API for the modeler implemented in `js/render/core/clay.js`

The root node, called `model`, is the only object that is already created for you.

METHODS:

```
model = model.animate(() => PROCEDURE) // called every animation frame
child = obj.add(type)         // 'sphere'|'tubeX'|'tubeY'|'tubeZ'|'cube'|'donut'
                              // For an animation node, call with no argument.
                              // Eg: hip = pelvis.add(); leg = hip.add('tubeY');
child = obj.child(index)         // return the nth child of this node
obj = obj.remove(child or index)  // remove a child by reference or by index
obj = obj.identity()             // remove all matrix transformations
obj = obj.move([x,y,z] or x,y,z)  // translate by [x,y,z]
obj = obj.turnX(radians)          // rotate about X axis
obj = obj.turnY(radians)          // rotate about Y axis
obj = obj.turnZ(radians)          // rotate about Z axis
obj = obj.aimX(vec)               // rotate so that X axis faces specified vec
obj = obj.aimY(vec)               // rotate so that Y axis faces specified vec
obj = obj.aimZ(vec)               // rotate so that Z axis faces specified vec
obj = obj.texture(source_file)    // apply a texture image
obj = obj.scale(s)                // scale by [s,s,s] (uniform scaling)
obj = obj.scale([x,y,z] or x,y,z) // scale by [x,y,z] (non-uniform scaling)
obj = obj.color([r,g,b] or r,g,b) // set color: values are between 0.0 and 1.0
obj = obj.info(info)              // add extra shape info like donut thickness
```

VALUES:

```
model.time                        // time in seconds since creation of model
model.deltaTime                   // time since previous animation frame
```

When any property is left unspecified for an object, that property is inherited from the object's parent. If the parent does not have that property, the search for a value continues all the way up to the root. All properties are defined by default for the root.