression**

The known facial expression Cozmo has recognized on the face.

Like [**expression()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face.expression) but returns an empty string for the unknown expression.

| **Type:** | string |
| --- | --- |

**left\_eye**

points representing the outline of the left eye

| **Type:** | sequence of tuples of float (x,y) |
| --- | --- |

**mouth**

points representing the outline of the mouth

| **Type:** | sequence of tuples of float (x,y) |
| --- | --- |

**name**

The name Cozmo has associated with the face in his memory.

This string will be empty if the face is not recognized or enrolled.

| **Type:** | string |
| --- | --- |

**name\_face(*name*)**

Assign a name to this face. Cozmo will remember this name between SDK runs.

| **Parameters:** | **name** (*string*) – The name that will be assigned to this face. Must be a non-empty ASCII string of alphabetic characters only. |
| --- | --- |
| **Returns:** | An instance of [**cozmo.behavior.Behavior**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.Behavior) object |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) if name is invalid. |

**nose**

points representing the outline of the nose

| **Type:** | sequence of tuples of float (x,y) |
| --- | --- |

**rename\_face(*new\_name*)**

Change the name assigned to the face. Cozmo will remember this name between SDK runs.

| **Parameters:** | **new\_name** (*string*) – The new name that will be assigned to this face. Must be a non-empty ASCII string of alphabetic characters only. |
| --- | --- |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) if new\_name is invalid. |

**right\_eye**

points representing the outline of the right eye

| **Type:** | sequence of tuples of float (x,y) |
| --- | --- |

**updated\_face\_id**

The ID for the face that superseded this one (if any, otherwise [**face\_id()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face.face_id))

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**visibility\_timeout*= 0.4***

Length of time in seconds to go without receiving an observed event before assuming that Cozmo can no longer see a face.

**cozmo.faces.erase\_all\_enrolled\_faces(*conn*)**

Erase the enrollment (name) records for all faces.

| **Parameters:** | **conn** ([**CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection)) – The connection to send the message over |
| --- | --- |

**cozmo.faces.erase\_enrolled\_face\_by\_id(*conn*, *face\_id*)**

Erase the enrollment (name) record for the face with this ID.

| **Parameters:** | * **conn** ([**CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection)) – The connection to send the message over * **face\_id** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The ID of the face to erase. |
| --- | --- |

**cozmo.faces.update\_enrolled\_face\_by\_id(*conn*, *face\_id*, *old\_name*, *new\_name*)**

Update the name enrolled for a given face.

| **Parameters:** | * **conn** ([**CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection)) – The connection to send the message over. * **face\_id** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The ID of the face to rename. * **old\_name** (*string*) – The old name of the face (must be correct, otherwise message is ignored). * **new\_name** (*string*) – The new name for the face. |
| --- | --- |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html)

# **cozmo.lights**

Helper routines for dealing with Cozmo’s lights and colors.

**Classes**

| [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color)([int\_color, rgb, name]) | A Color to be used with a Light. |
| --- | --- |
| [**Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)([on\_color, off\_color, on\_period\_ms, …]) | Lights are used with LightCubes and Cozmo’s backpack. |

**cozmo.lights.green*= <cozmo.lights.Color object>***

Green color instance.

| **Type:** | [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color) |
| --- | --- |

**cozmo.lights.red*= <cozmo.lights.Color object>***

Red color instance.

| **Type:** | [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color) |
| --- | --- |

**cozmo.lights.blue*= <cozmo.lights.Color object>***

Blue color instance.

| **Type:** | [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color) |
| --- | --- |

**cozmo.lights.white*= <cozmo.lights.Color object>***

White color instance.

| **Type:** | [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color) |
| --- | --- |

**cozmo.lights.off*= <cozmo.lights.Color object>***

instance representing no color (LEDs off).

| **Type:** | [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color) |
| --- | --- |

**cozmo.lights.green\_light*= <cozmo.lights.Light object>***

A steady green colored LED light.

| **Type:** | [**Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light) |
| --- | --- |

**cozmo.lights.red\_light*= <cozmo.lights.Light object>***

A steady red colored LED light.

| **Type:** | [**Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light) |
| --- | --- |

**cozmo.lights.blue\_light*= <cozmo.lights.Light object>***

A steady blue colored LED light.

| **Type:** | [**Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light) |
| --- | --- |

**cozmo.lights.white\_light*= <cozmo.lights.Light object>***

A steady white colored LED light.

| **Type:** | [**Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light) |
| --- | --- |

**cozmo.lights.off\_light*= <cozmo.lights.Light object>***

A steady off (non-illuminated LED light).

| **Type:** | [**Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light) |
| --- | --- |

***class*cozmo.lights.Color(*int\_color=None*, *rgb=None*, *name=None*)**

A Color to be used with a Light.

Either int\_color or rgb may be used to specify the actual color. Any alpha components (from int\_color) are ignored - all colors are fully opaque.

| **Parameters:** | * **int\_color** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – A 32 bit value holding the binary RGBA value (where A is ignored and forced to be fully opaque). * **rgb** ([*tuple*](https://docs.python.org/3.5/library/stdtypes.html#tuple)) – A tuple holding the integer values from 0-255 for (red, green, blue) * **name** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – A name to assign to this color |
| --- | --- |

**int\_color**

The encoded integer value of the color.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

***class*cozmo.lights.Light(*on\_color=<cozmo.lights.Color object>*, *off\_color=<cozmo.lights.Color object>*, *on\_period\_ms=250*, *off\_period\_ms=0*, *transition\_on\_period\_ms=0*, *transition\_off\_period\_ms=0*)**

Lights are used with LightCubes and Cozmo’s backpack.

Lights may either be “on” or “off”, though in practice any colors may be assigned to either state (including no color/light).

**flash(*on\_period\_ms=250*, *off\_period\_ms=250*, *off\_color=<cozmo.lights.Color object>*)**

Convenience function to make a flashing version of an existing Light instance.

| **Parameters:** | * **on\_period\_ms** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The number of milliseconds the light should be “on” for for each cycle. * **off\_period\_ms** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The number of milliseconds the light should be “off” for for each cycle. * **off\_color** ([**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color)) – The color to flash to for the off state. |
| --- | --- |
| **Returns:** | [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color) instance. |

**off\_color**

The Color shown when the light is off.

| **Type:** | [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color) |
| --- | --- |

**off\_period\_ms**

The number of milliseconds the light should be “off” for for each cycle.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**on\_color**

The Color shown when the light is on.

| **Type:** | [**Color**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Color) |
| --- | --- |

**on\_period\_ms**

The number of milliseconds the light should be “on” for for each cycle.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**transition\_off\_period\_ms**

The number of milliseconds to take to transition the light to the off color.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**transition\_on\_period\_ms**

The number of milliseconds to take to transition the light to the on color.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html)

# **cozmo.nav\_memory\_map**

A 2D navigation memory map of the world around Cozmo.

Cozmo builds a memory map of the navigable world around him as he drives around. This is mostly based on where objects are seen (the cubes, charger, and any custom objects), and also includes where Cozmo detects cliffs/drops, and visible edges (e.g. sudden changes in color).

This differs from a standard occupancy map in that it doesn’t deal with probabilities of occupancy, but instead encodes what type of content is there.

To use the map you must first call [**cozmo.world.World.request\_nav\_memory\_map()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.request_nav_memory_map) with a positive frequency so that the data is streamed to the SDK.

**Classes**

| [**EvtNewNavMemoryMap**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.EvtNewNavMemoryMap)(\*\*kwargs) | Dispatched when a new memory map is received. |
| --- | --- |
| [**NavMemoryMapGrid**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGrid)(origin\_id, root\_depth, …) | A navigation memory map, stored as a quad-tree. |
| [**NavMemoryMapGridNode**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGridNode)(depth, size, center, parent) | A node in a [**NavMemoryMapGrid**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGrid). |
| [**NodeContentTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NodeContentTypes) | The content types for a [**NavMemoryMapGridNode**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGridNode). |

***class*cozmo.nav\_memory\_map.EvtNewNavMemoryMap(*\*\*kwargs*)**

Dispatched when a new memory map is received.

**nav\_memory\_map*= 'A NavMemoryMapGrid object'***

***class*cozmo.nav\_memory\_map.NavMemoryMapGrid(*origin\_id*, *root\_depth*, *root\_size*, *root\_center\_x*, *root\_center\_y*)**

A navigation memory map, stored as a quad-tree.

**center**

The center of this map.

| **Type:** | [**Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3) |
| --- | --- |

**contains\_point(*x*, *y*)**

Test if the map contains the given x,y coordinates.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – x coordinate for the point * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – y coordinate for the point |
| --- | --- |
| **Returns:** | True if the map contains the point, False otherwise. |
| **Return type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |

**get\_content(*x*, *y*)**

Get the map’s content at the given x,y coordinates.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – x coordinate for the point * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – y coordinate for the point |
| --- | --- |
| **Returns:** | The content included at that point. Will be [**NodeContentTypes.Unknown**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NodeContentTypes.Unknown) if the point is outside of the map. |
| **Return type:** | **\_NodeContentType** |

**get\_node(*x*, *y*)**

Get the node at the given x,y coordinates.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – x coordinate for the point * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – y coordinate for the point |
| --- | --- |
| **Returns:** | The smallest node that includes the point. Will be **None** if the point is outside of the map. |
| **Return type:** | [**NavMemoryMapGridNode**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGridNode) |

**origin\_id*= None***

The origin ID for the map. Only maps and [**Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose) objects of the same origin ID are in the same coordinate frame and can therefore be compared.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**root\_node**

The root node for the grid, contains all other nodes.

| **Type:** | [**NavMemoryMapGridNode**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGridNode) |
| --- | --- |

**size**

The size (width or length) of the square grid.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.nav\_memory\_map.NavMemoryMapGridNode(*depth*, *size*, *center*, *parent*)**

A node in a [**NavMemoryMapGrid**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGrid).

Leaf nodes contain content, all other nodes are split into 4 equally sized children.

Child node indices are stored in the following X,Y orientation:

| ^ | 2 | 0 |
| --- | --- | --- |
| Y | 3 | 1 |
|  | X-> |  |

**center*= None***

The center of this node.

| **Type:** | [**Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3) |
| --- | --- |

**children*= None***

**None** for leaf nodes, a list of 4 child nodes otherwise.

| **Type:** | list of [**NavMemoryMapGridNode**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGridNode) |
| --- | --- |

**contains\_point(*x*, *y*)**

Test if the node contains the given x,y coordinates.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – x coordinate for the point * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – y coordinate for the point |
| --- | --- |
| **Returns:** | True if the node contains the point, False otherwise. |
| **Return type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |

**content*= None***

The content type in this node. Only leaf nodes have content, this is **None** for all other nodes.

| **Type:** | An attribute of [**NodeContentTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NodeContentTypes) |
| --- | --- |

**depth*= None***

The depth of this node. I.e. how far down the quad-tree is it.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**get\_content(*x*, *y*)**

Get the node’s content at the given x,y coordinates.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – x coordinate for the point * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – y coordinate for the point |
| --- | --- |
| **Returns:** | The content included at that point. Will be [**NodeContentTypes.Unknown**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NodeContentTypes.Unknown) if the point is outside of the map. |
| **Return type:** | **\_NodeContentType** |

**get\_node(*x*, *y*)**

Get the node at the given x,y coordinates.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – x coordinate for the point * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – y coordinate for the point |
| --- | --- |
| **Returns:** | The smallest node that includes the point. Will be **None** if the point is outside of the map. |
| **Return type:** | [**NavMemoryMapGridNode**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGridNode) |

**parent*= None***

The parent of this node. Is **None** for the root node.

| **Type:** | [**NavMemoryMapGridNode**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGridNode) |
| --- | --- |

**size*= None***

The size (width or length) of this square node.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.nav\_memory\_map.NodeContentTypes**

The content types for a [**NavMemoryMapGridNode**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGridNode).

**ClearOfCliff*= \_NodeContentType(name='ClearOfCliff', id=2)***

The node is clear of any cliffs (a sharp drop) or obstacles.

**ClearOfObstacle*= \_NodeContentType(name='ClearOfObstacle', id=1)***

The node is clear of obstacles, because Cozmo has seen objects on the other side, but it might contain a cliff. The node will be marked as either [**Cliff**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NodeContentTypes.Cliff) or [**ClearOfCliff**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NodeContentTypes.ClearOfCliff) once Cozmo has driven there.

**Cliff*= \_NodeContentType(name='Cliff', id=6)***

The node contains a cliff (a sharp drop).

**ObstacleCharger*= \_NodeContentType(name='ObstacleCharger', id=4)***

The node contains a [**Charger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.Charger).

**ObstacleCube*= \_NodeContentType(name='ObstacleCube', id=3)***

The node contains a [**LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube).

**Unknown*= \_NodeContentType(name='Unknown', id=0)***

The contents of the node is unknown.

**VisionBorder*= \_NodeContentType(name='VisionBorder', id=7)***

The node contains a visible edge (based on the camera feed).

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html)

# **cozmo.objects**

Object and Power Cube recognition.

Cozmo can recognize and track a number of different types of objects.

These objects may be visible (currently observed by the robot’s camera) and tappable (in the case of the Power Cubes that ship with the robot).

Power Cubes are known as a [**LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube) by the SDK. Each cube has controllable lights, and sensors that can determine when its being moved or tapped.

Objects can emit several events such as [**EvtObjectObserved**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectObserved) when the robot sees (or continues to see) the object with its camera, or [**EvtObjectTapped**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectTapped) if a power cube is tapped by a player. You can either observe the object’s instance directly, or capture all such events for all objects by observing them on [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) instead.

All observable objects have a marker attached to them, which allows Cozmo to recognize the object and it’s position and rotation(“pose”). You can attach markers to your own objects for Cozmo to recognize by printing them out from the online documentation. They will be detected as [**CustomObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObject) instances.

**Classes**

| [**Charger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.Charger)(\*a, \*\*kw) | Cozmo’s charger object, which the robot can observe and drive toward. |
| --- | --- |
| [**CustomObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObject)(conn, world, object\_type, …) | An object defined by the SDK. |
| [**CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers) | Defines all available custom object markers. |
| [**CustomObjectTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectTypes) | Defines all available custom object types. |
| [**EvtObjectAppeared**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectAppeared)(\*\*kwargs) | Triggered whenever an object is first visually identified by a robot. |
| [**EvtObjectConnectChanged**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectConnectChanged)(\*\*kwargs) | Triggered when an active object has connected or disconnected from the robot. |
| [**EvtObjectConnected**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectConnected)(\*\*kwargs) | Triggered when the engine reports that an object is connected (i.e. |
| [**EvtObjectDisappeared**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectDisappeared)(\*\*kwargs) | Triggered whenever an object that was previously being observed is no longer visible. |
| [**EvtObjectLocated**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectLocated)(\*\*kwargs) | Triggered when the engine reports that an object is located (i.e. |
| [**EvtObjectMoving**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectMoving)(\*\*kwargs) | Triggered when an active object is currently moving. |
| [**EvtObjectMovingStarted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectMovingStarted)(\*\*kwargs) | Triggered when an active object starts moving. |
| [**EvtObjectMovingStopped**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectMovingStopped)(\*\*kwargs) | Triggered when an active object stops moving. |
| [**EvtObjectObserved**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectObserved)(\*\*kwargs) | Triggered whenever an object is visually identified by the robot. |
| [**EvtObjectTapped**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectTapped)(\*\*kwargs) | Triggered when an active object is tapped. |
| [**FixedCustomObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.FixedCustomObject)(pose, x\_size\_mm, …) | A fixed object defined by the SDK. |
| [**LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)(cube\_id, \*a, \*\*kw) | A light cube object has four LEDs that Cozmo can actively manipulate and communicate with. |
| [**ObservableElement**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableElement)(conn, world, robot, \*\*kw) | The base type for anything Cozmo can see. |
| [**ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject)(conn, world[, object\_id]) | The base type for objects in Cozmo’s world. |

**cozmo.objects.LightCube1Id*= 1***

LightCube1Id’s markers look a bit like a paperclip

**cozmo.objects.LightCube2Id*= 2***

LightCube2Id’s markers look a bit like a lamp (or a heart)

**cozmo.objects.LightCube3Id*= 3***

LightCube3Id’s markers look a bit like the letters ‘ab’ over ‘T’

**cozmo.objects.LightCubeIDs*= [1, 2, 3]***

An ordered list of the 3 light cube IDs for convenience

**cozmo.objects.OBJECT\_VISIBILITY\_TIMEOUT*= 0.4***

Length of time in seconds to go without receiving an observed event before assuming that Cozmo can no longer see an object.

***class*cozmo.objects.EvtObjectAppeared(*\*\*kwargs*)**

Triggered whenever an object is first visually identified by a robot.

This differs from EvtObjectObserved in that it’s only triggered when an object initially becomes visible. If it disappears for more than OBJECT\_VISIBILITY\_TIMEOUT seconds and then is seen again, a EvtObjectDisappeared will be dispatched, followed by another EvtObjectAppeared event.

For continuous tracking information about a visible object, see EvtObjectObserved.

**image\_box*= "A comzo.util.ImageBox defining where the object is within Cozmo's camera view"***

**obj*= 'The object that was observed'***

**pose*= 'The cozmo.util.Pose defining the position and rotation of the object'***

**updated*= 'A set of field names that have changed'***

***class*cozmo.objects.EvtObjectConnectChanged(*\*\*kwargs*)**

Triggered when an active object has connected or disconnected from the robot.

**connected*= 'True if the object connected, False if it disconnected'***

**obj*= 'The object that connected or disconnected'***

***class*cozmo.objects.EvtObjectConnected(*\*\*kwargs*)**

Triggered when the engine reports that an object is connected (i.e. exists).

This will usually occur at the start of the program in response to the SDK sending RequestConnectedObjects to the engine.

**connected*= 'True if the object connected, False if it disconnected'***

**obj*= 'The object that is connected'***

***class*cozmo.objects.EvtObjectDisappeared(*\*\*kwargs*)**

Triggered whenever an object that was previously being observed is no longer visible.

**obj*= 'The object that is no longer being observed'***

***class*cozmo.objects.EvtObjectLocated(*\*\*kwargs*)**

Triggered when the engine reports that an object is located (i.e. pose is known).

This will usually occur at the start of the program in response to the SDK sending RequestLocatedObjectStates to the engine.

**obj*= 'The object that is located'***

**pose*= 'The cozmo.util.Pose defining the position and rotation of the object'***

**updated*= 'A set of field names that have changed'***

***class*cozmo.objects.EvtObjectMoving(*\*\*kwargs*)**

Triggered when an active object is currently moving.

**acceleration*= 'The currently measured acceleration'***

**move\_duration*= 'The current duration of time (in seconds) that the object has spent moving'***

**obj*= 'The object that is currently moving'***

***class*cozmo.objects.EvtObjectMovingStarted(*\*\*kwargs*)**

Triggered when an active object starts moving.

**acceleration*= 'The currently measured acceleration'***

The currently measured acceleration

| **Type:** | [**Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3) |
| --- | --- |

**obj*= 'The object that started moving'***

***class*cozmo.objects.EvtObjectMovingStopped(*\*\*kwargs*)**

Triggered when an active object stops moving.

**move\_duration*= 'The duration of time (in seconds) that the object spent moving'***

**obj*= 'The object that stopped moving'***

***class*cozmo.objects.EvtObjectObserved(*\*\*kwargs*)**

Triggered whenever an object is visually identified by the robot.

A stream of these events are produced while an object is visible to the robot. Each event has an updated image\_box field.

See EvtObjectAppeared if you only want to know when an object first becomes visible.

**image\_box*= "A comzo.util.ImageBox defining where the object is within Cozmo's camera view"***

**obj*= 'The object that was observed'***

**pose*= 'The cozmo.util.Pose defining the position and rotation of the object'***

**updated*= 'A set of field names that have changed'***

***class*cozmo.objects.EvtObjectTapped(*\*\*kwargs*)**

Triggered when an active object is tapped.

**obj*= 'The object that was tapped'***

**tap\_count*= 'Number of taps detected'***

**tap\_duration*= 'The duration of the tap in ms'***

**tap\_intensity*= 'The intensity of the tap'***

***class*cozmo.objects.ObservableElement(*conn*, *world*, *robot*, *\*\*kw*)**

The base type for anything Cozmo can see.

**is\_visible**

True if the element has been observed recently.

