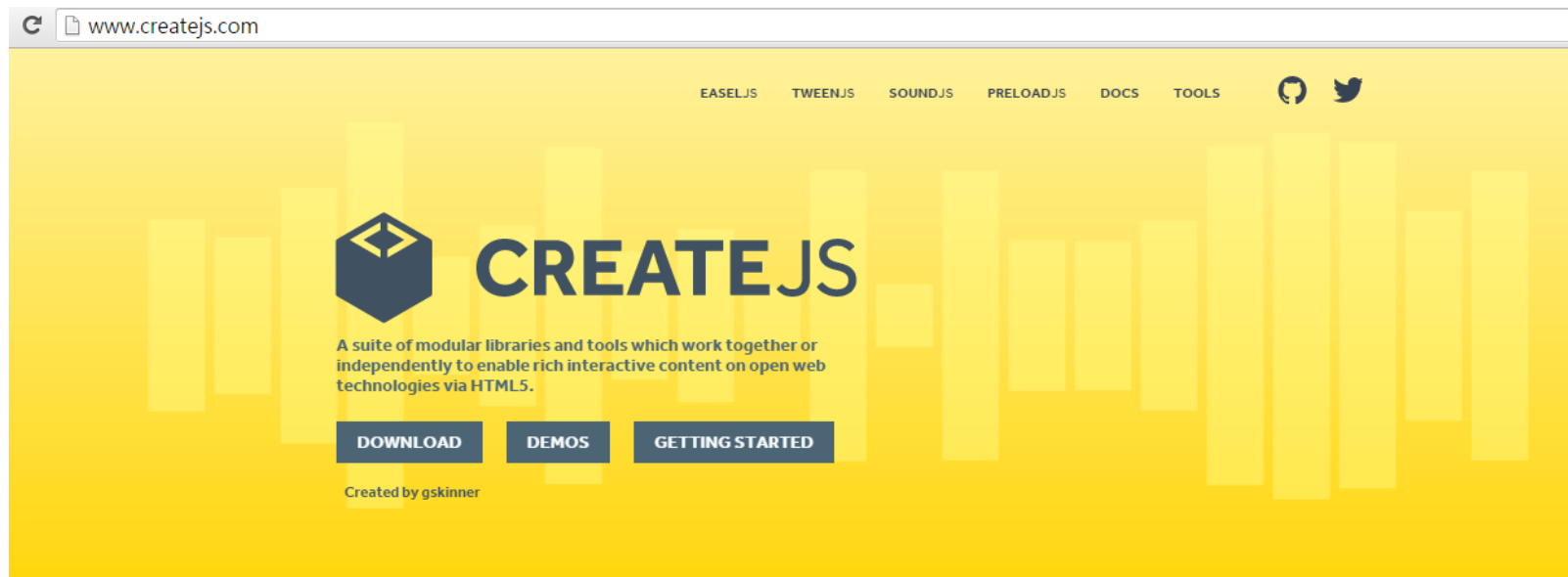


CREATEJS INTRODUCTION

By Dannis Mok

CreateJS Official Site



EASELJS

A JavaScript library that makes working with the HTML5 Canvas element easy.

[LEARN MORE](#)



TWEENJS

A JavaScript library for tweening and animating HTML5 and JavaScript properties.

[LEARN MORE](#)



SOUNDJS

A JavaScript library that lets you easily and efficiently work with HTML5 audio.

[LEARN MORE](#)



PRELOADJS

A JavaScript library that lets you manage and co-ordinate the loading of assets.

[LEARN MORE](#)

CreateJS Basics

- CreateJS is a canvas based JavaScript game framework which is open source and sponsored by Adobe.
- It is comprised of 4 different libraries
 - EaselJS– used for canvas drawing and sprite management
 - TweenJS – used for tween animation like Flash
 - SoundJS – used for sound management
 - PreloadJS – used for preloading assets for images or sounds



EASELJS



TWEENJS



SOUNDJS



PRELOADJS

CreateJS Setup

- One can load all the libraries or just load the required by add the appropriate scripts in the header section.
- For quick access, can use the CDN methods

**CREATEJS**
(ALL LIBRARIES)

2015.05.21	https://code.createjs.com/createjs-2015.05.21.min.js	2015/05/21
EaselJS 0.8.1, TweenJS 0.6.1, SoundJS 0.6.1, PreloadJS 0.6.1		

