http://animateyourhtml5.appspot.com

Goal 1: move the tile with a click

```
#tile.begin { left: 20px; top: 20px; }
#tile.end { left: 220px; top; 220px; }
```

Goal 2: add a transition

-webkit-transition: 1000ms ease-in;

-webkit-transition-property: all;
-webkit-transition-duration: 1000ms;
-webkit-transition-timing-function: ease-in;
essayer: ease-in, ease-out, ease-in-out, linear, default

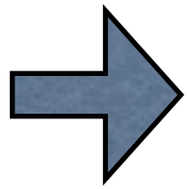
Goal 3: use a *cubic-bezier* speed

-webkit-transition-timing-function: cubic-bezier(0, 0.5, 1, 0.5);

Goal 4: animate other CSS properties

exemples:

```
opacity: 1;
background-color: red;
border: none;
```



```
opacity: 0;
background-color: blue;
border: 10px solid black;
```

Goal 5: use CSS transforms

exemples:

```
-webkit-transform: rotate(360Deg);
-webkit-transform-origin: 0 0;
-webkit-transform: scale (0.5);
-webkit-transform: rotateX(360Deg);
-webkit-transform: skew (10Deg, 0Deg);
```

Goal 6: use animations with multiple keyframes

```
 -webkit-animation: letsrock 1000ms

@-webkit-keyframes letsrock
{
   from  {left: 10px; top: 10px;}
   50%   {left: 210px; top: 10px;}
   to    {left: 210px; top: 10px; -webkit-transform: rotate (90Deg);}
}
```

Goal 7: go crazy!

Goal 1: rotate the "cube" object.

```
-webkit-animation: rocknroll 15s linear infinite

@-webkit-keyframes rocknroll
{
   to { -webkit-transform: rotateZ(1080Deg) rotateY(360Deg); }
}

-webkit-animation-direction: normal / alternate;
-webkit-animation-duration: 1000ms;
-webkit-animation-timing-function: ease-in / linear / ...
-webkit-animation-delay: 2s;
-webkit-animation-iteration-count: 1 / 2 / ... / infinite;
```

Goal 2: add perspective.

```
-webkit-perspective: 500px;
-webkit-transform-style: preserve-3d;
```

Goal 3: put one face in place, then add more.

```
#tile_A
{
   -webkit-transform: translateZ(150px);
}

#tile_B
{
   -webkit-transform: translateX(150px) rotateY(90Deg);
}
```

Goal 4: open the cube on a mouse hover.

```
#tile_A {-webkit-transform: translateZ(150px); }

:hover > #tile_A {-webkit-transform: translateZ(250px); }

-webkit-transition: 500ms ease-out;
```

Goal 5: go crazy!

Goal 1: move the tile with a click

```
var nod = document.getElementById(id);

nod.style.left = 220 + “px”;
nod.style.top = 220 + “px”;
```

Goal 2: move the tile in an animation loop

```
function on_move(time)
{

   // Move the object here - Use a global variable to store
   // its position for the time being.

   webkitRequestAnimationFrame(on_move);
}
```

Goal 3: animate multiple objects at the same time, with correct parameters

```
var animations = new Object;
...
animations[id] = new Object;
animations[id].nod = nod;
animations[id].starttime = new Date().getTime();
animations[id].startX = nod.offsetLeft;
...
for (var id in animations) { ... }
...
delete animations[id];
```

Goal 4: get the calculations right

```
// animation parameter, between 0 and 1
var t = (time - animations[id].start) / animations[id].duration

// current position
new_x = animations[id].startx + t * animations[id].relx
```

Goal 5: go crazy!

Goal 1: create a canvas and draw a square onto it

```
cnv = document.getElementByID('canvas')
                                        .getContext('2d');
cnv.lineWidth = 3;
cnv.strokeStyle = #RRGGBB;
cnv.beginPath();
cnv.moveTo(x, y);
cnv.LineTo(x, y);
cnv.stroke();
```

Goal 2: draw a tile with a custom rotation angle

```
var tile = new Image();
tile.src = “img/tile_world.png”;
cnv.save();
cnv.translate(x, y);
cnv.rotate(0.1);  // radians!
cnv.drawImage(tile, xi, yi);
cnv.restore();
```

Goal 3: initialize the world and draw it

```
function setupWorld()
{
   createWorldWithGravity(); // Box2d helper
   tile = createBox(...); // Box2d helper
}
function drawWorldIn(world, canvas)
{
   canvas.clearRect(...);
   drawWorldWireframe(world, canvas); // Box2D helper
}
```

Goal 4: prepare 2 animation loops and launch them!

```
var world, canvas;  // + initialization of the world and the canvas
...
function runWorld ()
{
   world.Step (1/50, 1);
   setTimeout (runWorld, 1000/50);
}
function runAnimation ()
{
   draw (world, canvas);
   webkitRequestAnimationFrame (runAnimation);
}
```

Goal 5: add a ground and other objects

ground:

```
var fixed = true;
createBox(world, x_center, y_center, w, h, fixed);
...
```

box:

```
tile = createBox(world, x, y, w, h);
// to rotate the box in box2d (a in radians)
tile.setCenterPosition(tile.getCenterPosition(), a);
// to render as an image
tile.image = “img/tile_world.png”;
...
```

ball:

```
createBall(world, x, y, r /*, fixed*/);
```

Goal 6: go crazy!

html5rocks.com