“recently” is defined as [**visibility\_timeout**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableElement.visibility_timeout) seconds.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**last\_event\_time*= None***

The time the last event was received. **None** if no events have yet been received.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**last\_observed\_image\_box*= None***

The ImageBox defining where the object was last visible within Cozmo’s camera view. **None** if the element has not yet been observed.

| **Type:** | [**ImageBox**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.ImageBox) |
| --- | --- |

**last\_observed\_robot\_timestamp*= None***

The robot’s timestamp of the last observed event. **None** if the element has not yet been observed. In milliseconds relative to robot epoch.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**last\_observed\_time*= None***

The time the element was last observed by the robot. **None** if the element has not yet been observed.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**pose**

The pose of the element in the world.

Is **None** for elements that don’t have pose information.

| **Type:** | [**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose) |
| --- | --- |

**time\_since\_last\_seen**

time since this element was last seen (math.inf if never)

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**visibility\_timeout*= 0.4***

Length of time in seconds to go without receiving an observed event before assuming that Cozmo can no longer see an element. Can be overridden in sub classes.

**world*= None***

The robot’s world in which this element is located.

| **Type:** | [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) |
| --- | --- |

***class*cozmo.objects.ObservableObject(*conn*, *world*, *object\_id=None*, *\*\*kw*)**

The base type for objects in Cozmo’s world.

See parent class [**ObservableElement**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableElement) for additional properties and methods.

**descriptive\_name**

A descriptive name for this ObservableObject instance.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**object\_id**

The internal ID assigned to the object.

This value can only be assigned once as it is static in the engine.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**pickupable*= False***

True if this type of object can be physically picked up by Cozmo

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**place\_objects\_on\_this*= False***

True if this type of object can have objects physically placed on it by Cozmo

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

***class*cozmo.objects.LightCube(*cube\_id*, *\*a*, *\*\*kw*)**

A light cube object has four LEDs that Cozmo can actively manipulate and communicate with.

See parent class [**ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject) for additional properties and methods.

**EMPTY\_VOLTAGE*= 1.0***

Voltage where a cube’s battery can be considered empty

**FULL\_VOLTAGE*= 1.5***

Voltage where a cube’s battery can be considered full

**battery\_percentage**

Battery level as a percentage.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**battery\_str**

String representation of the battery level.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**battery\_voltage*= None***

Battery voltage. **None** if no voltage reading has been received yet

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**cube\_id**

The Light Cube ID.

This will be one of [**LightCube1Id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube1Id), [**LightCube2Id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube2Id) and [**LightCube3Id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube3Id). Note: the cube\_id is not the same thing as the object\_id.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**descriptive\_name**

A descriptive name for this LightCube instance.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**is\_connected*= None***

True if the cube is currently connected to the robot via radio.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_moving*= None***

True if the cube’s accelerometer indicates that the cube is moving.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**last\_moved\_robot\_timestamp*= None***

The robot’s timestamp of the last move event. **None** if the cube wasn’t moved yet. In milliseconds relative to robot epoch.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**last\_moved\_start\_robot\_timestamp*= None***

The robot’s timestamp of when the object started moving when last moved **None** if the cube wasn’t moved yet. In milliseconds relative to robot epoch.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**last\_moved\_start\_time*= None***

The time the object started moving when last moved

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**last\_moved\_time*= None***

The time the object was last moved **None** if the cube wasn’t moved yet.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**last\_tapped\_robot\_timestamp*= None***

The robot’s timestamp of the last tapped event. **None** if the cube wasn’t tapped yet. In milliseconds relative to robot epoch.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**last\_tapped\_time*= None***

The time the object was last tapped **None** if the cube wasn’t tapped yet.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**set\_light\_corners(*light1*, *light2*, *light3*, *light4*)**

Set the light for each corner

**set\_lights(*light*)**

Set all lights on the cube

| **Parameters:** | **light** ([**cozmo.lights.Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)) – The settings for the lights. |
| --- | --- |

**set\_lights\_off()**

Turn off all the lights on the cube.

**wait\_for\_tap(*timeout=None*)**

Wait for the object to receive a tap event.

| **Parameters:** | **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Maximum time to wait for a tap, in seconds. None for indefinite |
| --- | --- |
| **Returns:** | A [**EvtObjectTapped**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectTapped) object if a tap was received. |

***class*cozmo.objects.Charger(*\*a*, *\*\*kw*)**

Cozmo’s charger object, which the robot can observe and drive toward.

See parent class [**ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject) for additional properties and methods.

***class*cozmo.objects.CustomObject(*conn*, *world*, *object\_type*, *x\_size\_mm*, *y\_size\_mm*, *z\_size\_mm*, *marker\_width\_mm*, *marker\_height\_mm*, *is\_unique*, *\*\*kw*)**

An object defined by the SDK. It is bound to a specific objectType e.g **CustomType00**.

This defined object is given a size in the x,y and z axis. The dimensions of the markers on the object are also defined. We get an [**cozmo.objects.EvtObjectObserved**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectObserved) message when the robot sees these markers.

See parent class [**ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject) for additional properties and methods.

These objects are created automatically by the engine when Cozmo observes an object with custom markers. For Cozmo to see one of these you must first define an object with custom markers, via one of the following methods: [**define\_custom\_box()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_box). [**define\_custom\_cube()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_cube), or [**define\_custom\_wall()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_wall)

**descriptive\_name**

A descriptive name for this CustomObject instance.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**is\_unique**

True if there should only be one of this object type in the world.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**marker\_height\_mm**

Height in millimeters of the marker on this object.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**marker\_width\_mm**

Width in millimeters of the marker on this object.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**x\_size\_mm**

Size of this object in its X axis, in millimeters.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**y\_size\_mm**

Size of this object in its Y axis, in millimeters.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**z\_size\_mm**

Size of this object in its Z axis, in millimeters.

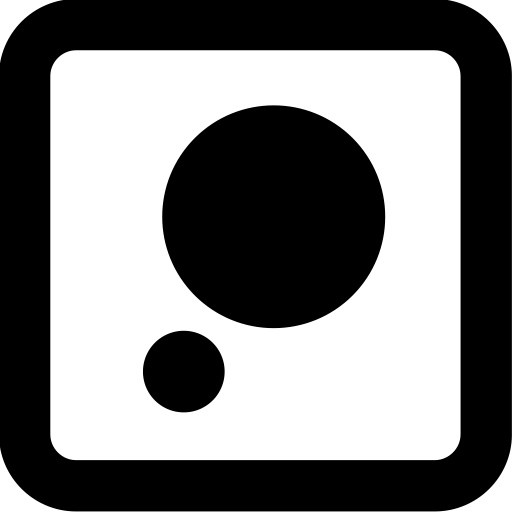
| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.objects.CustomObjectMarkers**

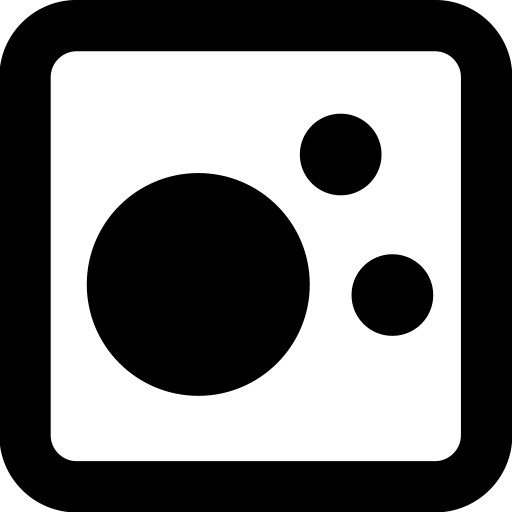
Defines all available custom object markers.

For use with world.define\_custom methods such as [**cozmo.world.World.define\_custom\_box()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_box), [**cozmo.world.World.define\_custom\_cube()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_cube), and [**cozmo.world.World.define\_custom\_wall()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_wall)

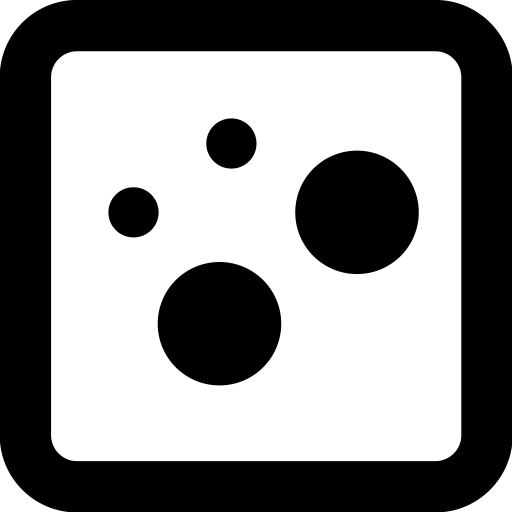
**Circles2*= \_CustomObjectMarker(name='Circles2', id=0)***

******

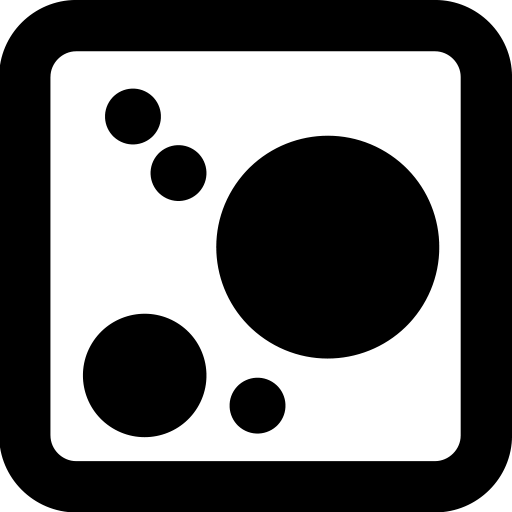
**Circles3*= \_CustomObjectMarker(name='Circles3', id=1)***

******

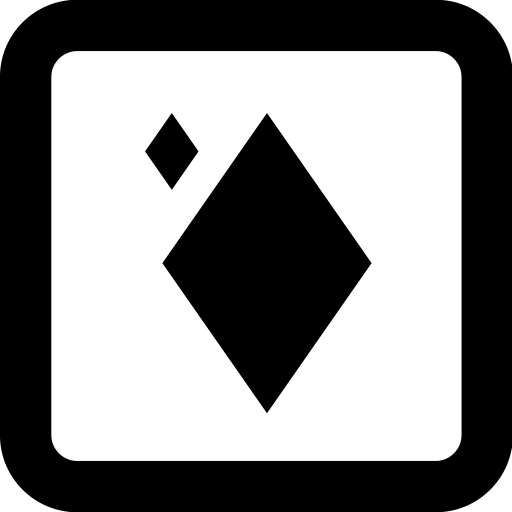
**Circles4*= \_CustomObjectMarker(name='Circles4', id=2)***

******

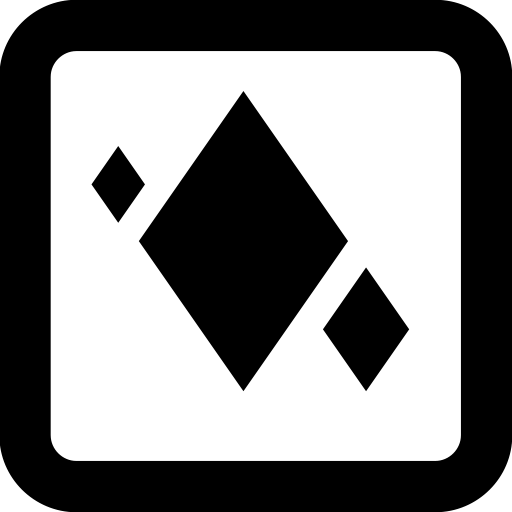
**Circles5*= \_CustomObjectMarker(name='Circles5', id=3)***

******

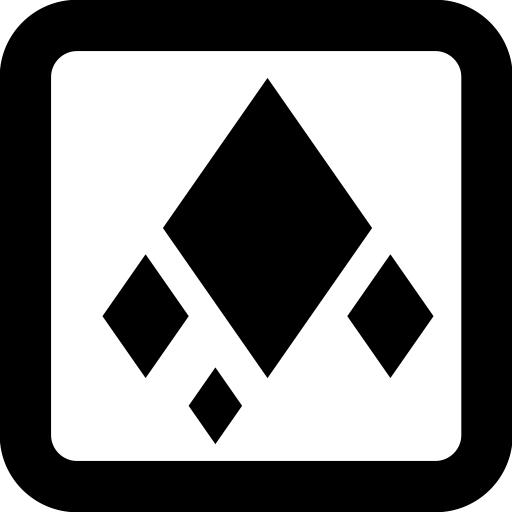
**Diamonds2*= \_CustomObjectMarker(name='Diamonds2', id=4)***

******

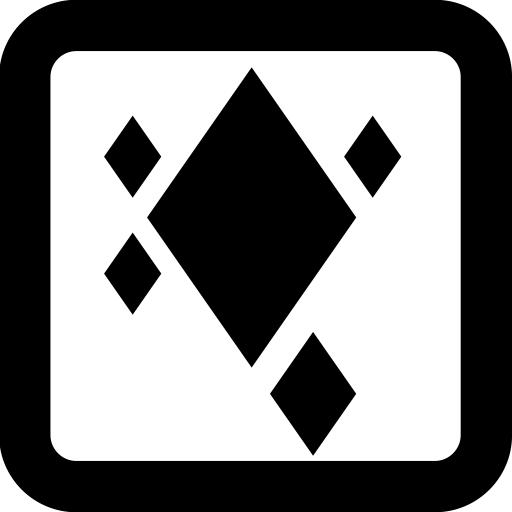
**Diamonds3*= \_CustomObjectMarker(name='Diamonds3', id=5)***

******

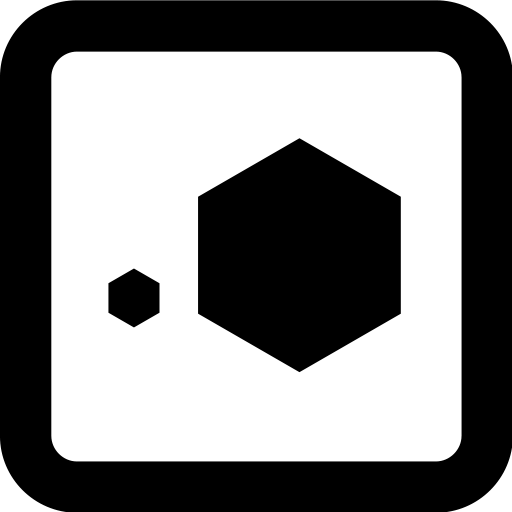
**Diamonds4*= \_CustomObjectMarker(name='Diamonds4', id=6)***

******

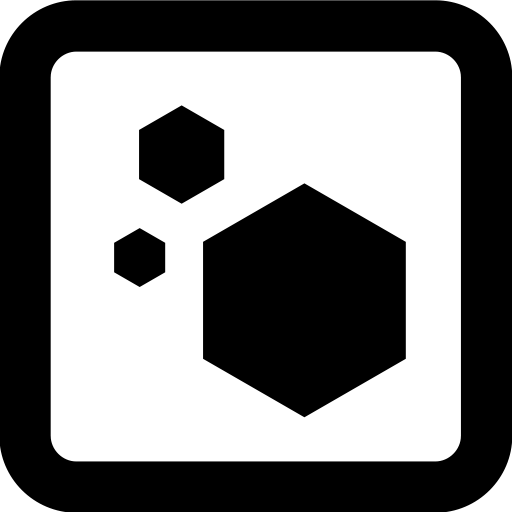
**Diamonds5*= \_CustomObjectMarker(name='Diamonds5', id=7)***

******

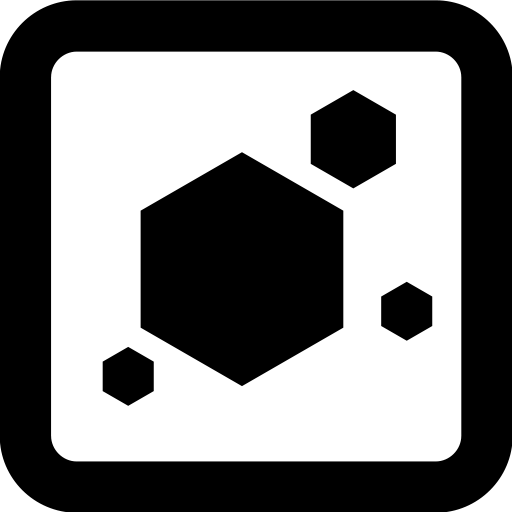
**Hexagons2*= \_CustomObjectMarker(name='Hexagons2', id=8)***

******

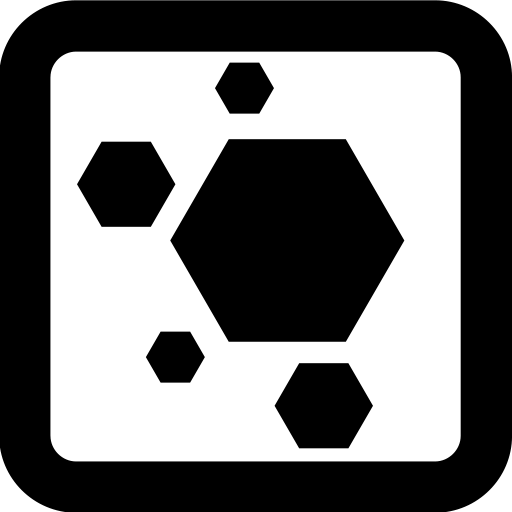
**Hexagons3*= \_CustomObjectMarker(name='Hexagons3', id=9)***

******

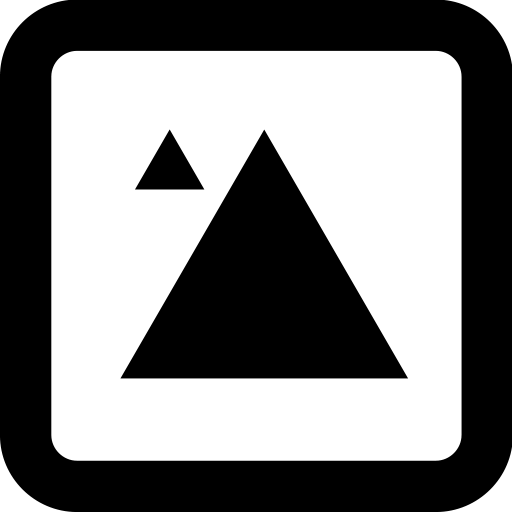
**Hexagons4*= \_CustomObjectMarker(name='Hexagons4', id=10)***

******

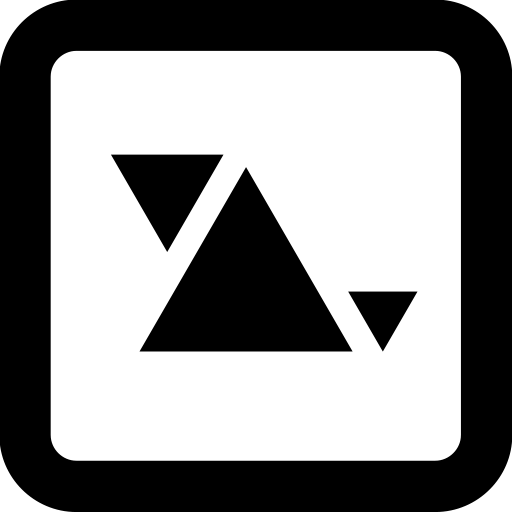
**Hexagons5*= \_CustomObjectMarker(name='Hexagons5', id=11)***

******

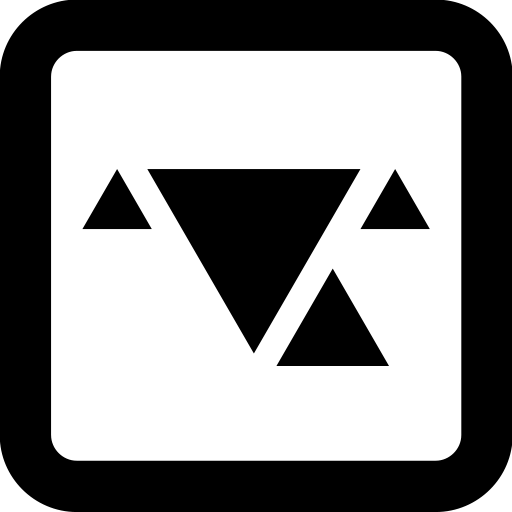
**Triangles2*= \_CustomObjectMarker(name='Triangles2', id=12)***

******

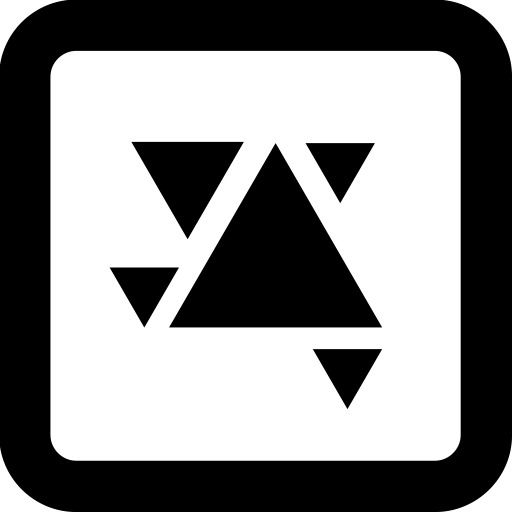
**Triangles3*= \_CustomObjectMarker(name='Triangles3', id=13)***

******

**Triangles4*= \_CustomObjectMarker(name='Triangles4', id=14)***

******

**Triangles5*= \_CustomObjectMarker(name='Triangles5', id=15)***

******

***class*cozmo.objects.CustomObjectTypes**

Defines all available custom object types.

