Robotics with **Attitude**

Programming a
Real Life Pixar Character

@timcomport

Media courtesy of **anki.**
and **Disney · PIXAR**
“Any sufficiently advanced technology is indistinguishable from magic.”

Arthur C. Clarke
I'll show you my source code if you show me yours.
The Magic Revealed
Encoders
4 motors
50+ gears
Tank treads
Encoders
Speaker

128 x 64 OLED display
<table>
<thead>
<tr>
<th>Vision System</th>
<th>Face Tracking</th>
<th>Planning / A.I.</th>
<th>Manipulation</th>
<th>Animations</th>
</tr>
</thead>
</table>
| • Face Recognition  
  • Edge Detection  
  • "Interesting Objects“  
  • Marker Detection  
  • Motion Detection  
  • Colour Detection  
  • Camera Streaming | • High Speed Tracking  
  • Position / Orientation  
  • 44 Face Markers  
  • Face Recognition  
  • Age / Gender  
  • Eye-Openness  
  • Facial Expressions | • Realtime Path Planner  
  • Realtime SLAM  
  • World Exploration  
  • Emotion Engine  
  • Behaviour Engine | • 3D Position Tracking  
  • Push  
  • Pickup  
  • Stack  
  • Roll | • 128 x 64 OLED Face  
  • Maya Animations  
  • JSON Animation format  
  • Sound Streaming  
  • 4 DOF |

All the low level access to Robot, Sensors, Cubes, etc.
```python
import sys
import cozmo

def run(sdk_conn):
    '''The run method runs once Cozmo is connected.''
    robot = sdk_conn.wait_for_robot()
    robot.say_text("Hello World").wait_for_completed()

if __name__ == '__main__':
    cozmo.setup_basic_logging()
    try:
        cozmo.connect(run)
    except cozmo.ConnectionError as e:
        sys.exit("A connection error occurred: %s" % e)
```
```python
def run(sdk_conn):
    """The run method runs once Cozmo is connected."""
    robot = sdk_conn.wait_for_robot()

    lookaround = robot.start_behavior(cozmo.behavior.BehaviorTypes.LookAroundInPlace)
cubes = robot.world.wait_until_observe_num_objects(num=2, object_type=cozmo.objects.LightCube)
    lookaround.stop()

    if len(cubes) < 2:
        print("Error: need 2 Cubes but only found", len(cubes), "Cube(s)")
    else:
        current_action = robot.pickup_object(cubes[0])
current_action.wait_for_completed()
        if current_action.has_failed:
            code, reason = current_action.failure_reason
            print("Pickup Cube failed: code=%s reason=%s" % (code, reason))

        current_action = robot.place_on_object(cubes[1])
current_action.wait_for_completed()
        if current_action.has_failed:
            code, reason = current_action.failure_reason
            print("Place On Cube failed: code=%s reason=%s" % (code, reason))
```