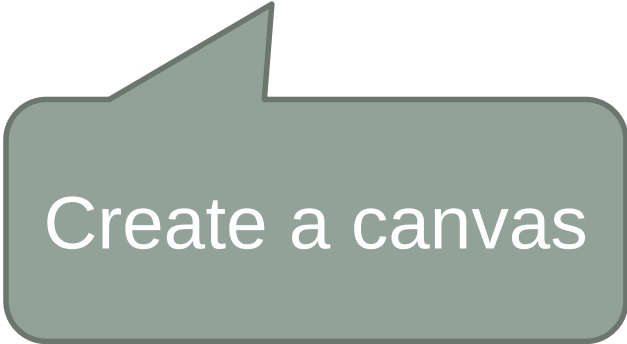
OLD VERSIONS

**EASELJS**

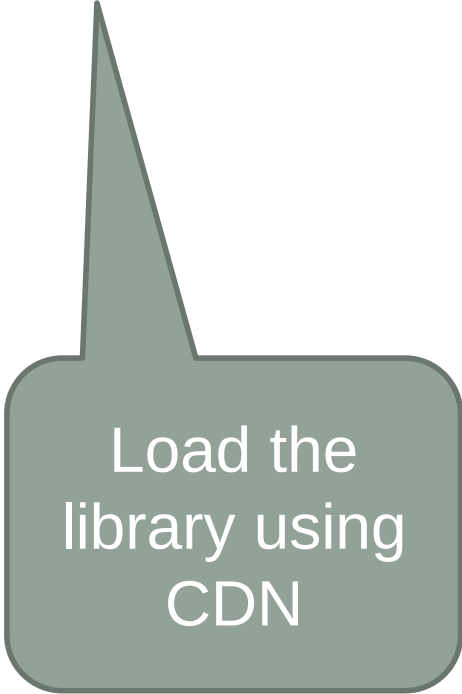
0.8.1	https://code.createjs.com/easeljs-0.8.1.min.js	2015/05/21
--------------	--	-------------------

CreateJS Setup

```
<head>  
  <script src="https://code.createjs.com/createjs-2015.05.21.min.js"></script>  
</head>  
<body>  
  <canvas id="c1" width="500" height="500"></canvas>  
</body>
```



Create a canvas



Load the
library using
CDN

Game Script Usage

```
var canvas = document.querySelector("#c1");  
var stage = new createjs.Stage(canvas);
```

Setup the game loop by using
the Ticker object

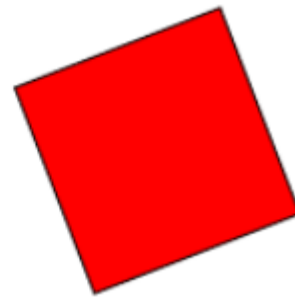
Setup the stage using the
canvas

```
createjs.Ticker.setFPS(60);  
createjs.Ticker.addEventListener("tick",handleTick);  
  
function handleTick(e) {  
    update();  
    stage.update();  
}
```

Game Script Usage

```
var g = new createjs.Graphics();  
g.beginStroke('#000').beginFill('#FF0000').drawRect(0, 0, 100, 100);  
var square = new createjs.Shape(g);  
square.x = 150;  
square.y = 100;  
stage.addChild(square);  
createjs.Tween.get(square).to({rotation:360},3000);
```

Create a red rectangle
and add to the stage.
Apply tween animation to
it by rotating 360 degree in
3 seconds



Game Script Usage

```
var circle = new createjs.Shape();
circle.graphics.beginFill('#0000FF').drawCircle(0,0,50);
circle.x = stage.canvas.width / 2;
circle.y = stage.canvas.height / 2;
stage.addChild(circle);

circle.addEventListener("mousedown",function(e) {

    stage.addEventListener("stagemousemove",function(e) {

        circle.x = stage.mouseX;
        circle.y = stage.mouseY;

    });

    stage.addEventListener("stagemouseup",function(e) {

        e.target.removeAllEventListeners();

    });

});
```



Create a blue circle
and let it to be
dragged and
dropped by mouse

Game Script Usage

Enable mouseover detection (which is disabled by default) and change the transparency of the square

```
stage.enableMouseOver();

square.addEventListener("mouseover", function (e) {
    square.alpha = .5;
});

square.addEventListener("mouseout", function (e) {
    square.alpha = 1;
});
```