For use with world.define\_custom methods such as [**cozmo.world.World.define\_custom\_box()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_box), [**cozmo.world.World.define\_custom\_cube()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_cube), and [**cozmo.world.World.define\_custom\_wall()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.define_custom_wall)

**CustomType00*= \_CustomObjectType(name='CustomType00', id=17)***

CustomType00 - the first custom object type

**CustomType01*= \_CustomObjectType(name='CustomType01', id=18)***

**CustomType02*= \_CustomObjectType(name='CustomType02', id=19)***

**CustomType03*= \_CustomObjectType(name='CustomType03', id=20)***

**CustomType04*= \_CustomObjectType(name='CustomType04', id=21)***

**CustomType05*= \_CustomObjectType(name='CustomType05', id=22)***

**CustomType06*= \_CustomObjectType(name='CustomType06', id=23)***

**CustomType07*= \_CustomObjectType(name='CustomType07', id=24)***

**CustomType08*= \_CustomObjectType(name='CustomType08', id=25)***

**CustomType09*= \_CustomObjectType(name='CustomType09', id=26)***

**CustomType10*= \_CustomObjectType(name='CustomType10', id=27)***

**CustomType11*= \_CustomObjectType(name='CustomType11', id=28)***

**CustomType12*= \_CustomObjectType(name='CustomType12', id=29)***

**CustomType13*= \_CustomObjectType(name='CustomType13', id=30)***

**CustomType14*= \_CustomObjectType(name='CustomType14', id=31)***

**CustomType15*= \_CustomObjectType(name='CustomType15', id=32)***

**CustomType16*= \_CustomObjectType(name='CustomType16', id=33)***

**CustomType17*= \_CustomObjectType(name='CustomType17', id=34)***

**CustomType18*= \_CustomObjectType(name='CustomType18', id=35)***

**CustomType19*= \_CustomObjectType(name='CustomType19', id=36)***

CustomType19 - the last custom object type

***class*cozmo.objects.FixedCustomObject(*pose*, *x\_size\_mm*, *y\_size\_mm*, *z\_size\_mm*, *object\_id*, *\*a*, *\*\*kw*)**

A fixed object defined by the SDK. It is given a pose and x,y,z sizes.

This object cannot be observed by the robot so its pose never changes. The position is static in Cozmo’s world view; once instantiated, these objects never move. This could be used to make Cozmo aware of objects and know to plot a path around them even when they don’t have any markers.

To create these use [**create\_custom\_fixed\_object()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.create_custom_fixed_object)

**object\_id**

The internal ID assigned to the object.

This value can only be assigned once as it is static in the engine.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**pose**

The pose of the object in the world.

| **Type:** | [**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose) |
| --- | --- |

**x\_size\_mm**

The length of the object in its X axis, in millimeters.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**y\_size\_mm**

The length of the object in its Y axis, in millimeters.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**z\_size\_mm**

The length of the object in its Z axis, in millimeters.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.oled_face.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html)

# **cozmo.oled\_face**

Cozmo’s OLED screen that displays his face - related functions and values.

**Functions**

| [**convert\_image\_to\_screen\_data**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.oled_face.html#cozmo.oled_face.convert_image_to_screen_data)(image[, …]) | Convert an image into the correct format to display on Cozmo’s face. |
| --- | --- |
| [**convert\_pixels\_to\_screen\_data**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.oled_face.html#cozmo.oled_face.convert_pixels_to_screen_data)(pixel\_data, …) | Convert a sequence of pixel data to the correct format to display on Cozmo’s face. |
| [**dimensions**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.oled_face.html#cozmo.oled_face.dimensions)() | Return the dimension (width, height) of the oled screen. |

**cozmo.oled\_face.dimensions()**

Return the dimension (width, height) of the oled screen.

Note: The screen is displayed interlaced, with only every other line displayed This alternates every time the image is changed (no longer than 30 seconds) to prevent screen burn-in. Therefore to ensure the image looks correct on either scan-line offset we use half the vertical resolution

| **Returns:** | A tuple of ints (width, height) |
| --- | --- |

**cozmo.oled\_face.convert\_pixels\_to\_screen\_data(*pixel\_data*, *image\_width*, *image\_height*)**

Convert a sequence of pixel data to the correct format to display on Cozmo’s face.

| **Parameters:** | * **pixel\_data** ([**bytes**](https://docs.python.org/3.5/library/functions.html#bytes)) – sequence of pixel values, should be in binary (1s or 0s) * **image\_width** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – width of the image defined by the pixel\_data * **image\_height** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – height of the image defined by the pixel\_data |
| --- | --- |
| **Returns:** | A [**bytearray**](https://docs.python.org/3.5/library/functions.html#bytearray) object representing all of the pixels (8 pixels packed per byte) |
| **Raises:** | * [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) – Invalid Dimensions * [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) – Bad image\_width * [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) – Bad image\_height |

**cozmo.oled\_face.convert\_image\_to\_screen\_data(*image*, *invert\_image=False*, *pixel\_threshold=127*)**

Convert an image into the correct format to display on Cozmo’s face.

| **Parameters:** | * **image** ([**Image**](https://pillow.readthedocs.io/en/latest/reference/Image.html#PIL.Image.Image)) – The image to display on Cozmo’s face * **invert\_image** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – If true then pixels darker than the threshold are set on * **pixel\_threshold** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The grayscale threshold for what to consider on or off (0..255) |
| --- | --- |
| **Returns:** | A [**bytearray**](https://docs.python.org/3.5/library/functions.html#bytearray) object representing all of the pixels (8 pixels packed per byte) |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html)

# **cozmo.opengl**

This module provides a 3D visualizer for Cozmo’s world state.

It uses PyOpenGL, a Python OpenGL 3D graphics library which is available on most platforms. It also depends on the Pillow library for image processing.

The easiest way to make use of this viewer is to call [**cozmo.run\_program()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.html#cozmo.run_program) with use\_3d\_viewer=True or [**cozmo.run.connect\_with\_3dviewer()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect_with_3dviewer).

**Warning**

This package requires Python to have the PyOpenGL package installed, along with an implementation of GLUT (OpenGL Utility Toolkit).

To install the Python packages do pip3 install --user "cozmo[3dviewer]"

On Windows and Linux you must also install freeglut (macOS / OSX has one preinstalled).

On Linux: sudo apt-get install freeglut3

On Windows: Go to <http://freeglut.sourceforge.net/> to get a freeglut.dll file. It’s included in any of the Windows binaries downloads. Place the DLL next to your Python script, or install it somewhere in your PATH to allow any script to use it.”

**Functions**

| [**LoadMtlFile**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#cozmo.opengl.LoadMtlFile)(filename) | Load a .mtl material file, and return the contents as a dictionary. |
| --- | --- |

**Classes**

| **CubeRenderFrame**(cube) | Minimal copy of a Cube’s state for 1 frame of rendering. |
| --- | --- |
| **CustomObjectRenderFrame**(obj, is\_fixed) | Minimal copy of a CustomObject’s state for 1 frame of rendering. |
| [**DynamicTexture**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#cozmo.opengl.DynamicTexture)() | Wrapper around An OpenGL Texture that can be dynamically updated. |
| **FaceRenderFrame**(face) | Minimal copy of a Face’s state for 1 frame of rendering. |
| [**LoadedObjFile**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#cozmo.opengl.LoadedObjFile)(filename) | The loaded / parsed contents of a 3D Wavefront OBJ file. |
| **ObservableElementRenderFrame**(element) | Minimal copy of a Cube’s state for 1 frame of rendering. |
| [**OpenGLViewer**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#cozmo.opengl.OpenGLViewer)(enable\_camera\_view[, …]) | OpenGL based 3D Viewer. |
| [**OpenGLWindow**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#cozmo.opengl.OpenGLWindow)(x, y, width, height, …) | A Window displaying an OpenGL viewport. |
| [**RenderableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#cozmo.opengl.RenderableObject)(object\_data[, override\_mtl]) | Container for an object that can be rendered via OpenGL. |
| **RobotControlIntents**([left\_wheel\_speed, …]) | Input intents for controlling the robot. |
| **RobotRenderFrame**(robot) | Minimal copy of a Robot’s state for 1 frame of rendering. |
| **WorldRenderFrame**(robot) | Minimal copy of the World’s state for 1 frame of rendering. |

***class*cozmo.opengl.DynamicTexture**

Wrapper around An OpenGL Texture that can be dynamically updated.

**bind()**

Bind the texture for rendering.

**update(*pil\_image: PIL.Image.Image*)**

Update the texture to contain the provided image.

| **Parameters:** | **pil\_image** ([*PIL.Image.Image*](https://pillow.readthedocs.io/en/latest/reference/Image.html#PIL.Image.Image)) – The image to write into the texture. |
| --- | --- |

***class*cozmo.opengl.LoadedObjFile(*filename*)**

The loaded / parsed contents of a 3D Wavefront OBJ file.

This is the intermediary step between the file on the disk, and a renderable 3D object. It supports the subset of the OBJ file that was used in the Cozmo and Cube assets, and does not attempt to exhaustively support every possible setting.

| **Parameters:** | **filename** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – The filename of the OBJ file to load. |
| --- | --- |

***class*cozmo.opengl.OpenGLViewer(*enable\_camera\_view*, *show\_viewer\_controls=True*)**

OpenGL based 3D Viewer.

Handles rendering of both a 3D world view and a 2D camera window.

| **Parameters:** | * **enable\_camera\_view** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to also open a 2nd window to display the live camera view. * **show\_viewer\_controls** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to draw controls on the view. |
| --- | --- |

**disconnect()**

Called from the SDK when the program is complete and it’s time to exit.

***class*cozmo.opengl.OpenGLWindow(*x*, *y*, *width*, *height*, *window\_name*, *is\_3d*)**

A Window displaying an OpenGL viewport.

| **Parameters:** | * **x** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The initial x coordinate of the window in pixels. * **y** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The initial y coordinate of the window in pixels. * **width** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The initial height of the window in pixels. * **height** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The initial height of the window in pixels. * **window\_name** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – The name / title for the window. * **is\_3d** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to create a Window for 3D rendering. |
| --- | --- |

**height*= None***

The height of the window

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**init\_display()**

Initialze the OpenGL display parts of the Window.

**Warning**

Must be called on the same thread as OpenGL (usually the main thread), and after glutInit().

**width*= None***

The width of the window

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

***class*cozmo.opengl.RenderableObject(*object\_data: cozmo.opengl.LoadedObjFile*, *override\_mtl=None*)**

Container for an object that can be rendered via OpenGL.

Can contain multiple meshes, for e.g. articulated objects.

| **Parameters:** | * **object\_data** ([*LoadedObjFile*](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#cozmo.opengl.LoadedObjFile)) – The object data (vertices, faces, etc.) to generate the renderable object from. * **override\_mtl** ([*dict*](https://docs.python.org/3.5/library/stdtypes.html#dict)) – An optional material to use as an override instead of the material specified in the data. This allows one OBJ file to be used to create multiple objects with different materials and textures. Use [**LoadMtlFile()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html#cozmo.opengl.LoadMtlFile) to generate a dict from a MTL file. |
| --- | --- |

**draw\_all()**

Draw all of the meshes.

**meshes*= None***

The individual meshes, indexed by name, for this object.

| **Type:** | [dict](https://docs.python.org/3.5/library/stdtypes.html#dict) |
| --- | --- |

**cozmo.opengl.LoadMtlFile(*filename*)**

Load a .mtl material file, and return the contents as a dictionary.

Supports the subset of MTL required for the Cozmo 3D viewer assets.

| **Parameters:** | **filename** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – The filename of the file to load. |
| --- | --- |
| **Returns:** | A dictionary mapping named MTL attributes to values. |
| **Return type:** | [dict](https://docs.python.org/3.5/library/stdtypes.html#dict) |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.oled_face.html)

# **cozmo.pets**

Pet detection.

Cozmo is capable of detecting pet faces (cats and dogs).

The [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) object keeps track of pets the robot currently knows about, along with those that are currently visible to the camera.

Each pet is assigned a [**Pet**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#cozmo.pets.Pet) object, which generates a number of observable events whenever the pet is observed, etc.

If a pet goes off-screen, it will be assigned a new object\_id (and therefore a new Pet object will be created) when it returns. This is because the system can only tell if something appears to be a cat or a dog; it cannot recognize a specific pet or, for instance, tell the difference between two dogs.

Note that these pet-specific events are also passed up to the [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) object, so events for all pets can be observed by adding handlers there.

**Classes**

| [**EvtPetAppeared**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#cozmo.pets.EvtPetAppeared)(\*\*kwargs) | Triggered whenever a pet is first visually identified by a robot. |
| --- | --- |
| [**EvtPetDisappeared**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#cozmo.pets.EvtPetDisappeared)(\*\*kwargs) | Triggered whenever a pet that was previously being observed is no longer visible. |
| [**EvtPetObserved**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#cozmo.pets.EvtPetObserved)(\*\*kwargs) | Triggered whenever a pet is visually identified by the robot. |
| [**Pet**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#cozmo.pets.Pet)(conn, world, robot[, pet\_id]) | A single pet that Cozmo has detected. |

**cozmo.pets.PET\_VISIBILITY\_TIMEOUT*= 0.4***

Length of time in seconds to go without receiving an observed event before assuming that Cozmo can no longer see a pet.

**cozmo.pets.PET\_TYPE\_CAT*= 'cat'***

Pet Type reported by Cozmo when he thinks it’s a cat

**cozmo.pets.PET\_TYPE\_DOG*= 'dog'***

Pet Type reported by Cozmo when he thinks it’s a dog

**cozmo.pets.PET\_TYPE\_UNKNOWN*= 'unknown'***

Pet Type reported by Cozmo when unsure of type of pet

***class*cozmo.pets.EvtPetAppeared(*\*\*kwargs*)**

Triggered whenever a pet is first visually identified by a robot.

This differs from EvtPetObserved in that it’s only triggered when a pet initially becomes visible. If it disappears for more than PET\_VISIBILITY\_TIMEOUT seconds and then is seen again, a EvtPetDisappeared will be dispatched, followed by another EvtPetAppeared event.

For continuous tracking information about a visible pet, see EvtPetObserved.

**image\_box*= "A comzo.util.ImageBox defining where the pet is within Cozmo's camera view"***

**pet*= 'The Pet instance that was observed'***

**updated*= 'A set of field names that have changed'***

***class*cozmo.pets.EvtPetDisappeared(*\*\*kwargs*)**

Triggered whenever a pet that was previously being observed is no longer visible.

**pet*= 'The Pet instance that is no longer being observed'***

***class*cozmo.pets.EvtPetObserved(*\*\*kwargs*)**

Triggered whenever a pet is visually identified by the robot.

A stream of these events are produced while a pet is visible to the robot. Each event has an updated image\_box field.

See EvtPetAppeared if you only want to know when a pet first becomes visible.

**image\_box*= "A comzo.util.ImageBox defining where the pet is within Cozmo's camera view"***

**pet*= 'The Pet instance that was observed'***

**updated*= 'A set of field names that have changed'***

***class*cozmo.pets.Pet(*conn*, *world*, *robot*, *pet\_id=None*, *\*\*kw*)**

A single pet that Cozmo has detected.

See parent class [**ObservableElement**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableElement) for additional properties and methods.

**pet\_id**

The internal ID assigned to the pet.

This value can only be assigned once as it is static in the engine.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**pet\_type*= None***

The type of Pet (PET\_TYPE\_CAT, PET\_TYPE\_DOG or PET\_TYPE\_UNKNOWN)

**visibility\_timeout*= 0.4***

Length of time in seconds to go without receiving an observed event before assuming that Cozmo can no longer see a pet.

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.opengl.html)

# **cozmo.robot**

Classes and functions relating to an individual Cozmo robot.

The [**cozmo.conn.CozmoConnection.wait\_for\_robot()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection.wait_for_robot) method returns an instance of [**Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) which controls a single Cozmo robot.

The [**Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) class has methods and properties to determine its current state, control its low-level motors, play animations and start behaviors as well as performing high-level actions such as detecting faces and picking up objects.

Each [**Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) has a [**Robot.world**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.world) attribute which represents an instance of a [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World). This tracks the state of the world that Cozmo knows about: The objects and faces it’s currently observing, the camera images it’s receiving, etc. You can monitor the world instance for various events that occur, or monitor individual objects directly: The world instance receives all events that the robot triggers, and nearly all SDK objects inherit from [**cozmo.event.Dispatcher**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher) and therefore inherit methods such as [**wait\_for()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.wait_for) and [**add\_event\_handler()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.add_event_handler).

**Classes**

| [**DisplayOledFaceImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DisplayOledFaceImage)(screen\_data, …) | Represents the “display oled face image” action in progress. |
| --- | --- |
| [**DockWithCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DockWithCube)(obj, approach\_angle, …) | Represents the dock with cube action in progress. |
| [**DriveOffChargerContacts**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DriveOffChargerContacts)(\*\*kw) | Represents the drive off charger contacts action in progress. |
| [**DriveStraight**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DriveStraight)(distance, speed, …) | Represents the “drive straight” action in progress. |
| [**EvtRobotReady**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.EvtRobotReady)(\*\*kwargs) | Generated when the robot has been initialized and is ready for commands. |
| [**EvtRobotStateUpdated**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.EvtRobotStateUpdated)(\*\*kwargs) | Dispatched whenever the robot’s state is updated (multiple times per second). |
| [**EvtUnexpectedMovement**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.EvtUnexpectedMovement)(\*\*kwargs) | Triggered whenever the robot does not move as expected (typically rotation). |
| [**GoToObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.GoToObject)(object\_id, distance\_from\_object, \*\*kw) | Represents the go to object action in progress. |
| [**GoToPose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.GoToPose)(pose, \*\*kw) | Represents the go to pose action in progress. |
| [**LiftPosition**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.LiftPosition)([height, ratio, angle]) | Represents the position of Cozmo’s lift. |
| [**PerformOffChargerContext**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PerformOffChargerContext)(robot, \*\*kw) | A helper class to provide a context manager to do operations while Cozmo is off charger. |
| [**PickupObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PickupObject)(obj[, use\_pre\_dock\_pose]) | Represents the pickup object action in progress. |
| [**PlaceObjectOnGroundHere**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PlaceObjectOnGroundHere)(obj, \*\*kw) | Tracks the state of the “place object on ground here” action. |
| [**PlaceOnObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PlaceOnObject)(obj[, use\_pre\_dock\_pose]) | Tracks the state of the “place on object” action. |
| [**PopAWheelie**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PopAWheelie)(obj, approach\_angle, \*\*kw) | Tracks the progress of a “pop a wheelie” robot action. |
| [**Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot)(conn, robot\_id, is\_primary, \*\*kw) | The interface to a Cozmo robot. |
| [**RollCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.RollCube)(obj, approach\_angle, …) | Represents the roll cube action in progress. |
| [**SayText**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SayText)(text, play\_excited\_animation, …) | Tracks the progress of a say text robot action. |
| [**SetHeadAngle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SetHeadAngle)(angle, max\_speed, accel, …) | Represents the Set Head Angle action in progress. |
| [**SetLiftHeight**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SetLiftHeight)(height, max\_speed, accel, …) | Represents the Set Lift Height action in progress. |
| [**TurnInPlace**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.TurnInPlace)(angle, speed, accel, …) | Tracks the progress of a turn in place robot action. |
| [**TurnTowardsFace**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.TurnTowardsFace)(face, \*\*kw) | Tracks the progress of a turn towards face robot action. |
| [**UnexpectedMovementSide**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.UnexpectedMovementSide) | Defines the side of collision that caused unexpected movement. |
| [**UnexpectedMovementType**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.UnexpectedMovementType) | Defines the type of unexpected movement. |

**cozmo.robot.MIN\_HEAD\_ANGLE*= <Angle -0.44 radians (-25.00 degrees)>***

The minimum angle the robot’s head can be set to

**cozmo.robot.MAX\_HEAD\_ANGLE*= <Angle 0.78 radians (44.50 degrees)>***

The maximum angle the robot’s head can be set to

**cozmo.robot.MIN\_LIFT\_HEIGHT*= <Distance 32.00 mm (1.26 inches)>***

The lowest height-above-ground that lift can be moved to

**cozmo.robot.MAX\_LIFT\_HEIGHT*= <Distance 92.00 mm (3.62 inches)>***

The largest height-above-ground that lift can be moved to

**cozmo.robot.MIN\_LIFT\_ANGLE*= <Angle -0.20 radians (-11.36 degrees)>***

The minimum angle the robot’s lift can be set to

**cozmo.robot.MAX\_LIFT\_ANGLE*= <Angle 0.79 radians (45.41 degrees)>***

The maximum angle the robot’s lift can be set to

***class*cozmo.robot.EvtRobotReady(*\*\*kwargs*)**

Generated when the robot has been initialized and is ready for commands.

**robot*= 'Robot object representing the robot to command'***

***class*cozmo.robot.EvtRobotStateUpdated(*\*\*kwargs*)**

Dispatched whenever the robot’s state is updated (multiple times per second).

**robot*= 'Robot object representing the robot to command'***

***class*cozmo.robot.EvtUnexpectedMovement(*\*\*kwargs*)**

Triggered whenever the robot does not move as expected (typically rotation).

**movement\_side*= 'An UnexpectedMovementSide Object representing the side that is obstructing movement'***

**movement\_type*= 'An UnexpectedMovementType Object representing the type of unexpected movement'***

**robot*= 'Robot object representing the robot to command'***

**timestamp*= 'Robot timestamp for when the unexpected movement occurred'***

***class*cozmo.robot.LiftPosition(*height=None*, *ratio=None*, *angle=None*)**

Represents the position of Cozmo’s lift.

The class allows the position to be referred to as either absolute height above the ground, as a ratio from 0.0 to 1.0, or as the angle of the lift arm relative to the ground.

| **Parameters:** | * **height** ([**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance)) – The height of the lift above the ground. * **ratio** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The ratio from 0.0 to 1.0 that the lift is raised from the ground. * **angle** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – The angle of the lift arm relative to the ground. |
| --- | --- |

**angle**

The angle of the lift arm relative to the ground.

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**height**

The height above the ground.

| **Type:** | [**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance) |
| --- | --- |

**ratio**

The ratio from 0 to 1 that the lift is raised, 0 at the bottom, 1 at the top.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.robot.UnexpectedMovementSide**

Defines the side of collision that caused unexpected movement.

This will always be UNKNOWN while reaction triggers are disabled. Call [**cozmo.robot.Robot.enable\_all\_reaction\_triggers()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.enable_all_reaction_triggers) to enable reaction triggers.

**Back*= \_UnexpectedMovementSide(name='Back', id=2)***

Obstruction detected behind the robot.

**Front*= \_UnexpectedMovementSide(name='Front', id=1)***

Obstruction detected in front of the robot.

**Left*= \_UnexpectedMovementSide(name='Left', id=3)***

Obstruction detected to the left of the robot

**Right*= \_UnexpectedMovementSide(name='Right', id=4)***

Obstruction detected to the right of the robot

**Unknown*= \_UnexpectedMovementSide(name='Unknown', id=0)***

Unable to tell what side obstructed movement. Usually caused by reaction triggers being disabled.

***class*cozmo.robot.UnexpectedMovementType**

Defines the type of unexpected movement.

**TurnedButStopped*= \_UnexpectedMovementType(name='TurnedButStopped', id=0)***

Tried to turn, but couldn’t.

**TurnedInOppositeDirection*= \_UnexpectedMovementType(name='TurnedInOppositeDirection', id=2)***

Expected to turn in one direction, but turned the other way. Also happens when rotation is unexpected.

***class*cozmo.robot.DisplayOledFaceImage(*screen\_data*, *duration\_ms*, *\*\*kw*)**

Represents the “display oled face image” action in progress.

Returned by [**display\_oled\_face\_image()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.display_oled_face_image)

**duration\_ms*= None***

time to keep displaying this image on Cozmo’s face

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**screen\_data*= None***

a sequence of pixels (8 pixels per byte)

| **Type:** | [**bytes**](https://docs.python.org/3.5/library/functions.html#bytes) |
| --- | --- |

***class*cozmo.robot.DockWithCube(*obj*, *approach\_angle*, *alignment\_type*, *distance\_from\_marker*, *\*\*kw*)**

Represents the dock with cube action in progress.

Returned by [**dock\_with\_cube()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.dock_with_cube)

**obj*= None***

The object (e.g. an instance of [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) that is being put down

***class*cozmo.robot.DriveOffChargerContacts(*\*\*kw*)**

Represents the drive off charger contacts action in progress.

Returned by [**drive\_off\_charger\_contacts()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.drive_off_charger_contacts)

***class*cozmo.robot.DriveStraight(*distance*, *speed*, *should\_play\_anim*, *\*\*kw*)**

Represents the “drive straight” action in progress.

Returned by [**drive\_straight()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.drive_straight)

**distance*= None***

The distance to drive

| **Type:** | [**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance) |
| --- | --- |

**should\_play\_anim*= None***

Whether to play an animation whilst driving

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**speed*= None***

The speed to drive at

| **Type:** | [**cozmo.util.Speed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Speed) |
| --- | --- |

***class*cozmo.robot.GoToObject(*object\_id*, *distance\_from\_object*, *\*\*kw*)**

Represents the go to object action in progress.

Returned by [**go\_to\_object()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.go_to_object)

***class*cozmo.robot.GoToPose(*pose*, *\*\*kw*)**

Represents the go to pose action in progress.

Returned by [**go\_to\_pose()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.go_to_pose)

***class*cozmo.robot.PerformOffChargerContext(*robot*, *\*\*kw*)**

A helper class to provide a context manager to do operations while Cozmo is off charger.

***class*cozmo.robot.PickupObject(*obj*, *use\_pre\_dock\_pose=True*, *\*\*kw*)**

Represents the pickup object action in progress.

Returned by [**pickup\_object()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.pickup_object)

**obj*= None***

The object (e.g. an instance of [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) that was picked up

**use\_pre\_dock\_pose*= None***

A bool that is true when Cozmo needs to go to a pose before attempting to navigate to the object

***class*cozmo.robot.PlaceObjectOnGroundHere(*obj*, *\*\*kw*)**

Tracks the state of the “place object on ground here” action.

Returned by [**place\_object\_on\_ground\_here()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.place_object_on_ground_here)

**obj*= None***

The object (e.g. an instance of [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) that is being put down

***class*cozmo.robot.PlaceOnObject(*obj*, *use\_pre\_dock\_pose=True*, *\*\*kw*)**

Tracks the state of the “place on object” action.

return by [**place\_on\_object()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.place_on_object)

**obj*= None***

The object (e.g. an instance of [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) that the held object will be placed on

**use\_pre\_dock\_pose*= None***

A bool that is true when Cozmo needs to go to a pose before attempting to navigate to the object

***class*cozmo.robot.PopAWheelie(*obj*, *approach\_angle*, *\*\*kw*)**

Tracks the progress of a “pop a wheelie” robot action.

Returned by [**pop\_a\_wheelie()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.pop_a_wheelie)

**obj*= None***

An object (e.g. an instance of [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) being used as leverage to push cozmo on his back

***class*cozmo.robot.RollCube(*obj*, *approach\_angle*, *check\_for\_object\_on\_top*, *\*\*kw*)**

Represents the roll cube action in progress.

Returned by [**roll\_cube()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.roll_cube)

**check\_for\_object\_on\_top*= None***

whether to check if there is an object on top

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**obj*= None***

The object (e.g. an instance of [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) that is being put down

***class*cozmo.robot.SayText(*text*, *play\_excited\_animation*, *use\_cozmo\_voice*, *duration\_scalar*, *voice\_pitch*, *\*\*kw*)**

Tracks the progress of a say text robot action.

Returned by [**say\_text()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.say_text)

***class*cozmo.robot.SetHeadAngle(*angle*, *max\_speed*, *accel*, *duration*, *warn\_on\_clamp*, *\*\*kw*)**

Represents the Set Head Angle action in progress. Returned by [**set\_head\_angle()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.set_head_angle)

**accel*= None***

Acceleration of Cozmo’s head in radians per second squared

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**duration*= None***

Time for Cozmo’s head to turn in seconds

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**max\_speed*= None***

Maximum speed of Cozmo’s head in radians per second

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.robot.SetLiftHeight(*height*, *max\_speed*, *accel*, *duration*, *\*\*kw*)**

Represents the Set Lift Height action in progress. Returned by [**set\_lift\_height()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.set_lift_height)

**accel*= None***

Acceleration of Cozmo’s lift in radians per second squared

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**duration*= None***

Time for Cozmo’s lift to turn in seconds

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**max\_speed*= None***

Maximum speed of Cozmo’s lift in radians per second

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.robot.TurnInPlace(*angle*, *speed*, *accel*, *angle\_tolerance*, *is\_absolute*, *\*\*kw*)**

Tracks the progress of a turn in place robot action.

Returned by [**turn\_in\_place()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.turn_in_place)

**accel*= None***

Acceleration of angular turn (per second squared).

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**angle*= None***

The angle to turn

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**angle\_tolerance*= None***

The minimum angular tolerance to consider the action complete (this is clamped to a minimum of 2 degrees internally).

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**is\_absolute*= None***

True to turn to a specific angle, False to turn relative to the current pose.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**speed*= None***

Angular turn speed (per second).

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

***class*cozmo.robot.TurnTowardsFace(*face*, *\*\*kw*)**

Tracks the progress of a turn towards face robot action.

Returned by [**turn\_towards\_face()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.turn_towards_face)

**face*= None***

The face to turn towards

| **Type:** | [**Face**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face) |
| --- | --- |

***class*cozmo.robot.Robot(*conn*, *robot\_id: int*, *is\_primary: bool*, *\*\*kw*)**

The interface to a Cozmo robot.

A robot has access to:

* **A** [**World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) **object (**[**cozmo.robot.Robot.world**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.world)**),**
* which tracks the state of the world the robot knows about
* **A** [**Camera**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera) **object (**[**cozmo.robot.Robot.camera**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.camera)**),**
* which provides access to Cozmo’s camera
* An Animations object, controlling the playing of animations on the robot
* A Behaviors object, starting and ending robot behaviors such as looking around

Robots are instantiated by the [**CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) object and emit a [**EvtRobotReady**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.EvtRobotReady) when it has been configured and is ready to be commanded.

**abort\_all\_actions(*log\_abort\_messages=False*)**

Abort all actions on this robot

| **Parameters:** | **log\_abort\_messages** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to log info on every action that is aborted. |
| --- | --- |

Abort / Cancel any action that is currently either running or queued within the engine

**accelerometer*= None***

The current accelerometer reading (x,y,z) In mm/s^2, measured in Cozmo’s head (e.g. x=0 when Cozmo’s head is level but x = z = ~7000 mm/s^2 when Cozmo’s head is angled 45 degrees up)

| **Type:** | [**cozmo.util.Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3) |
| --- | --- |

**anim\_names**

Set of all the available animation names

An alias of **cozmo.conn.anim\_names**.

Generally animation triggers are preferred over explict animation names: See [**cozmo.anim.Triggers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Triggers) for available animation triggers.

| **Type:** | set of string |
| --- | --- |

**anim\_triggers**

list of [**cozmo.anim.Triggers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Triggers), specifying available animation triggers

These can be sent to the play\_anim\_trigger to make the robot perform animations.

An alias of **cozmo.anim.Triggers.trigger\_list**.

**animation\_factory*= functools.partial(<class 'cozmo.anim.Animation'>, loop=None)***

The factory function that returns a [**cozmo.anim.Animation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Animation) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**animation\_trigger\_factory*= functools.partial(<class 'cozmo.anim.AnimationTrigger'>, loop=None)***

The factory function that returns a [**cozmo.anim.AnimationTrigger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.AnimationTrigger) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**are\_wheels\_moving**

True if Cozmo’s wheels/treads are currently moving.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**backup\_onto\_charger(*max\_drive\_time=3*)**

Attempts to reverse robot onto its charger.

This method assumes the charger is directly behind the robot and will keep driving straight back until charger is in contact, or until a timeout is reached.

| **Parameters:** | **max\_drive\_time** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The maximum amount of time in seconds to reverse the robot without detecting the charger. |
| --- | --- |

**battery\_voltage*= None***

The current battery voltage (not linear, but < 3.5 is low)

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**behavior\_factory*= functools.partial(<class 'cozmo.behavior.Behavior'>, loop=None)***

The factory function that returns a [**cozmo.behavior.Behavior**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.Behavior) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**camera*= None***

Provides access to the robot’s camera

| **Type:** | [**cozmo.camera.Camera**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera) |
| --- | --- |

**camera\_config**

The read-only config/calibration for this robot’s camera

*Deprecated since version 0.12.0:* Use: [**cozmo.camera.Camera.config()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera.config) instead.

| **Type:** | **cozmo.robot.CameraConfig** |
| --- | --- |

**camera\_factory*= functools.partial(<class 'cozmo.camera.Camera'>, loop=None)***

The factory function that returns a [**cozmo.camera.Camera**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**carrying\_object\_id*= None***

The ID of the object currently being carried (-1 if none)

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**carrying\_object\_on\_top\_id*= None***

The ID of the object on top of the object currently being carried (-1 if none)

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**clear\_idle\_animation()**

Clears any Idle Animation currently playing on Cozmo

**conn*= None***

The active connection to the engine.

| **Type:** | [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) |
| --- | --- |

**current\_behavior**

Cozmo’s currently active behavior.

| **Type:** | [**cozmo.behavior.Behavior**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.Behavior) |
| --- | --- |

**device\_accel\_raw*= None***

The current accelerometer reading for the connected mobile device. Requires that you have first called [**enable\_device\_imu()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.enable_device_imu) with enable\_raw = True. See [**device\_accel\_user**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.device_accel_user) for a user-filtered equivalent.

| **Type:** | [**cozmo.util.Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3) |
| --- | --- |

**device\_accel\_user*= None***

The current user-filtered accelerometer reading for the connected mobile device. Requires that you have first called [**enable\_device\_imu()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.enable_device_imu) with enable\_user = True. This filtered version removes the constant acceleration from Gravity. See [**device\_accel\_raw**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.device_accel_raw) for a raw version.

| **Type:** | [**cozmo.util.Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3) |
| --- | --- |

**device\_gyro*= None***

The current gyro reading for the connected mobile device. Requires that you have first called [**enable\_device\_imu()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.enable_device_imu) with enable\_gyro = True

| **Type:** | [**cozmo.util.Quaternion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Quaternion) |
| --- | --- |

**display\_oled\_face\_image(*screen\_data*, *duration\_ms*, *in\_parallel=True*)**

Display a bitmap image on Cozmo’s OLED face screen.

| **Parameters:** | * **screen\_data** ([**bytes**](https://docs.python.org/3.5/library/functions.html#bytes)) – a sequence of pixels (8 pixels per byte) (from e.g. [**cozmo.oled\_face.convert\_pixels\_to\_screen\_data()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.oled_face.html#cozmo.oled_face.convert_pixels_to_screen_data)). * **duration\_ms** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – time to keep displaying this image on Cozmo’s face (clamped to 30 seconds in engine). * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.DisplayOledFaceImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DisplayOledFaceImage) **action object which**  can be queried to see when it is complete. |
| **Raises:** | *:class:`cozmo.exceptions.RobotBusy` if another action is already* – running and in\_parallel==False |

**display\_oled\_face\_image\_factory*= functools.partial(<class 'cozmo.robot.DisplayOledFaceImage'>, loop=None)***

The factory function that returns a [**DisplayOledFaceImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DisplayOledFaceImage) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**dock\_with\_cube(*target\_object*, *approach\_angle=None*, *alignment\_type=\_RobotAlignmentType(name='LiftPlate'*, *id=1)*, *distance\_from\_marker=None*, *in\_parallel=False*, *num\_retries=0*)**

Tells Cozmo to dock with a specified cube object.

| **Parameters:** | * **target\_object** ([**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) – The cube to dock with. * **approach\_angle** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – The angle to approach the cube from. For example, 180 degrees will cause cozmo to drive past the cube and approach it from behind. * **alignment\_type** ([**cozmo.robot\_alignment.RobotAlignmentTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot_alignment.html#cozmo.robot_alignment.RobotAlignmentTypes)) – which part of the robot to line up with the front of the object. * **distance\_from\_marker** ([**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance)) – distance from the cube marker to stop when using Custom alignment * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.DockWithCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DockWithCube) **action object which can be queried**  to see when it is complete. |

**dock\_with\_cube\_factory*= functools.partial(<class 'cozmo.robot.DockWithCube'>, loop=None)***

The factory function that returns a [**DockWithCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DockWithCube) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**drive\_off\_charger\_contacts(*in\_parallel=False*, *num\_retries=0*)**

Tells Cozmo to drive forward slightly to get off the charger contacts.

All motor movement is disabled while Cozmo is on the charger to prevent hardware damage. This command is the one exception and provides a way to drive forward a little to disconnect from the charger contacts and thereby re-enable all other commands.

| **Parameters:** | * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.DriveOffChargerContacts**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DriveOffChargerContacts) **action object which**  can be queried to see when it is complete. |

**drive\_off\_charger\_contacts\_factory*= functools.partial(<class 'cozmo.robot.DriveOffChargerContacts'>, loop=None)***

The factory function that returns a [**DriveOffChargerContacts**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DriveOffChargerContacts) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**drive\_off\_charger\_on\_connect*= True***

Set to True if the robot should drive off the charger as soon as the SDK connects to the engine. Defaults to True.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**drive\_straight(*distance*, *speed*, *should\_play\_anim=True*, *in\_parallel=False*, *num\_retries=0*)**

Tells Cozmo to drive in a straight line

Cozmo will drive for the specified distance (forwards or backwards)

| **Parameters:** | * **distance** ([**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance)) – The distance to drive (>0 for forwards, <0 for backwards) * **speed** ([**cozmo.util.Speed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Speed)) – The speed to drive at (should always be >0, the abs(speed) is used internally) * **should\_play\_anim** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Whether to play idle animations whilst driving (tilt head, hum, animated eyes, etc.) * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.DriveStraight**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DriveStraight) **action object which**  can be queried to see when it is complete. |

**drive\_straight\_factory*= functools.partial(<class 'cozmo.robot.DriveStraight'>, loop=None)***

The factory function that returns a [**DriveStraight**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.DriveStraight) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**drive\_wheel\_motors(*l\_wheel\_speed*, *r\_wheel\_speed*, *l\_wheel\_acc=None*, *r\_wheel\_acc=None*)**

Tell Cozmo to move his wheels / treads at a given speed.

The wheels will continue to move at that speed until commanded to drive at a new speed, or if [**stop\_all\_motors()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.stop_all_motors) is called.

| **Parameters:** | * **l\_wheel\_speed** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Speed of the left tread (in millimeters per second) * **r\_wheel\_speed** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Speed of the right tread (in millimeters per second) * **l\_wheel\_acc** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Acceleration of left tread (in millimeters per second squared) **None** value defaults this to the same as l\_wheel\_speed. * **r\_wheel\_acc** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Acceleration of right tread (in millimeters per second squared) **None** value defaults this to the same as r\_wheel\_speed. |
| --- | --- |

**drive\_wheels(*l\_wheel\_speed*, *r\_wheel\_speed*, *l\_wheel\_acc=None*, *r\_wheel\_acc=None*, *duration=None*)**

Tell Cozmo to move his wheels / treads at a given speed, and optionally stop them after a given duration.

If duration is **None** then this is equivalent to the non-async [**drive\_wheel\_motors()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.drive_wheel_motors) method.

| **Parameters:** | * **l\_wheel\_speed** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Speed of the left tread (in millimeters per second). * **r\_wheel\_speed** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Speed of the right tread (in millimeters per second). * **l\_wheel\_acc** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Acceleration of left tread (in millimeters per second squared). **None** value defaults this to the same as l\_wheel\_speed. * **r\_wheel\_acc** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Acceleration of right tread (in millimeters per second squared). **None** value defaults this to the same as r\_wheel\_speed. * **duration** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Time for the robot to drive. Will call [**stop\_all\_motors()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.stop_all_motors) after this duration has passed. |
| --- | --- |

**enable\_all\_reaction\_triggers(*should\_enable*)**

Enable or disable Cozmo’s responses to being handled or observing the world.

| **Parameters:** | **should\_enable** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True if the robot should react to its environment. |
| --- | --- |

**enable\_auto\_exposure()**

*Deprecated since version 0.12.0:* Use: [**cozmo.camera.Camera.enable\_auto\_exposure()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera.enable_auto_exposure) instead.

**enable\_device\_imu(*enable\_raw=False*, *enable\_user=False*, *enable\_gyro=False*)**

Enable streaming of the connected Mobile devices’ IMU data.

The accelerometer and gyro data for the connected phone or tablet can be streamed from the app to the SDK. You can request any combination of the 3 data types.

| **Parameters:** | * **enable\_raw** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to enable streaming of the raw accelerometer data, which can be accessed via [**device\_accel\_raw**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.device_accel_raw) * **enable\_user** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to enable streaming of the user-filtered accelerometer data, which can be accessed via [**device\_accel\_user**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.device_accel_user) * **enable\_gyro** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to enable streaming of the gyro data, which can be accessed via [**device\_gyro**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.device_gyro) |
| --- | --- |

**enable\_facial\_expression\_estimation(*enable=True*)**

Enable or Disable facial expression estimation

Cozmo can optionally estimate the facial expression for human faces to see if he thinks they’re happy, sad, etc.

| **Parameters:** | **enable** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to enable facial expression estimation, False to disable it. By default Cozmo starts with it disabled to save on processing time. |
| --- | --- |

**enable\_freeplay\_cube\_lights(*enable=True*)**

Enable, or disable, the automatic cube light mode used in freeplay.

Enabling the freeplay cube light mode causes the cubes to automatically pulse blue when Cozmo can see them - as seen in the Cozmo app during freeplay mode. This is disabled by default in SDK mode because it overrides any other calls to set the cube light colors.

| **Parameters:** | **enable** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to enable the freeplay cube light mode, False to disable it. |
| --- | --- |

**enable\_stop\_on\_cliff(*enable*)**

Enable or disable Cozmo’s ability to drive off a cliff.

| **Parameters:** | **enable** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True if the robot should stop moving when a cliff is encountered. |
| --- | --- |

**go\_to\_object(*target\_object*, *distance\_from\_object*, *in\_parallel=False*, *num\_retries=0*)**

Tells Cozmo to drive to the specified object.

| **Parameters:** | * **target\_object** ([**cozmo.objects.ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject)) – The destination object. CustomObject instances are not supported. * **distance\_from\_object** ([**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance)) – The distance from the object to stop. This is the distance between the origins. For instance, the distance from the robot’s origin (between Cozmo’s two front wheels) to the cube’s origin (at the center of the cube) is ~40mm. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.GoToObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.GoToObject) **action object which can be queried**  to see when it is complete. |

**go\_to\_object\_factory*= functools.partial(<class 'cozmo.robot.GoToObject'>, loop=None)***

The factory function that returns a [**GoToObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.GoToObject) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**go\_to\_pose(*pose*, *relative\_to\_robot=False*, *in\_parallel=False*, *num\_retries=0*)**

Tells Cozmo to drive to the specified pose and orientation.

If relative\_to\_robot is set to True, the given pose will assume the robot’s pose as its origin.

Since the robot understands position by monitoring its tread movement, it does not understand movement in the z axis. This means that the only applicable elements of pose in this situation are position.x position.y and rotation.angle\_z.

| **Parameters:** | * **pose** – ([**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose)): The destination pose. * **relative\_to\_robot** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Whether the given pose is relative to the robot’s pose. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.GoToPose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.GoToPose) **action object which can be queried**  to see when it is complete. |

**go\_to\_pose\_factory*= functools.partial(<class 'cozmo.robot.GoToPose'>, loop=None)***

The factory function that returns a [**GoToPose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.GoToPose) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**gyro*= None***

The current gyro reading (x,y,z) In radians/s, measured in Cozmo’s head. Therefore a large value in a given component would indicate Cozmo is being rotated around that axis (where x=forward, y=left, z=up), e.g. y = -5 would indicate that Cozmo is being rolled onto his back

| **Type:** | [**cozmo.util.Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3) |
| --- | --- |

**has\_in\_progress\_actions**

True if Cozmo has any SDK-triggered actions still in progress.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**head\_angle**

Cozmo’s head angle (up/down).

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**head\_tracking\_object\_id*= None***

The ID of the object the head is tracking to (-1 if none)

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**is\_anim\_buffer\_full**

True if Cozmo’s animation buffer is full (on robot).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_animating**

True if Cozmo is currently playing an animation.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_animating\_idle**

True if Cozmo is currently playing an idle animation.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_behavior\_running**

True if Cozmo is currently running a behavior.

When Cozmo is running a behavior he will behave fairly autonomously (playing animations and other actions as desired). Attempting to drive Cozmo whilst in this mode will likely have unexpected behavior on the robot and confuse Cozmo.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_carrying\_block**

True if Cozmo is currently carrying a block.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_charging**

True if Cozmo is currently charging.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_cliff\_detected**

True if Cozmo detected a cliff (in front of the robot).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_device\_accelerometer\_supported**

True if the attached mobile device supports accelerometer data.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_device\_gyro\_supported**

True if the attached mobile device supports gyro data.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_falling**

True if Cozmo is currently falling.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_freeplay\_mode\_active**

True if Cozmo is in freeplay mode.

When Cozmo is in freeplay mode he will behave autonomously (playing behaviors, animations and other actions as desired). Attempting to drive Cozmo whilst in this mode will likely have unexpected behavior on the robot and confuse Cozmo.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_head\_in\_pos**

True if Cozmo’s head is in the desired position (False if still trying to move there).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_lift\_in\_pos**

True if Cozmo’s lift is in the desired position (False if still trying to move there).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_localized**

True if Cozmo is localized (i.e. knows where he is with respect to a cube, and has both treads on the ground).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_moving**

True if Cozmo is currently moving anything (head, lift or wheels/treads).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_on\_charger**

True if Cozmo is currently on the charger.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_pathing**

True if Cozmo is currently traversing a path.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_picked\_up**

True if Cozmo is currently picked up (in the air).

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_picking\_or\_placing**

True if Cozmo is picking or placing something.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_primary*= None***

Specifies that this is the primary robot (always True currently)

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_ready**

True if the robot has been initialized and is ready to accept commands.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**last\_image\_robot\_timestamp*= None***

The robot’s timestamp for the last image seen. **None** if no image was received yet. In milliseconds relative to robot epoch.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**left\_wheel\_speed*= None***

Speed of the left wheel

| **Type:** | [**cozmo.util.Speed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Speed) |
| --- | --- |

**lift\_angle**

Angle of Cozmo’s lift relative to the ground.

In [**MIN\_LIFT\_ANGLE**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.MIN_LIFT_ANGLE) to [**MAX\_LIFT\_ANGLE**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.MAX_LIFT_ANGLE) range.

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**lift\_height**

Height of Cozmo’s lift from the ground.

In [**MIN\_LIFT\_HEIGHT**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.MIN_LIFT_HEIGHT) to [**MAX\_LIFT\_HEIGHT**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.MAX_LIFT_HEIGHT) range.

| **Type:** | [**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance) |
| --- | --- |

**lift\_position**

The position of Cozmo’s lift.

| **Type:** | [**LiftPosition**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.LiftPosition) |
| --- | --- |

**lift\_ratio**

Ratio from 0 to 1 of how high Cozmo’s lift is.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**localized\_to\_object\_id*= None***

The ID of the object that the robot is localized to (-1 if none)

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**move\_head(*speed*)**

Tell Cozmo’s head motor to move with a certain speed.

Positive speed for up, negative speed for down. Measured in radians per second.

| **Parameters:** | **speed** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Motor speed for Cozmo’s head, measured in radians per second. |
| --- | --- |

**move\_lift(*speed*)**

Tell Cozmo’s lift motor to move with a certain speed.

Positive speed for up, negative speed for down. Measured in radians per second.

| **Parameters:** | **speed** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Motor speed for Cozmo’s lift, measured in radians per second. |
| --- | --- |

**perform\_off\_charger()**

Returns a context manager to move the robot off of and back onto the charger.

If the robot is on the charger, it will move a short distance off the contacts, perform the code wrapped by the context and then move the robot back onto the charger after the wrapped code completes.

Synchronous example:

**with** robot.perform\_off\_charger():

action = robot.say\_text("Hello")

action.wait\_for\_completed()

Asynchronous example:

**async** **with** robot.perform\_off\_charger():

action = robot.say\_text("Hello")

**await** action.wait\_for\_completed()

**perform\_off\_charger\_factory*= functools.partial(<class 'cozmo.robot.PerformOffChargerContext'>, loop=None)***

The factory function that returns a [**cozmo.robot.PerformOffChargerContext**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PerformOffChargerContext) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**pickup\_object(*obj*, *use\_pre\_dock\_pose=True*, *in\_parallel=False*, *num\_retries=0*)**

Instruct the robot to pick up the supplied object.

| **Parameters:** | * **obj** ([**cozmo.objects.ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject)) – The target object to pick up where **obj.pickupable** is True. * **use\_pre\_dock\_pose** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – whether or not to try to immediately pick up an object or first position the robot next to the object. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.PickupObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PickupObject) **action object which can be**  queried to see when it is complete. |
| **Raises:** | * *:class:`cozmo.exceptions.RobotBusy` if another action is already* – running and in\_parallel==False * [**cozmo.exceptions.NotPickupable**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.NotPickupable) if object type can’t be picked up. |

**pickup\_object\_factory*= functools.partial(<class 'cozmo.robot.PickupObject'>, loop=None)***

The factory function that returns a [**PickupObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PickupObject) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**place\_object\_on\_ground\_here(*obj*, *in\_parallel=False*, *num\_retries=0*)**

Ask Cozmo to place the object he is carrying on the ground at the current location.

| **Parameters:** | * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.PlaceObjectOnGroundHere**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PlaceObjectOnGroundHere) **action object which**  can be queried to see when it is complete. |
| **Raises:** | *:class:`cozmo.exceptions.RobotBusy` if another action is already* – running and in\_parallel==False |

**place\_object\_on\_ground\_here\_factory*= functools.partial(<class 'cozmo.robot.PlaceObjectOnGroundHere'>, loop=None)***

The factory function that returns a [**PlaceObjectOnGroundHere**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PlaceObjectOnGroundHere) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**place\_on\_object(*obj*, *use\_pre\_dock\_pose=True*, *in\_parallel=False*, *num\_retries=0*)**

Asks Cozmo to place the currently held object onto a target object.

| **Parameters:** | * **obj** ([**cozmo.objects.ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject)) – The target object to place current held object on, where obj.place\_objects\_on\_this is True. * **use\_pre\_dock\_pose** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Whether or not to try to immediately pick up an object or first position the robot next to the object. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.PlaceOnObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PlaceOnObject) **action object which can be**  queried to see when it is complete. |
| **Raises:** | * *:class:`cozmo.exceptions.RobotBusy` if another action is already* – running and in\_parallel==False * [**cozmo.exceptions.CannotPlaceObjectsOnThis**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.exceptions.html#cozmo.exceptions.CannotPlaceObjectsOnThis) if the object cannot have objects * placed on it. |

**place\_on\_object\_factory*= functools.partial(<class 'cozmo.robot.PlaceOnObject'>, loop=None)***

The factory function that returns a [**PlaceOnObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PlaceOnObject) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**play\_anim(*name*, *loop\_count=1*, *in\_parallel=False*, *num\_retries=0*, *ignore\_body\_track=False*, *ignore\_head\_track=False*, *ignore\_lift\_track=False*)**

Starts an animation playing on a robot.

Returns an Animation object as soon as the request to play the animation has been sent. Call the wait\_for\_completed method on the animation if you wish to wait for completion (or listen for the [**cozmo.anim.EvtAnimationCompleted**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.EvtAnimationCompleted) event).

**Warning: Specific animations may be renamed or removed in future updates of the app.**

If you want your program to work more reliably across all versions we recommend using [**play\_anim\_trigger()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.play_anim_trigger) instead.

| **Parameters:** | * **name** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – The name of the animation to play. * **loop\_count** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to play the animation. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. * **ignore\_body\_track** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to ignore the animation track for Cozmo’s body (i.e. the wheels / treads). * **ignore\_head\_track** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to ignore the animation track for Cozmo’s head. * **ignore\_lift\_track** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to ignore the animation track for Cozmo’s lift. |
| --- | --- |
| **Returns:** | **A** [**cozmo.anim.Animation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Animation) **action object which can be queried**  to see when it is complete. |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) if supplied an invalid animation name. |

**play\_anim\_trigger(*trigger*, *loop\_count=1*, *in\_parallel=False*, *num\_retries=0*, *use\_lift\_safe=False*, *ignore\_body\_track=False*, *ignore\_head\_track=False*, *ignore\_lift\_track=False*)**

Starts an animation trigger playing on a robot.

As noted in the Triggers class, playing a trigger requests that an animation of a certain class starts playing, rather than an exact animation name as influenced by the robot’s mood, and other factors.

| **Parameters:** | * **trigger** ([*object*](https://docs.python.org/3.5/library/functions.html#object)) – An attribute of the [**cozmo.anim.Triggers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Triggers) class * **loop\_count** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to play the animation * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. * **use\_lift\_safe** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to automatically ignore the lift track if Cozmo is currently carrying an object. * **ignore\_body\_track** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to ignore the animation track for Cozmo’s body (i.e. the wheels / treads). * **ignore\_head\_track** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to ignore the animation track for Cozmo’s head. * **ignore\_lift\_track** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to ignore the animation track for Cozmo’s lift. |
| --- | --- |
| **Returns:** | **A** [**cozmo.anim.AnimationTrigger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.AnimationTrigger) **action object which can be**  queried to see when it is complete |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) if supplied an invalid animation trigger. |

**play\_audio(*audio\_event*)**

Sends an audio event to the engine

Most of these come in pairs, with one to start an audio effect, and one to stop if desired.

**Example**

[**cozmo.audio.AudioEvents.SfxSharedSuccess**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.audio.html#cozmo.audio.AudioEvents.SfxSharedSuccess) starts a sound [**cozmo.audio.AudioEvents.SfxSharedSuccessStop**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.audio.html#cozmo.audio.AudioEvents.SfxSharedSuccessStop) interrupts that sound in progress

Some events are part of the TinyOrchestra system which have special behavior. This system can be intitialized and stopped, and various musical instruments can be turned on and off while it is running.

| **Parameters:** | **audio\_event** ([*object*](https://docs.python.org/3.5/library/functions.html#object)) – An attribute of the [**cozmo.audio.AudioEvents**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.audio.html#cozmo.audio.AudioEvents) class |
| --- | --- |

**play\_song(*song\_notes*, *loop\_count=1*, *in\_parallel=False*, *num\_retries=0*)**

Starts playing song on the robot.

Plays a provided array of SongNotes using a custom animation on the robot.

| **Parameters:** | **song\_notes** ([*object*](https://docs.python.org/3.5/library/functions.html#object)*[]*) – An array of [**cozmo.song.SongNote**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html#cozmo.song.SongNote) classes |
| --- | --- |
| **Returns:** | **A** [**cozmo.anim.Animation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Animation) **action object which can be queried**  to see when it is complete. |

**pop\_a\_wheelie(*target\_object*, *approach\_angle=None*, *in\_parallel=False*, *num\_retries=0*)**

Tells Cozmo to “pop a wheelie” using a light cube.

| **Parameters:** | * **target\_object** ([**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) – The cube to push down on with cozmo’s lift, to start the wheelie. * **approach\_angle** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – The angle to approach the cube from. For example, 180 degrees will cause cozmo to drive past the cube and approach it from behind. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.PopAWheelie**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PopAWheelie) **action object which can be queried**  to see when it is complete. |

**pop\_a\_wheelie\_factory*= functools.partial(<class 'cozmo.robot.PopAWheelie'>, loop=None)***

The factory function that returns a [**PopAWheelie**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.PopAWheelie) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**pose**

The current pose (position and orientation) of Cozmo

| **Type:** | [**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose) |
| --- | --- |

**pose\_angle**

Cozmo’s pose angle (heading in X-Y plane).

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**pose\_pitch**

Cozmo’s pose pitch (angle up/down).

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**right\_wheel\_speed*= None***

Speed of the right wheel

| **Type:** | [**cozmo.util.Speed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Speed) |
| --- | --- |

**robot\_id*= None***

The internal ID number of the robot.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**roll\_cube(*target\_object*, *approach\_angle=None*, *check\_for\_object\_on\_top=False*, *in\_parallel=False*, *num\_retries=0*)**

Tells Cozmo to roll a specified cube object.

| **Parameters:** | * **target\_object** ([**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube)) – The cube to roll. * **approach\_angle** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – The angle to approach the cube from. For example, 180 degrees will cause cozmo to drive past the cube and approach it from behind. * **check\_for\_object\_on\_top** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – If there is a cube on top of the specified cube, and check\_for\_object\_on\_top is True, then Cozmo will ignore the action. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.RollCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.RollCube) **action object which can be queried**  to see when it is complete. |

**roll\_cube\_factory*= functools.partial(<class 'cozmo.robot.RollCube'>, loop=None)***

The factory function that returns a [**RollCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.RollCube) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**run\_timed\_behavior(*behavior\_type*, *active\_time*)**

Executes a behavior for a set number of seconds.

This call blocks and stops the behavior after active\_time seconds.

| **Parameters:** | * **behavior\_type** (**cozmo.behavior.\_BehaviorType**) – An attribute of [**cozmo.behavior.BehaviorTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.BehaviorTypes). * **active\_time** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – specifies the maximum time to execute in seconds |
| --- | --- |
| **Raises:** | [**TypeError**](https://docs.python.org/3.5/library/exceptions.html#TypeError) if an invalid behavior type is supplied. |

**say\_text(*text*, *play\_excited\_animation=False*, *use\_cozmo\_voice=True*, *duration\_scalar=1.0*, *voice\_pitch=0.0*, *in\_parallel=False*, *num\_retries=0*)**

Have Cozmo say text!

| **Parameters:** | * **text** (*string*) – The words for Cozmo to say. * **play\_excited\_animation** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Whether to also play an excited animation while speaking (moves Cozmo a lot). * **use\_cozmo\_voice** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Whether to use Cozmo’s robot voice (otherwise, he uses a generic human male voice). * **duration\_scalar** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Adjust the relative duration of the generated text to speech audio. * **voice\_pitch** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Adjust the pitch of Cozmo’s robot voice [-1.0, 1.0] * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.SayText**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SayText) **action object which can be**  queried to see when it is complete |

**say\_text\_factory*= functools.partial(<class 'cozmo.robot.SayText'>, loop=None)***

The factory function that returns a [**SayText**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SayText) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**serial**

The serial number, as a hex-string (e.g “02e08032”), for the robot.

This matches the Cozmo Serial value in the About section of the settings menu in the app.

| **Type:** | string |
| --- | --- |

**set\_all\_backpack\_lights(*light*)**

Set the lights on Cozmo’s backpack to the same color.

| **Parameters:** | **light** ([**cozmo.lights.Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)) – The lights for Cozmo’s backpack. |
| --- | --- |

**set\_backpack\_lights(*light1*, *light2*, *light3*, *light4*, *light5*)**

Set the lights on Cozmo’s backpack.

The light descriptions below are all from Cozmo’s perspective.

Note: The left and right lights only contain red LEDs, so e.g. setting them to green will look off, and setting them to white will look red

| **Parameters:** | * **light1** ([**cozmo.lights.Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)) – The left backpack light * **light2** ([**cozmo.lights.Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)) – The front backpack light * **light3** ([**cozmo.lights.Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)) – The center backpack light * **light4** ([**cozmo.lights.Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)) – The rear backpack light * **light5** ([**cozmo.lights.Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)) – The right backpack light |
| --- | --- |

**set\_backpack\_lights\_off()**

Set the lights on Cozmo’s backpack to off.

**set\_center\_backpack\_lights(*light*)**

Set the lights in the center of Cozmo’s backpack to the same color.

Forces the lights on the left and right to off (this is useful as those lights only support shades of red, so cannot generally be set to the same color as the center lights).

| **Parameters:** | **light** ([**cozmo.lights.Light**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.lights.html#cozmo.lights.Light)) – The lights for Cozmo’s backpack. |
| --- | --- |

**set\_head\_angle(*angle*, *accel=10.0*, *max\_speed=10.0*, *duration=0.0*, *warn\_on\_clamp=True*, *in\_parallel=False*, *num\_retries=0*)**

Tell Cozmo’s head to turn to a given angle.

| **Parameters:** | * **angle** – ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)): Desired angle for Cozmo’s head. ([**MIN\_HEAD\_ANGLE**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.MIN_HEAD_ANGLE) to [**MAX\_HEAD\_ANGLE**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.MAX_HEAD_ANGLE)). * **accel** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Acceleration of Cozmo’s head in radians per second squared. * **max\_speed** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Maximum speed of Cozmo’s head in radians per second. * **duration** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Time for Cozmo’s head to turn in seconds. A value of zero will make Cozmo try to do it as quickly as possible. * **warn\_on\_clamp** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to log a warning if the angle had to be clamped to the valid range ([**MIN\_HEAD\_ANGLE**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.MIN_HEAD_ANGLE) to [**MAX\_HEAD\_ANGLE**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.MAX_HEAD_ANGLE)). * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.SetHeadAngle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SetHeadAngle) **action object which can be**  queried to see when it is complete |

**set\_head\_angle\_factory*= functools.partial(<class 'cozmo.robot.SetHeadAngle'>, loop=None)***

The factory function that returns a [**SetHeadAngle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SetHeadAngle) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**set\_head\_light(*enable*)**

Turn Cozmo’s IR headlight on or off.

The headlight is on the front of Cozmo’s chassis, between his two front wheels, underneath his head. Cozmo’s camera is IR sensitive so although you cannot see the IR light with the naked eye you will see it in Cozmo’s camera feed.

| **Parameters:** | **enable** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True turns the light on, False turns it off. |
| --- | --- |

**set\_idle\_animation(*anim\_trigger*)**

Set the Idle Animation on Cozmo

Idle animations keep Cozmo alive inbetween the times other animations play. They behave the same as regular animations except that they loop forever until another animation is started.

| **Parameters:** | **anim\_trigger** ([**cozmo.anim.Triggers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Triggers)) – The animation trigger to set |
| --- | --- |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) if supplied an invalid animation trigger. |

**set\_lift\_height(*height*, *accel=10.0*, *max\_speed=10.0*, *duration=0.0*, *in\_parallel=False*, *num\_retries=0*)**

Tell Cozmo’s lift to move to a given height

| **Parameters:** | * **height** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – desired height for Cozmo’s lift 0.0 (bottom) to 1.0 (top) (we clamp it to this range internally). * **accel** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Acceleration of Cozmo’s lift in radians per second squared. * **max\_speed** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Maximum speed of Cozmo’s lift in radians per second. * **duration** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Time for Cozmo’s lift to move in seconds. A value of zero will make Cozmo try to do it as quickly as possible. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.SetLiftHeight**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SetLiftHeight) **action object which can be**  queried to see when it is complete. |

**set\_lift\_height\_factory*= functools.partial(<class 'cozmo.robot.SetLiftHeight'>, loop=None)***

The factory function that returns a [**SetLiftHeight**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.SetLiftHeight) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**set\_manual\_exposure(*exposure\_ms*, *gain*)**

*Deprecated since version 0.12.0:* Use: [**cozmo.camera.Camera.set\_manual\_exposure()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera.set_manual_exposure) instead.

**set\_needs\_levels(*repair\_value=1*, *energy\_value=1*, *play\_value=1*)**

Manually set Cozmo’s current needs levels.

The needs levels control whether Cozmo needs repairing, feeding or playing with. Values outside of the 0.0 to 1.0 range are clamped internally.

| **Parameters:** | * **repair\_value** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – How repaired is Cozmo - 0=’broken’, 1=’fully repaired’ * **energy\_value** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – How energetic is Cozmo - 0=’no-energy’, 1=’full energy’ * **play\_value** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – How in need of play is Cozmo - 0=’bored’, 1=’happy’ |
| --- | --- |

**set\_robot\_volume(*robot\_volume*)**

Set the volume for the speaker in the robot.

| **Parameters:** | **robot\_volume** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The new volume (0.0 = mute, 1.0 = max). |
| --- | --- |

**start\_behavior(*behavior\_type*)**

Starts executing a behavior.

Call the [**stop()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.Behavior.stop) method on the behavior object at some point in the future to terminate execution.

| **Parameters:** | **behavior\_type** (**cozmo.behavior.\_BehaviorType**) – An attribute of [**cozmo.behavior.BehaviorTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.BehaviorTypes). |
| --- | --- |
| **Returns:** | [**cozmo.behavior.Behavior**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.behavior.html#cozmo.behavior.Behavior) |
| **Raises:** | [**TypeError**](https://docs.python.org/3.5/library/exceptions.html#TypeError) if an invalid behavior type is supplied. |

**start\_freeplay\_behaviors()**

Start running freeplay behaviors on Cozmo

Puts Cozmo into a freeplay mode where he autonomously drives around and does stuff based on his mood and environment.

You shouldn’t attempt to drive Cozmo during this, as it will clash with whatever the current behavior is attempting to do.

**stop\_all\_motors()**

Tell Cozmo to stop all motors.

**stop\_freeplay\_behaviors()**

Stop running freeplay behaviors on Cozmo

Forces Cozmo out of Freeplay mode and stops any currently running behaviors and actions.

**turn\_in\_place(*angle*, *in\_parallel=False*, *num\_retries=0*, *speed=None*, *accel=None*, *angle\_tolerance=None*, *is\_absolute=False*)**

Turn the robot around its current position.

| **Parameters:** | * **angle** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – The angle to turn. Positive values turn to the left, negative values to the right. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. * **speed** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – Angular turn speed (per second). * **accel** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – Acceleration of angular turn (per second squared). * **angle\_tolerance** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – angular tolerance to consider the action complete (this is clamped to a minimum of 2 degrees internally). * **is\_absolute** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to turn to a specific angle, False to turn relative to the current pose. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.TurnInPlace**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.TurnInPlace) **action object which can be**  queried to see when it is complete. |

**turn\_in\_place\_factory*= functools.partial(<class 'cozmo.robot.TurnInPlace'>, loop=None)***

The factory function that returns a [**TurnInPlace**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.TurnInPlace) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**turn\_towards\_face(*face*, *in\_parallel=False*, *num\_retries=0*)**

Tells Cozmo to turn towards this face.

| **Parameters:** | * **face** – ([**cozmo.faces.Face**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face)): The face Cozmo will turn towards. * **in\_parallel** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True to run this action in parallel with previous actions, False to require that all previous actions be already complete. * **num\_retries** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – Number of times to retry the action if the previous attempt(s) failed. |
| --- | --- |
| **Returns:** | **A** [**cozmo.robot.TurnTowardsFace**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.TurnTowardsFace) **action object which can be**  queried to see when it is complete |

**turn\_towards\_face\_factory*= functools.partial(<class 'cozmo.robot.TurnTowardsFace'>, loop=None)***

The factory function that returns a [**TurnTowardsFace**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.TurnTowardsFace) class or subclass instance.

| **Type:** | callable |
| --- | --- |

**wait\_for\_all\_actions\_completed()**

Waits until all SDK-initiated actions are complete.

**world*= None***

Tracks state information about Cozmo’s world.

| **Type:** | [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) |
| --- | --- |

**world\_factory*= functools.partial(<class 'cozmo.world.World'>, loop=None)***

The factory function that returns a [**cozmo.world.World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) class or subclass instance.

| **Type:** | callable |
| --- | --- |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot_alignment.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html)

# **cozmo.robot\_alignment**

RobotAlignment related classes, functions, events and values.

**Classes**

| [**RobotAlignmentTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot_alignment.html#cozmo.robot_alignment.RobotAlignmentTypes) | Defines all robot alignment types. |
| --- | --- |

***class*cozmo.robot\_alignment.RobotAlignmentTypes**

Defines all robot alignment types.

**Body*= \_RobotAlignmentType(name='Body', id=2)***

Align the front of cozmo’s body (Useful for when the lift is up)

**Custom*= \_RobotAlignmentType(name='Custom', id=3)***

For use with distanceFromMarker parameter

**LiftFinger*= \_RobotAlignmentType(name='LiftFinger', id=0)***

Align the tips of the lift fingers with the target object

**LiftPlate*= \_RobotAlignmentType(name='LiftPlate', id=1)***

Align the flat part of the lift with the object (Useful for getting the fingers in the cube’s grooves)

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html)

# **cozmo.run**

The run module contains helper classes and functions for opening a connection to the engine.

To get started, the [**run\_program()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.run_program) function can be used for most cases, it handles connecting to a device and then running the function you provide with the SDK-provided Robot object passed in.

The [**connect()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect) function can be used to open a connection and run your own code connected to a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) instance. It takes care of setting up an event loop, finding the Android or iOS device running the Cozmo app and making sure the connection is ok.

You can also use the [**connect\_with\_tkviewer()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect_with_tkviewer) or [**connect\_with\_3dviewer()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect_with_3dviewer) functions which works in a similar way to [**connect()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect), but will also display either a a window on the screen showing a view from Cozmo’s camera (using Tk), or a 3d viewer (with optional 2nd window showing Cozmo’s camera) (using OpenGL), if supported on your system.

Finally, more advanced progarms can integrate the SDK with an existing event loop by using the [**connect\_on\_loop()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect_on_loop) function.

All of these functions make use of a [**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.DeviceConnector) subclass to deal with actually connecting to an Android or iOS device. There shouldn’t normally be a need to modify them or write your own.

**Functions**

| [**connect**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect)(f[, conn\_factory, connector]) | Connects to the Cozmo Engine on the mobile device and supplies the connection to a function. |
| --- | --- |
| [**connect\_on\_loop**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect_on_loop)(loop[, conn\_factory, connector]) | Uses the supplied event loop to connect to a device. |
| [**connect\_with\_3dviewer**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect_with_3dviewer)(f[, conn\_factory, …]) | Setup a connection to a device and run a user function while displaying Cozmo’s 3d world. |
| [**connect\_with\_tkviewer**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect_with_tkviewer)(f[, conn\_factory, …]) | Setup a connection to a device and run a user function while displaying Cozmo’s camera. |
| [**run\_program**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.run_program)(f[, use\_viewer, conn\_factory, …]) | Connect to Cozmo and run the provided program/function f. |
| [**setup\_basic\_logging**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.setup_basic_logging)([general\_log\_level, …]) | Helper to perform basic setup of the Python logging machinery. |

**Classes**

| [**AndroidConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.AndroidConnector)([adb\_cmd, serial]) | Connects to an attached Android device over USB. |
| --- | --- |
| [**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.DeviceConnector)([cozmo\_port, enable\_env\_vars]) | Base class for objects that setup the physical connection to a device. |
| **FirstAvailableConnector**() | Connects to the first Android or iOS device running the Cozmo app in SDK mode. |
| [**IOSConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.IOSConnector)([serial]) | Connects to an attached iOS device over USB. |
| [**TCPConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.TCPConnector)([tcp\_port, ip\_addr]) | Connects to the Cozmo app directly via TCP. |

***class*cozmo.run.DeviceConnector(*cozmo\_port=5106*, *enable\_env\_vars=True*)**

Base class for objects that setup the physical connection to a device.

**connect(*loop*, *protocol\_factory*, *conn\_check*)**

Connect attempts to open a connection transport to the Cozmo app on a device.

On opening a transport it will create a protocol from the supplied factory and connect it to the transport, returning a (transport, protocol) tuple. See **asyncio.BaseEventLoop.create\_connection()**

***class*cozmo.run.IOSConnector(*serial=None*, *\*\*kw*)**

Connects to an attached iOS device over USB.

Opens a connection to the first iOS device that’s found to be running the Cozmo app in SDK mode.

iTunes (or another service providing usbmuxd) must be installed in order for this connector to be able to open a connection to a device.

An instance of this class can be passed to the **connect\_** prefixed functions in this module.

| **Parameters:** | **serial** (*string*) – Serial number of the device to connect to. If None, then connect to the first available iOS device running the Cozmo app in SDK mode. |
| --- | --- |

**connect(*loop*, *protocol\_factory*, *conn\_check*)**

Connect attempts to open a connection transport to the Cozmo app on a device.

On opening a transport it will create a protocol from the supplied factory and connect it to the transport, returning a (transport, protocol) tuple. See **asyncio.BaseEventLoop.create\_connection()**

***class*cozmo.run.AndroidConnector(*adb\_cmd=None*, *serial=None*, *\*\*kw*)**

Connects to an attached Android device over USB.

This requires the Android Studio command line tools to be installed, specifically adb.

By default the connector will attempt to locate adb (or adb.exe on Windows) in common locations, but it may also be supplied by setting the **ANDROID\_ADB\_PATH** environment variable, or by passing it to the constructor.

An instance of this class can be passed to the **connect\_** prefixed functions in this module.

| **Parameters:** | **serial** (*string*) – Serial number of the device to connect to. If None, then connect to the first available Android device running the Cozmo app in SDK mode. |
| --- | --- |

**connect(*loop*, *protocol\_factory*, *conn\_check*)**

Connect attempts to open a connection transport to the Cozmo app on a device.

On opening a transport it will create a protocol from the supplied factory and connect it to the transport, returning a (transport, protocol) tuple. See **asyncio.BaseEventLoop.create\_connection()**

***class*cozmo.run.TCPConnector(*tcp\_port=None*, *ip\_addr='127.0.0.1'*, *\*\*kw*)**

Connects to the Cozmo app directly via TCP.

Generally only used for testing and debugging.

Requires that a SDK\_TCP\_PORT environment variable be set to the port number to connect to.

**connect(*loop*, *protocol\_factory*, *conn\_check*)**

Connect attempts to open a connection transport to the Cozmo app on a device.

On opening a transport it will create a protocol from the supplied factory and connect it to the transport, returning a (transport, protocol) tuple. See **asyncio.BaseEventLoop.create\_connection()**

**cozmo.run.connect(*f*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*)**

Connects to the Cozmo Engine on the mobile device and supplies the connection to a function.

Accepts a function, f, that is given a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) object as a parameter.

The supplied function may be either an asynchronous coroutine function (normally defined using **async def**) or a regular synchronous function.

If an asynchronous function is supplied it will be run on the same thread as the Cozmo event loop and must use the **await** keyword to yield control back to the loop.

If a synchronous function is supplied then it will run on the main thread and Cozmo’s event loop will run on a separate thread. Calls to asynchronous methods returned from CozmoConnection will automatically be translated to synchronous ones.

The connect function will return once the supplied function has completed, as which time it will terminate the connection to the robot.

| **Parameters:** | * **f** (*callable*) – The function to execute * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. |
| --- | --- |

**cozmo.run.connect\_with\_3dviewer(*f*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*, *enable\_camera\_view=False*, *show\_viewer\_controls=True*)**

Setup a connection to a device and run a user function while displaying Cozmo’s 3d world.

This displays an OpenGL window on the screen with a 3D view of Cozmo’s understanding of the world. Optionally, if use\_viewer is True, a 2nd OpenGL window will also display showing a view of Cozmo’s camera. It will return an error if the current system does not support PyOpenGL.

The function may be either synchronous or asynchronous (defined used **async def**).

The function must accept a **cozmo.CozmoConnection** object as its only argument. This call will block until the supplied function completes.

| **Parameters:** | * **f** (*callable*) – The function to execute * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. * **enable\_camera\_view** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to also open a 2D camera view in a second OpenGL window. * **show\_viewer\_controls** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to draw controls on the view. |
| --- | --- |

**cozmo.run.connect\_with\_tkviewer(*f*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*, *force\_on\_top=False*)**

Setup a connection to a device and run a user function while displaying Cozmo’s camera.

This displays a Tk window on the screen showing a view of Cozmo’s camera. It will return an error if the current system does not support Tk.

The function may be either synchronous or asynchronous (defined used **async def**).

The function must accept a **cozmo.CozmoConnection** object as its only argument. This call will block until the supplied function completes.

| **Parameters:** | * **f** (*callable*) – The function to execute * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. * **force\_on\_top** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether the window should be forced on top of all others |
| --- | --- |

**cozmo.run.connect\_on\_loop(*loop*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*)**

Uses the supplied event loop to connect to a device.

Will run the event loop in the current thread until the connection succeeds or fails.

If you do not want/need to manage your own loop, then use the [**connect()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect) function to handle setup/teardown and execute a user-supplied function.

| **Parameters:** | * **loop** ([**asyncio.BaseEventLoop**](https://docs.python.org/3.5/library/asyncio-eventloop.html#asyncio.BaseEventLoop)) – The event loop to use to connect to Cozmo. * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default, it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. |
| --- | --- |
| **Returns:** | A [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) instance. |

**cozmo.run.run\_program(*f*, *use\_viewer=False*, *conn\_factory=<class 'cozmo.conn.CozmoConnection'>*, *connector=None*, *force\_viewer\_on\_top=False*, *deprecated\_filter='default'*, *use\_3d\_viewer=False*, *show\_viewer\_controls=True*, *exit\_on\_connection\_error=True*)**

Connect to Cozmo and run the provided program/function f.

| **Parameters:** | * **f** (*callable*) – The function to execute, accepts a connected [**cozmo.robot.Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) as the parameter. * **use\_viewer** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to display a view of Cozmo’s camera in a window. * **conn\_factory** (*callable*) – Override the factory function to generate a [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) (or subclass) instance. * **connector** ([**DeviceConnector**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.DeviceConnector)) – Optional instance of a DeviceConnector subclass that handles opening the USB connection to a device. By default it will connect to the first Android or iOS device that has the Cozmo app running in SDK mode. * **force\_viewer\_on\_top** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether the window should be forced on top of all others (only relevant if use\_viewer is True). Note that this is ignored if use\_3d\_viewer is True (as it’s not currently supported on that windowing system). * **deprecated\_filter** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – The filter for any DeprecationWarning messages. This is defaulted to “default” which shows the warning once per location. You can hide all deprecated warnings by passing in “ignore”, see <https://docs.python.org/3/library/warnings.html#warning-filter> for more information. * **use\_3d\_viewer** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to display a 3D view of Cozmo’s understanding of the world in a window. Note that if both this and use\_viewer are set then the 2D camera view will render in an OpenGL window instead of a TkView window. * **show\_viewer\_controls** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to draw controls on the view. * **exit\_on\_connection\_error** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specify whether the program should exit on connection error or should an error be raised. Default to true. |
| --- | --- |

**cozmo.run.setup\_basic\_logging(*general\_log\_level=None*, *protocol\_log\_level=None*, *protocol\_log\_messages='all'*, *target=<\_io.TextIOWrapper name='<stderr>' mode='w' encoding='UTF-8'>*, *deprecated\_filter='default'*)**

Helper to perform basic setup of the Python logging machinery.

The SDK defines two loggers:

* **logger** (“cozmo.general”) - For general purpose information about events within the SDK; and
* **logger\_protocol** (“cozmo.protocol”) - For low level communication messages between the device and the SDK.

Generally only **logger** is interesting.

| **Parameters:** | * **general\_log\_level** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – ‘DEBUG’, ‘INFO’, ‘WARN’, ‘ERROR’ or an equivalent constant from the [**logging**](https://docs.python.org/3.5/library/logging.html#module-logging) module. If None then a value will be read from the COZMO\_LOG\_LEVEL environment variable. * **protocol\_log\_level** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – as general\_log\_level. If None then a value will be read from the COZMO\_PROTOCOL\_LOG\_LEVEL environment variable. * **protocol\_log\_messages** ([*list*](https://docs.python.org/3.5/library/stdtypes.html#list)) – The low level messages that should be logged to the protocol log. Defaults to all. Will read from the COMZO\_PROTOCOL\_LOG\_MESSAGES if available which should be a comma separated list of message names (case sensitive). * **target** ([*object*](https://docs.python.org/3.5/library/functions.html#object)) – The stream to send the log data to; defaults to stderr * **deprecated\_filter** ([*str*](https://docs.python.org/3.5/library/stdtypes.html#str)) – The filter for any DeprecationWarning messages. This is defaulted to “default” which shows the warning once per location. You can hide all deprecated warnings by passing in “ignore”, see <https://docs.python.org/3/library/warnings.html#warning-filter> for more information. |
| --- | --- |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot_alignment.html)

# **cozmo.song**

Song related classes, functions, events and values.

**Classes**

| [**NoteDurations**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html#cozmo.song.NoteDurations) | The possible values for a NoteDuration. |
| --- | --- |
| [**NoteTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html#cozmo.song.NoteTypes) | The possible values for an NoteType. |
| [**SongNote**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html#cozmo.song.SongNote)([noteType, id, noteDuration, id]) | Represents on element in a song. |

***class*cozmo.song.NoteTypes**

The possible values for an NoteType.

A pitch between C2 and C3\_Sharp can be specified, as well as a rest (for timed silence), giving cozmo a vocal range of slightly more than one octave.

B\_Flat and E\_Flat are represented as their corresponding sharps.

**A2*= \_NoteType(name='A2', id=9)***

**A2\_Sharp*= \_NoteType(name='A2\_Sharp', id=10)***

**B2*= \_NoteType(name='B2', id=11)***

**C2*= \_NoteType(name='C2', id=0)***

**C2\_Sharp*= \_NoteType(name='C2\_Sharp', id=1)***

**C3*= \_NoteType(name='C3', id=12)***

**C3\_Sharp*= \_NoteType(name='C3\_Sharp', id=13)***

**D2*= \_NoteType(name='D2', id=2)***

**D2\_Sharp*= \_NoteType(name='D2\_Sharp', id=3)***

**E2*= \_NoteType(name='E2', id=4)***

**F2*= \_NoteType(name='F2', id=5)***

**F2\_Sharp*= \_NoteType(name='F2\_Sharp', id=6)***

**G2*= \_NoteType(name='G2', id=7)***

**G2\_Sharp*= \_NoteType(name='G2\_Sharp', id=8)***

**Rest*= \_NoteType(name='Rest', id=14)***

***class*cozmo.song.NoteDurations**

The possible values for a NoteDuration.

**Half*= \_NoteDuration(name='Half', id=2)***

**Quarter*= \_NoteDuration(name='Quarter', id=3)***

**ThreeQuarter*= \_NoteDuration(name='ThreeQuarter', id=1)***

**Whole*= \_NoteDuration(name='Whole', id=0)***

***class*cozmo.song.SongNote(*noteType=\_NoteType(name='C2'*, *id=0)*, *noteDuration=\_NoteDuration(name='Whole'*, *id=0)*)**

Represents on element in a song. Consists of a [**cozmo.song.NoteTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html#cozmo.song.NoteTypes) which specifies either a pitch or rest, and a [**cozmo.song.NoteDurations**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html#cozmo.song.NoteDurations) specifying the length of the note.

# **cozmo.tkview**

This module provides a simple GUI viewer for Cozmo’s camera.

It uses Tkinter, the standard Python GUI toolkit which is optionally available on most platforms, and also depends on the Pillow and numpy libraries for image processing.

See the online SDK documentation for details on how to install these extra packages on your platform.

The easiest way to make use of this viewer is to call [**cozmo.run.connect\_with\_tkviewer()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.run.html#cozmo.run.connect_with_tkviewer).

**Warning**

This package requires Python to have Tkinter installed to display the GUI.

**Classes**

| [**TkImageViewer**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.tkview.html#cozmo.tkview.TkImageViewer)([tk\_root, refresh\_interval, …]) | Simple Tkinter camera viewer. |
| --- | --- |
| **TkThreadable**(\*a, \*\*kw) | A mixin for adding threadsafe calls to tkinter methods. |

***class*cozmo.tkview.TkImageViewer(*tk\_root=None*, *refresh\_interval=10*, *image\_scale=2*, *window\_name='CozmoView'*, *force\_on\_top=True*)**

Simple Tkinter camera viewer.

**configure(*event*)**

Configure resources of a widget.

The values for resources are specified as keyword arguments. To get an overview about the allowed keyword arguments call the method keys.

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.song.html)

# **cozmo.util**

Utility classes and functions.

**Functions**

| [**angle\_z\_to\_quaternion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.angle_z_to_quaternion)(angle\_z) | This function converts an angle in the z axis (Euler angle z component) to a quaternion. |
| --- | --- |
| [**degrees**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.degrees)(degrees) | Returns an [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) instance set to the specified number of degrees. |
| [**distance\_inches**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.distance_inches)(distance\_inches) | Returns an [**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance) instance set to the specified number of inches. |
| [**distance\_mm**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.distance_mm)(distance\_mm) | Returns an [**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance) instance set to the specified number of millimeters. |
| [**pose\_quaternion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.pose_quaternion)(x, y, z, q0, q1, q2, q3[, …]) | Returns a [**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose) instance set to the pose given in quaternion format. |
| [**pose\_z\_angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.pose_z_angle)(x, y, z, angle\_z[, origin\_id]) | Returns a [**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose) instance set to the pose given in z angle format. |
| [**radians**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.radians)(radians) | Returns an [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) instance set to the specified number of radians. |
| [**rotation\_quaternion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.rotation_quaternion)(q0, q1, q2, q3) | Returns a [**Rotation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Rotation) instance set by a quaternion. |
| [**rotation\_z\_angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.rotation_z_angle)(angle\_z) | Returns a class:Rotation instance set by an angle in the z axis |
| [**speed\_mmps**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.speed_mmps)(speed\_mmps) | Returns an [**cozmo.util.Speed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Speed) instance set to the specified millimeters per second speed |

**Classes**

| [**Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)([radians, degrees]) | Represents an angle. |
| --- | --- |
| [**Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance)([distance\_mm, distance\_inches]) | Represents a distance. |
| [**ImageBox**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.ImageBox) | Defines a bounding box within an image frame. |
| [**Matrix44**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Matrix44)(m00, m10, m20, m30, m01, m11, m21, …) | A 4x4 Matrix for representing the rotation and/or position of an object in the world. |
| [**Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose)(x, y, z[, q0, q1, q2, q3, angle\_z, …]) | Represents where an object is in the world. |
| [**Position**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Position)(x, y, z) | Represents the position of an object in the world. |
| [**Quaternion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Quaternion)([q0, q1, q2, q3, angle\_z]) | Represents the rotation of an object in the world. |
| [**Rotation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Rotation)([q0, q1, q2, q3, angle\_z]) | An alias for [**Quaternion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Quaternion) |
| [**Speed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Speed)([speed\_mmps]) | Represents a speed. |
| [**Timeout**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Timeout)([timeout, use\_inf]) | Utility class to keep track of a timeout condition. |
| [**Vector2**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector2)(x, y) | Represents a 2D Vector (type/units aren’t specified) |
| [**Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3)(x, y, z) | Represents a 3D Vector (type/units aren’t specified) |

***class*cozmo.util.Angle(*radians=None*, *degrees=None*)**

Represents an angle.

Use the [**degrees()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.degrees) or [**radians()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.radians) convenience methods to generate an Angle instance.

| **Parameters:** | * **radians** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The number of radians the angle should represent (cannot be combined with **degrees**) * **degrees** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The number of degress the angle should represent (cannot be combined with **radians**) |
| --- | --- |

**abs\_value**

The absolute value of the angle.

If the Angle is positive then it returns a copy of this Angle, otherwise it returns -Angle.

| **Type:** | [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) |
| --- | --- |

**degrees**

The angle in degrees.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**radians**

The angle in radians.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**cozmo.util.degrees(*degrees*)**

Returns an [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) instance set to the specified number of degrees.

**cozmo.util.radians(*radians*)**

Returns an [**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle) instance set to the specified number of radians.

***class*cozmo.util.ImageBox**

Defines a bounding box within an image frame.

This is used when objects, faces and pets are observed to denote where in the robot’s camera view the object, face or pet actually appears. It’s then used by the [**cozmo.annotate**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#module-cozmo.annotate) module to show an outline of a box around the object, face or pet.

**width**

float - The width of the box.

**height**

float - The height of the box.

**bottom\_y**

The y coordinate of the bottom of the box.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**center**

The x,y coordinates of the center of the box.

| **Type:** | ([float](https://docs.python.org/3.5/library/functions.html#float), [float](https://docs.python.org/3.5/library/functions.html#float)) |
| --- | --- |

**left\_x**

The x coordinate of the left of the box.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**right\_x**

The x coordinate of the right of the box.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**top\_y**

The y coordinate of the top of the box.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.util.Distance(*distance\_mm=None*, *distance\_inches=None*)**

Represents a distance.

The class allows distances to be returned in either millimeters or inches.

Use the [**distance\_inches()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.distance_inches) or [**distance\_mm()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.distance_mm) convenience methods to generate a Distance instance.

| **Parameters:** | * **distance\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The number of millimeters the distance should represent (cannot be combined with **distance\_inches**). * **distance\_inches** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The number of inches the distance should represent (cannot be combined with **distance\_mm**). |
| --- | --- |

**distance\_inches**

The distance in inches

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**distance\_mm**

The distance in millimeters

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**cozmo.util.distance\_mm(*distance\_mm*)**

Returns an [**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance) instance set to the specified number of millimeters.

**cozmo.util.distance\_inches(*distance\_inches*)**

Returns an [**cozmo.util.Distance**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Distance) instance set to the specified number of inches.

***class*cozmo.util.Matrix44(*m00*, *m10*, *m20*, *m30*, *m01*, *m11*, *m21*, *m31*, *m02*, *m12*, *m22*, *m32*, *m03*, *m13*, *m23*, *m33*)**

A 4x4 Matrix for representing the rotation and/or position of an object in the world.

Can be generated from a [**Quaternion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Quaternion) for a pure rotation matrix, or combined with a position for a full translation matrix, as done by [**Pose.to\_matrix()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose.to_matrix).

**forward\_xyz**

The x,y,z components representing the matrix’s forward vector.

| **Type:** | tuple of 3 floats |
| --- | --- |

**in\_column\_order**

The contents of the matrix in column order.

| **Type:** | tuple of 16 floats |
| --- | --- |

**in\_row\_order**

The contents of the matrix in row order.

| **Type:** | tuple of 16 floats |
| --- | --- |

**left\_xyz**

The x,y,z components representing the matrix’s left vector.

| **Type:** | tuple of 3 floats |
| --- | --- |

**pos\_xyz**

The x,y,z components representing the matrix’s position vector.

| **Type:** | tuple of 3 floats |
| --- | --- |

**set\_forward(*x*, *y*, *z*)**

Set the x,y,z components representing the matrix’s forward vector.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The X component. * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The Y component. * **z** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The Z component. |
| --- | --- |

**set\_left(*x*, *y*, *z*)**

Set the x,y,z components representing the matrix’s left vector.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The X component. * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The Y component. * **z** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The Z component. |
| --- | --- |

**set\_pos(*x*, *y*, *z*)**

Set the x,y,z components representing the matrix’s position vector.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The X component. * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The Y component. * **z** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The Z component. |
| --- | --- |

**set\_up(*x*, *y*, *z*)**

Set the x,y,z components representing the matrix’s up vector.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The X component. * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The Y component. * **z** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The Z component. |
| --- | --- |

**tabulated\_string**

A multi-line string formatted with tabs to show the matrix contents.

| **Type:** | [str](https://docs.python.org/3.5/library/stdtypes.html#str) |
| --- | --- |

**up\_xyz**

The x,y,z components representing the matrix’s up vector.

| **Type:** | tuple of 3 floats |
| --- | --- |

***class*cozmo.util.Pose(*x*, *y*, *z*, *q0=None*, *q1=None*, *q2=None*, *q3=None*, *angle\_z=None*, *origin\_id=-1*, *is\_accurate=True*)**

Represents where an object is in the world.

Use the :func:’pose\_quaternion’ to return pose in the form of position and rotation defined by a quaternion

Use the :func:’pose\_z\_angle’ to return pose in the form of position and rotation defined by rotation about the z axis

When the engine is initialized, and whenever Cozmo is de-localized (i.e. whenever Cozmo no longer knows where he is - e.g. when he’s picked up) Cozmo creates a new pose starting at (0,0,0) with no rotation, with origin\_id incremented to show that these poses cannot be compared with earlier ones. As Cozmo drives around, his pose (and the pose of other objects he observes - e.g. faces, cubes etc.) is relative to this initial position and orientation.

The coordinate space is relative to Cozmo, where Cozmo’s origin is the point on the ground between Cozmo’s two front wheels:

The X axis is Cozmo’s forward direction The Y axis is to Cozmo’s left The Z axis is up

Only poses of the same origin\_id can safely be compared or operated on

**define\_pose\_relative\_this(*new\_pose*)**

Creates a new pose such that new\_pose’s origin is now at the location of this pose.

| **Parameters:** | **new\_pose** ([**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose)) – The pose which origin is being changed. |
| --- | --- |
| **Returns:** | A **cozmo.util.pose** object for which the origin was this pose’s origin. |

**invalidate()**

Mark this pose as being invalid (unusable)

**is\_accurate**

Returns True if this pose is valid and accurate.

Poses are marked as inaccurate if we detect movement via accelerometer, or if they were observed from far enough away that we’re less certain of the exact pose.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**is\_comparable(*other\_pose*)**

Are these two poses comparable.

Poses are comparable if they’re valid and having matching origin IDs.

| **Parameters:** | **other\_pose** ([**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose)) – The other pose to compare against. |
| --- | --- |
| **Returns:** | True if the two poses are comparable, False otherwise. |
| **Return type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |

**is\_valid**

Returns True if this is a valid, usable pose.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**origin\_id**

An ID maintained by the engine which represents which coordinate frame this pose is in.

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**position**

The position component of this pose.

| **Type:** | [**cozmo.util.Position**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Position) |
| --- | --- |

**rotation**

The rotation component of this pose.

| **Type:** | [**cozmo.util.Rotation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Rotation) |
| --- | --- |

**to\_matrix()**

Convert the Pose to a Matrix44.

| **Returns:** | A matrix representing this Pose’s position and rotation. |
| --- | --- |
| **Return type:** | [**cozmo.util.Matrix44**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Matrix44) |

**cozmo.util.pose\_quaternion(*x*, *y*, *z*, *q0*, *q1*, *q2*, *q3*, *origin\_id=0*)**

Returns a [**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose) instance set to the pose given in quaternion format.

**cozmo.util.pose\_z\_angle(*x*, *y*, *z*, *angle\_z*, *origin\_id=0*)**

Returns a [**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose) instance set to the pose given in z angle format.

***class*cozmo.util.Position(*x*, *y*, *z*)**

Represents the position of an object in the world.

A position consists of its x, y and z values in millimeters.

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – X position in millimeters * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Y position in millimeters * **z** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Z position in millimeters |
| --- | --- |

***class*cozmo.util.Quaternion(*q0=None*, *q1=None*, *q2=None*, *q3=None*, *angle\_z=None*)**

Represents the rotation of an object in the world. Can be generated with quaternion to define its rotation in 3d space, or with only a z axis rotation to define things limited to the x-y plane like Cozmo.

Use the [**rotation\_quaternion()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.rotation_quaternion) to return rotation defined by a quaternion.

Use the **rotation\_angle\_z()** to return rotation defined by an angle in the z axis.

**angle\_z**

The z Euler component of the object’s rotation.

Defined as the rotation in the z axis.

| **Type:** | class |
| --- | --- |
| **Type:** | Angle |

**euler\_angles**

Euler angles of an object.

Returns the pitch, yaw, roll Euler components of the object’s rotation defined as rotations in the x, y, and z axis respectively.

It interprets the rotations performed in the order: Z, Y, X

| **Type:** | tuple of float |
| --- | --- |

**q0**

The q0 (w) value of the quaternion.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**q0\_q1\_q2\_q3**

Contains all elements of the quaternion (q0,q1,q2,q3)

| **Type:** | tuple of float |
| --- | --- |

**q1**

The q1 (i) value of the quaternion.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**q2**

The q2 (j) value of the quaternion.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**q3**

The q3 (k) value of the quaternion.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**to\_matrix(*pos\_x=0.0*, *pos\_y=0.0*, *pos\_z=0.0*)**

Convert the Quaternion to a 4x4 matrix representing this rotation.

A position can also be provided to generate a full translation matrix.

| **Parameters:** | * **pos\_x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The x component for the position. * **pos\_y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The y component for the position. * **pos\_z** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The z component for the position. |
| --- | --- |
| **Returns:** | A matrix representing this Quaternion’s rotation, with the provided position (which defaults to 0,0,0). |
| **Return type:** | [**cozmo.util.Matrix44**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Matrix44) |

***class*cozmo.util.Rotation(*q0=None*, *q1=None*, *q2=None*, *q3=None*, *angle\_z=None*)**

An alias for [**Quaternion**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Quaternion)

**cozmo.util.rotation\_quaternion(*q0*, *q1*, *q2*, *q3*)**

Returns a [**Rotation**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Rotation) instance set by a quaternion.

**cozmo.util.rotation\_z\_angle(*angle\_z*)**

Returns a class:Rotation instance set by an angle in the z axis

**cozmo.util.angle\_z\_to\_quaternion(*angle\_z*)**

This function converts an angle in the z axis (Euler angle z component) to a quaternion.

| **Parameters:** | **angle\_z** ([**cozmo.util.Angle**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Angle)) – The z axis angle. |
| --- | --- |
| **Returns:** | **A tuple with all the members**  of a quaternion defined by angle\_z. |
| **Return type:** | q0,q1,q2,q3 ([float](https://docs.python.org/3.5/library/functions.html#float), [float](https://docs.python.org/3.5/library/functions.html#float), [float](https://docs.python.org/3.5/library/functions.html#float), [float](https://docs.python.org/3.5/library/functions.html#float)) |

***class*cozmo.util.Speed(*speed\_mmps=None*)**

Represents a speed.

This class allows speeds to be measured in millimeters per second.

Use [**speed\_mmps()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.speed_mmps) convenience methods to generate a Speed instance.

| **Parameters:** | **speed\_mmps** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The number of millimeters per second the speed should represent. |
| --- | --- |

**speed\_mmps**

The speed in millimeters per second (mmps).

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**cozmo.util.speed\_mmps(*speed\_mmps*)**

Returns an [**cozmo.util.Speed**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Speed) instance set to the specified millimeters per second speed

***class*cozmo.util.Timeout(*timeout=None*, *use\_inf=False*)**

Utility class to keep track of a timeout condition.

This measures a timeout from the point in time that the class is instantiated.

| **Parameters:** | * **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Amount of time (in seconds) allotted to pass before considering the timeout condition to be met. * **use\_inf** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – If True, then [**remaining**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Timeout.remaining) will return **math.inf** if timeout is None, else it will return None. |
| --- | --- |

**is\_timed\_out**

True if the timeout has expired.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**remaining**

The number of seconds remaining before reaching the timeout.

Will return a number of zero or higher, even if the timer has since expired (it will never return a negative value).

Will return None or math.inf (if **use\_inf** was passed as **True** to the constructor) if the original timeout was **None**.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.util.Vector2(*x*, *y*)**

Represents a 2D Vector (type/units aren’t specified)

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – X component * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Y component |
| --- | --- |

**set\_to(*rhs*)**

Copy the x and y components of the given vector.

| **Parameters:** | **rhs** ([**Vector2**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector2)) – The right-hand-side of this assignment - the source vector to copy into this vector. |
| --- | --- |

**x**

The x component.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**x\_y**

The X, Y elements of the Vector2 (x,y)

| **Type:** | [tuple](https://docs.python.org/3.5/library/stdtypes.html#tuple) ([float](https://docs.python.org/3.5/library/functions.html#float), [float](https://docs.python.org/3.5/library/functions.html#float)) |
| --- | --- |

**y**

The y component.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

***class*cozmo.util.Vector3(*x*, *y*, *z*)**

Represents a 3D Vector (type/units aren’t specified)

| **Parameters:** | * **x** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – X component * **y** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Y component * **z** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Z component |
| --- | --- |

**set\_to(*rhs*)**

Copy the x, y and z components of the given vector.

| **Parameters:** | **rhs** ([**Vector3**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Vector3)) – The right-hand-side of this assignment - the source vector to copy into this vector. |
| --- | --- |

**x**

The x component.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**x\_y\_z**

The X, Y, Z elements of the Vector3 (x,y,z)

| **Type:** | [tuple](https://docs.python.org/3.5/library/stdtypes.html#tuple) ([float](https://docs.python.org/3.5/library/functions.html#float), [float](https://docs.python.org/3.5/library/functions.html#float), [float](https://docs.python.org/3.5/library/functions.html#float)) |
| --- | --- |

**y**

The y component.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**z**

The z component.

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

[Next](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html)

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.tkview.html)

# **cozmo.world**

The “world” represents the robot’s known view of its environment.

This view includes objects, faces and pets it knows about and can currently “see” with its camera, along with what actions or behaviors the robot is current performing and the images coming back from the camera (if any).

Almost all events emitted by the robot itself, objects, faces, pets and the camera can be observed directly on the [**World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) object, which is itself accessible as [**cozmo.robot.Robot.world**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot.world).

For example, if you only need to know whether a particular cube has been tapped, you can call the [**wait\_for()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.wait_for) method directly on that cube’s [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube) instance. Eg:

my\_cube.wait\_for(cozmo.objects.EvtObjectTapped)

If, however, you want to wait for any cube to be tapped, you could instead call the [**wait\_for()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.event.html#cozmo.event.Dispatcher.wait_for) method on the [**World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) object instead. Eg:

robot.world.wait\_for(cozmo.objects.EvtObjectTapped)

In either case, wait\_for will return the instance of the event’s [**EvtObjectTapped**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectTapped) class, which includes a [**obj**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.EvtObjectTapped.obj) attribute, which identifies exactly which cube has been tapped.

The [**World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World) object also has a [**cozmo.camera.Camera**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera) instance associated with it. It emits [**EvtNewCameraImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.EvtNewCameraImage) objects whenever a new camera image is available (generally up to 15 times per second), which includes the raw image from the camera, as well as an annotated version showing where faces, pets and objects have been observed.

**Note**

The camera must first be enabled to receive images by setting [**image\_stream\_enabled**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.camera.html#cozmo.camera.Camera.image_stream_enabled) to True.

**Classes**

| [**CameraImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.CameraImage)(raw\_image, image\_annotator[, …]) | A single image from Cozmo’s camera. |
| --- | --- |
| [**EvtNewCameraImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.EvtNewCameraImage)(\*\*kwargs) | Dispatched when a new camera image is received and processed from the robot’s camera. |
| [**World**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World)(conn, robot, \*\*kw) | Represents the state of the world, as known to a Cozmo robot. |

***class*cozmo.world.EvtNewCameraImage(*\*\*kwargs*)**

Dispatched when a new camera image is received and processed from the robot’s camera.

**image*= 'A CameraImage object'***

***class*cozmo.world.CameraImage(*raw\_image*, *image\_annotator*, *image\_number=0*)**

A single image from Cozmo’s camera.

This wraps a raw image and provides an [**annotate\_image()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.CameraImage.annotate_image) method that can resize and add dynamic annotations to the image, such as marking up the location of objects, faces and pets.

**annotate\_image(*scale=None*, *fit\_size=None*, *resample\_mode=0*)**

Adds any enabled annotations to the image.

Optionally resizes the image prior to annotations being applied. The aspect ratio of the resulting image always matches that of the raw image.

| **Parameters:** | * **scale** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – If set then the base image will be scaled by the supplied multiplier. Cannot be combined with fit\_size * **fit\_size** (*tuple of int*) – If set, then scale the image to fit inside the supplied (width, height) dimensions. The original aspect ratio will be preserved. Cannot be combined with scale. * **resample\_mode** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The resampling mode to use when scaling the image. Should be either [**RESAMPLE\_MODE\_NEAREST**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.RESAMPLE_MODE_NEAREST) (fast) or [**RESAMPLE\_MODE\_BILINEAR**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.RESAMPLE_MODE_BILINEAR) (slower, but smoother). |
| --- | --- |
| **Returns:** | [**PIL.Image.Image**](https://pillow.readthedocs.io/en/latest/reference/Image.html#PIL.Image.Image) |

**image\_annotator*= None***

the image annotation object

| **Type:** | [**cozmo.annotate.ImageAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator) |
| --- | --- |

**image\_number*= None***

An image number that increments on every new image received

| **Type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| --- | --- |

**image\_recv\_time*= None***

The time the image was received and processed by the SDK

| **Type:** | [float](https://docs.python.org/3.5/library/functions.html#float) |
| --- | --- |

**raw\_image*= None***

the raw unprocessed image from the camera

| **Type:** | [**PIL.Image.Image**](https://pillow.readthedocs.io/en/latest/reference/Image.html#PIL.Image.Image) |
| --- | --- |

***class*cozmo.world.World(*conn*, *robot*, *\*\*kw*)**

Represents the state of the world, as known to a Cozmo robot.

**active\_action**

True if Cozmo is currently executing an action.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**active\_behavior**

True if the robot is currently executing a behavior.

| **Type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |
| --- | --- |

**annotator\_factory*= functools.partial(<class 'cozmo.annotate.ImageAnnotator'>, loop=None)***

The factory function that returns an **annotate.ImageAnnotator** class or subclass instance.

| **Type:** | callable |
| --- | --- |

**auto\_disconnect\_from\_cubes\_at\_end(*enable=True*)**

Tell the SDK to auto disconnect from cubes at the end of every SDK program.

This can be used to save cube battery life if you spend a lot of time in SDK mode but aren’t running programs as much (as you’re busy writing them). Call [**connect\_to\_cubes()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.connect_to_cubes) to re-connect to the cubes later.

| **Parameters:** | **enable** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – True if cubes should disconnect after every SDK program exits. |
| --- | --- |

**charger*= None***

Cozmo’s charger. **None** if no charger connected or known about yet.

| **Type:** | [**cozmo.objects.Charger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.Charger) |
| --- | --- |

**charger\_factory*= functools.partial(<class 'cozmo.objects.Charger'>, loop=None)***

The factory function that returns an **objects.Charger** class or subclass instance.

| **Type:** | callable |
| --- | --- |

**conn*= None***

The underlying connection to a device.

| **Type:** | [**cozmo.conn.CozmoConnection**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.conn.html#cozmo.conn.CozmoConnection) |
| --- | --- |

**connect\_to\_cubes()**

Connect to all cubes.

Request that Cozmo connects to all cubes - this is required if you previously called [**disconnect\_from\_cubes()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.disconnect_from_cubes) or [**auto\_disconnect\_from\_cubes\_at\_end()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.auto_disconnect_from_cubes_at_end) with enable=False. Connecting to a cube can take up to about 5 seconds, and this method will wait until either all 3 cubes are connected, or it has timed out waiting for this.

| **Returns:** | True if all 3 cubes are now connected. |
| --- | --- |
| **Return type:** | [bool](https://docs.python.org/3.5/library/functions.html#bool) |

**connected\_light\_cubes**

yields each LightCube that Cozmo is currently connected to.

| **Returns:** | A generator yielding [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube) instances |
| --- | --- |
| **Type:** | generator |

**create\_custom\_fixed\_object(*pose*, *x\_size\_mm*, *y\_size\_mm*, *z\_size\_mm*, *relative\_to\_robot=False*, *use\_robot\_origin=True*)**

Defines a cuboid of custom size and places it in the world. It cannot be observed.

| **Parameters:** | * **pose** ([**cozmo.util.Pose**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html#cozmo.util.Pose)) – The pose of the object we are creating. * **x\_size\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – size of the object (in millimeters) in the x axis. * **y\_size\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – size of the object (in millimeters) in the y axis. * **z\_size\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – size of the object (in millimeters) in the z axis. * **relative\_to\_robot** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – whether or not the pose given assumes the robot’s pose as its origin. * **use\_robot\_origin** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – whether or not to override the origin\_id in the given pose to be the origin\_id of Cozmo. |
| --- | --- |
| **Returns:** | A [**cozmo.objects.FixedCustomObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.FixedCustomObject) instance with the specified dimensions and pose. |

**custom\_object\_factory*= functools.partial(<class 'cozmo.objects.CustomObject'>, loop=None)***

The factory function that returns an **objects.CustomObject** class or subclass instance.

| **Type:** | callable |
| --- | --- |

**define\_custom\_box(*custom\_object\_type*, *marker\_front*, *marker\_back*, *marker\_top*, *marker\_bottom*, *marker\_left*, *marker\_right*, *depth\_mm*, *width\_mm*, *height\_mm*, *marker\_width\_mm*, *marker\_height\_mm*, *is\_unique=True*)**

Defines a cuboid of custom size and binds it to a specific custom object type.

The engine will now detect the markers associated with this object and send an object\_observed message when they are seen. The markers must be placed in the center of their respective sides. All 6 markers must be unique.

| **Parameters:** | * **custom\_object\_type** ([**cozmo.objects.CustomObjectTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectTypes)) – the object type you are binding this custom object to * **marker\_front** ([**cozmo.objects.CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers)) – the marker affixed to the front of the object * **marker\_back** ([**cozmo.objects.CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers)) – the marker affixed to the back of the object * **marker\_top** ([**cozmo.objects.CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers)) – the marker affixed to the top of the object * **marker\_bottom** ([**cozmo.objects.CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers)) – the marker affixed to the bottom of the object * **marker\_left** ([**cozmo.objects.CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers)) – the marker affixed to the left of the object * **marker\_right** ([**cozmo.objects.CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers)) – the marker affixed to the right of the object * **depth\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – depth of the object (in millimeters) (X axis) * **width\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – width of the object (in millimeters) (Y axis) * **height\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – height of the object (in millimeters) (Z axis) (the height of the object) * **marker\_width\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – width of the printed marker (in millimeters). * **maker\_height\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – height of the printed marker (in millimeters). * **is\_unique** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – If True, the engine will assume there is only 1 of this object (and therefore only 1 of each of any of these markers) in the world. |
| --- | --- |
| **Returns:** | **A cozmo.object.CustomObject instance with the specified dimensions.**  This is None if the definition failed internally. Note: No instances of this object are added to the world until they have been seen. |
| **Raises:** | * TypeError if the custom\_object\_type is of the wrong type. * ValueError if the 6 markers aren’t unique. |

**define\_custom\_cube(*custom\_object\_type*, *marker*, *size\_mm*, *marker\_width\_mm*, *marker\_height\_mm*, *is\_unique=True*)**

Defines a cube of custom size and binds it to a specific custom object type.

The engine will now detect the markers associated with this object and send an object\_observed message when they are seen. The markers must be placed in the center of their respective sides.

| **Parameters:** | * **custom\_object\_type** ([**cozmo.objects.CustomObjectTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectTypes)) – the object type you are binding this custom object to. * **marker** – ([**cozmo.objects.CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers)): the marker affixed to every side of the cube. * **size\_mm** – size of each side of the cube (in millimeters). * **marker\_width\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – width of the printed marker (in millimeters). * **maker\_height\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – height of the printed marker (in millimeters). * **is\_unique** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – If True, the engine will assume there is only 1 of this object (and therefore only 1 of each of any of these markers) in the world. |
| --- | --- |
| **Returns:** | **A cozmo.object.CustomObject instance with the specified dimensions.**  This is None if the definition failed internally. Note: No instances of this object are added to the world until they have been seen. |
| **Raises:** | TypeError if the custom\_object\_type is of the wrong type. |

**define\_custom\_wall(*custom\_object\_type*, *marker*, *width\_mm*, *height\_mm*, *marker\_width\_mm*, *marker\_height\_mm*, *is\_unique=True*)**

Defines a wall of custom width and height, with a fixed depth of 10mm, and binds it to a specific custom object type.

The engine will now detect the markers associated with this object and send an object\_observed message when they are seen. The markers must be placed in the center of their respective sides.

| **Parameters:** | * **custom\_object\_type** ([**cozmo.objects.CustomObjectTypes**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectTypes)) – the object type you are binding this custom object to. * **marker** – ([**cozmo.objects.CustomObjectMarkers**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.CustomObjectMarkers)): the marker affixed to the front and back of the wall * **width\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – width of the object (in millimeters). (Y axis). * **height\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – height of the object (in millimeters). (Z axis). * **width\_mm** – width of the wall (along Y axis) (in millimeters). * **height\_mm** – height of the wall (along Z axis) (in millimeters). * **marker\_width\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – width of the printed marker (in millimeters). * **maker\_height\_mm** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – height of the printed marker (in millimeters). * **is\_unique** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – If True, the engine will assume there is only 1 of this object (and therefore only 1 of each of any of these markers) in the world. |
| --- | --- |
| **Returns:** | **A cozmo.object.CustomObject instance with the specified dimensions.**  This is None if the definition failed internally. Note: No instances of this object are added to the world until they have been seen. |
| **Raises:** | TypeError if the custom\_object\_type is of the wrong type. |

**delete\_all\_custom\_objects()**

Causes the robot to forget about all custom (fixed + marker) objects it currently knows about.

Note: This includes all fixed custom objects, and all custom marker object instances, BUT this does NOT remove the custom marker object definitions, so Cozmo will continue to add new objects if he sees the markers again. To remove the definitions for those objects use: [**undefine\_all\_custom\_marker\_objects()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.undefine_all_custom_marker_objects)

**delete\_custom\_marker\_objects()**

Causes the robot to forget about all custom marker objects it currently knows about.

Note: This removes custom marker object instances only, it does NOT remove fixed custom objects, nor does it remove the custom marker object definitions, so Cozmo will continue to add new objects if he sees the markers again. To remove the definitions for those objects use: [**undefine\_all\_custom\_marker\_objects()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.undefine_all_custom_marker_objects)

**delete\_fixed\_custom\_objects()**

Causes the robot to forget about all fixed custom objects it currently knows about.

Note: This removes fixed custom objects only, it does NOT remove the custom marker object instances or definitions.

**disconnect\_from\_cubes()**

Disconnect from all cubes (to save battery life etc.).

Call [**connect\_to\_cubes()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.connect_to_cubes) to re-connect to the cubes later.

**enable\_block\_tap\_filter(*enable=True*)**

Enable or disable the block tap filter in the engine.

The block (AKA LightCube) tap filter removes low intensity taps, and filters out taps that come in rapidly together and instead just sends the strongest one

| **Parameters:** | **enable** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – specifies whether the filter should be enabled or disabled |
| --- | --- |

**face\_factory*= functools.partial(<class 'cozmo.faces.Face'>, loop=None)***

The factory function that returns a **faces.Face** class or subclass instance.

| **Type:** | callable |
| --- | --- |

**get\_light\_cube(*cube\_id*)**

Returns the light cube with the given cube ID

| **Parameters:** | **cube\_id** ([*int*](https://docs.python.org/3.5/library/functions.html#int)) – The light cube ID - should be one of [**LightCube1Id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube1Id), [**LightCube2Id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube2Id) and [**LightCube3Id**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube3Id). Note: the cube\_id is not the same thing as the object\_id. |
| --- | --- |
| **Returns:** | The LightCube object with that cube\_id |
| **Return type:** | [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube) |
| **Raises:** | [**ValueError**](https://docs.python.org/3.5/library/exceptions.html#ValueError) if the cube\_id is invalid. |

**image\_annotator*= None***

The image annotator used to add annotations to the raw camera images.

| **Type:** | [**cozmo.annotate.ImageAnnotator**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.annotate.html#cozmo.annotate.ImageAnnotator) |
| --- | --- |

**latest\_image*= None***

The latest image received, or None.

| **Type:** | [**CameraImage**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.CameraImage) |
| --- | --- |

**light\_cube\_factory*= functools.partial(<class 'cozmo.objects.LightCube'>, loop=None)***

The factory function that returns an **objects.LightCube** class or subclass instance.

| **Type:** | callable |
| --- | --- |

**nav\_memory\_map**

Returns the latest navigation memory map for Cozmo.

| **Returns:** | **Current navigation**  memory map. This will be none unless you’ve previously called **request\_nav\_memory\_map()** with a positive frequency to request the data be sent over from the engine. |
| --- | --- |
| **Return type:** | [**NavMemoryMapGrid**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.nav_memory_map.html#cozmo.nav_memory_map.NavMemoryMapGrid) |

**pet\_factory*= functools.partial(<class 'cozmo.pets.Pet'>, loop=None)***

The factory function that returns a **pets.Pet** class or subclass instance.

| **Type:** | callable |
| --- | --- |

**request\_nav\_memory\_map(*frequency\_s*)**

Request navigation memory map data from Cozmo.

The memory map can be accessed via [**nav\_memory\_map**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.nav_memory_map), it will be None until [**request\_nav\_memory\_map()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.request_nav_memory_map) has been called and a map has been received. The memory map provides a quad-tree map of where Cozmo thinks there are objects, and where Cozmo thinks it is safe to drive.

| **Parameters:** | **frequency\_s** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – number of seconds between each update being sent. Negative values, e.g. -1.0, will disable any updates being sent. |
| --- | --- |

**robot*= None***

The primary robot

| **Type:** | [**cozmo.robot.Robot**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.robot.html#cozmo.robot.Robot) |
| --- | --- |

**undefine\_all\_custom\_marker\_objects()**

Remove all custom marker object definitions, and any instances of them in the world.

**visible\_face\_count()**

Returns the number of faces that Cozmo can currently see.

| **Returns:** | The number of faces currently visible. |
| --- | --- |
| **Return type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |

**visible\_faces**

yields each face that Cozmo can currently see.

| **Returns:** | A generator yielding [**cozmo.faces.Face**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face) instances |
| --- | --- |
| **Type:** | generator |

**visible\_object\_count(*object\_type=None*)**

Returns the number of objects that Cozmo can currently see.

| **Parameters:** | **object\_type** ([**ObservableObject**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.ObservableObject) subclass) – Which type of object to count. If None, return the total number of currently visible objects. |
| --- | --- |
| **Returns:** | The number of objects that Cozmo can currently see. |
| **Return type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |

**visible\_objects**

yields each object that Cozmo can currently see.

For faces, see [**visible\_faces()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.visible_faces). For pets, see [**visible\_pets()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.visible_pets).

| **Returns:** | A generator yielding **cozmo.objects.BaseObject** instances |
| --- | --- |
| **Type:** | generator |

**visible\_pet\_count()**

Returns the number of pets that Cozmo can currently see.

| **Returns:** | The number of pets currently visible. |
| --- | --- |
| **Return type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |

**visible\_pets**

yields each pet that Cozmo can currently see.

| **Returns:** | A generator yielding [**cozmo.pets.Pet**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#cozmo.pets.Pet) instances |
| --- | --- |
| **Type:** | generator |

**wait\_for\_observed\_charger(*timeout=None*, *include\_existing=True*)**

Waits for a charger to be observed by the robot.

| **Parameters:** | * **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Number of seconds to wait for a charger to be observed, or None for indefinite * **include\_existing** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to include chargers that are already visible. |
| --- | --- |
| **Returns:** | The [**cozmo.objects.Charger**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.Charger) object that was observed. |

**wait\_for\_observed\_face(*timeout=None*, *include\_existing=True*)**

Waits for a face to be observed by the robot.

| **Parameters:** | * **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Number of seconds to wait for a face to be observed, or None for indefinite * **include\_existing** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to include faces that are already visible. |
| --- | --- |
| **Returns:** | The [**cozmo.faces.Face**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.faces.html#cozmo.faces.Face) object that was observed. |

**wait\_for\_observed\_light\_cube(*timeout=None*, *include\_existing=True*)**

Waits for one of the light cubes to be observed by the robot.

| **Parameters:** | * **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Number of seconds to wait for a cube to be observed, or None for indefinite * **include\_existing** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to include light cubes that are already visible. |
| --- | --- |
| **Returns:** | The [**cozmo.objects.LightCube**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.objects.html#cozmo.objects.LightCube) object that was observed. |

**wait\_for\_observed\_pet(*timeout=None*, *include\_existing=True*)**

Waits for a pet to be observed by the robot.

| **Parameters:** | * **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Number of seconds to wait for a pet to be observed, or None for indefinite * **include\_existing** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to include pets that are already visible. |
| --- | --- |
| **Returns:** | The [**cozmo.pets.Pet**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.pets.html#cozmo.pets.Pet) object that was observed. |

**wait\_until\_num\_objects\_visible(*num*, *object\_type=None*, *timeout=None*)**

Waits for at least a specific number of objects to be seen concurrently.

Unlike [**wait\_until\_observe\_num\_objects()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.wait_until_observe_num_objects) which returns when several objects have become visible, but not necessarily simultaneously, this method will only return if the specific number of objects are visible to the camera at the same time (as defined by **objects.OBJECT\_VISIBILITY\_TIMEOUT**).

| **Parameters:** | * **num** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The number of unique objects to wait for. * **(class** (*object\_type*) – cozmo.objects.ObservableObject): If provided this will cause only the selected object types to be counted. * **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Maximum amount of time in seconds to wait for the requested number of objects to be observed. |
| --- | --- |
| **Returns:** | The number of objects seen (num or higher). |
| **Return type:** | [int](https://docs.python.org/3.5/library/functions.html#int) |
| **Raises:** | asyncio.TimeoutError if the required count wasn’t seen. |

**wait\_until\_observe\_num\_objects(*num*, *object\_type=None*, *timeout=None*, *include\_existing=True*)**

Waits for a certain number of unique objects to be seen at least once.

This method waits for a number of unique objects to be seen, but not necessarily concurrently. That is, if cube 1 appears to the camera and then moves out of view to be replaced by cube 2, then that will count as 2 observed objects.

To wait for multiple objects to be visible simultaneously, see [**wait\_until\_num\_objects\_visible()**](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.world.html#cozmo.world.World.wait_until_num_objects_visible).

| **Parameters:** | * **num** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – The number of unique objects to wait for. * **(class** (*object\_type*) – cozmo.objects.ObservableObject): If provided this will cause only the selected object types to be counted. * **timeout** ([*float*](https://docs.python.org/3.5/library/functions.html#float)) – Maximum amount of time in seconds to wait for the requested number of objects to be observed. * **include\_existing** ([*bool*](https://docs.python.org/3.5/library/functions.html#bool)) – Specifies whether to include objects that are already visible. |
| --- | --- |
| **Returns:** | A list of length <= num of the unique objects class:cozmo.objects.ObservableObject observed during this wait. |

[Previous](https://data.bit-bots.de/cozmo_sdk_doc/cozmosdk.anki.com/docs/generated/cozmo.util.html)