Game Script Usage

```
var direction = 1;
var speed = 5;

function update() {

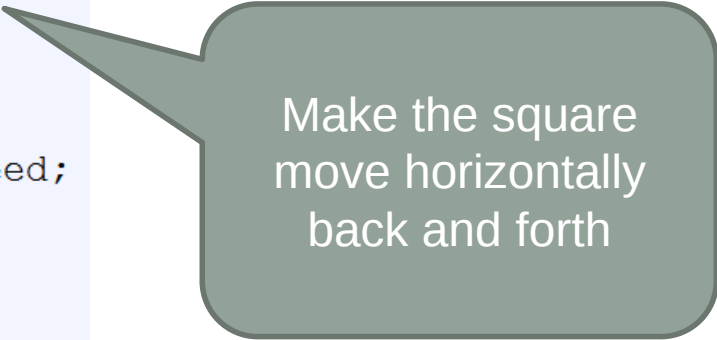
    square.x = square.x + direction * speed;
    if(square.x > canvas.width - 50) {

        square.x = canvas.width - 50;
        direction = -1;
    }

    if(square.x < 0 + 50) {

        square.x = 0 + 50;
        direction = 1;
    }

}
```



Make the square
move horizontally
back and forth

Game Script Usage

```
img1 = new createjs.Bitmap("butterfly.png");  
img1.x = 90;  
img1.y = 40;  
  
img2 = new createjs.Bitmap("butterfly.png");  
img2.x = 200;  
img2.y = 40;  
  
stage.addChild(img1);  
stage.addChild(img2);
```

Add a Bitmap object onto the stage

Moving Butterfly

```
var butterfly;  
var image1 = new Image();  
image1.src = "butterfly.png";  
image1.addEventListener("load", function() {  
  
    butterfly = new createjs.Bitmap(image1);  
    butterfly.addEventListener("click", function() {  
  
        alert("It is clicked !!");  
        butterfly.visible = false;  
    });  
    butterfly.vx = 0;  
    butterfly.vy = 0;  
    stage.addChild(butterfly);  
  
});
```

Load the image
and then create
the object.

When the butterfly
is clicked, it will be
disappeared

Set the initial
velocity to 0.



Moving Butterfly



```
function update() {
```

```
    butterfly.vx = butterfly.vx + 4;
```

```
    if(butterfly.vx > 100) {
```

```
        butterfly.vx = 100;
```

```
    }
```

```
    butterfly.x = butterfly.x + butterfly.vx;
```

```
    if(butterfly.x > canvas.width - butterfly.getBounds().width) {
```

```
        butterfly.x = canvas.width - butterfly.getBounds().width;
```

```
        butterfly.vx = butterfly.vx * -1;
```

```
    }
```

```
    butterfly.y++;
```

```
    butterfly.alpha = butterfly.alpha - 0.01;
```

```
    butterfly.rotation = butterfly.rotation - 0.1;
```

```
}
```

Increase the velocity of the sprite and set the max to 100

Bounce back and forth and update alpha and rotation value gradually

Spritesheet handling

```
var data = {  
  "images":["catAndMonster.png"],  
  "frames": [  
    [0,0,64,64],  
    [64,0,64,64],  
    [128,0,64,64]  
  ],  
  "animations": {  
    "f1": [0],  
    "f2": [1],  
    "f3": [2],  
    "f4": [1,2]  
  }  
};
```



Filename of the
spritesheet

Label for single
sprite and also a
sequence of sprites

The bounding
rectangle of the 3
sprites

Spritesheet handling

```
var spritesheet = new createjs.SpriteSheet(data);  
var cat = new createjs.Sprite(spritesheet, "f1");  
cat.x = 100;  
cat.y = 100;  
cat.vx = 0;  
cat.vy = 0;  
stage.addChild(cat);  
  
var monster = new createjs.Sprite(spritesheet, "f4");  
monster.x = 100;  
monster.y = 100;  
monster.vx = 0;  
monster.vy = 0;  
stage.addChild(monster);
```

Add the spritesheet
and the sprites to the
stage



Game Loop and Update Function

```
var canvas = document.querySelector("#c1");  
var stage = new createjs.Stage(canvas);  
createjs.Ticker.setFPS(5);  
createjs.Ticker.addEventListener("tick", tick);
```

```
function tick() {  
    update();  
    stage.update();  
}
```

Game Loop

```
function update() {  
    cat.vx = cat.vx + 4;  
    if(cat.vx > 100) {  
        cat.vx = 100;  
    }  
  
    cat.x = cat.x + cat.vx;  
    if(cat.x > canvas.width - cat.getBounds().width) {  
        cat.x = canvas.width - cat.getBounds().width;  
        cat.vx = cat.vx * -1;  
    }  
    cat.y++;  
    cat.alpha = cat.alpha - 0.01;  
    cat.rotation = cat.rotation - 0.1;  
}
```

Update function

Sound Handling

```
var sounds = [  
  {id:"Background", src:"dreamRaid1.mp3"},  
  {id:"Explosion", src:"explosion.mp3"}  
];  
createjs.Sound.alternateExtensions = ["ogg"];  
createjs.Sound.addEventListener("fileload", handleLoad);  
createjs.Sound.registerSounds(sounds);
```

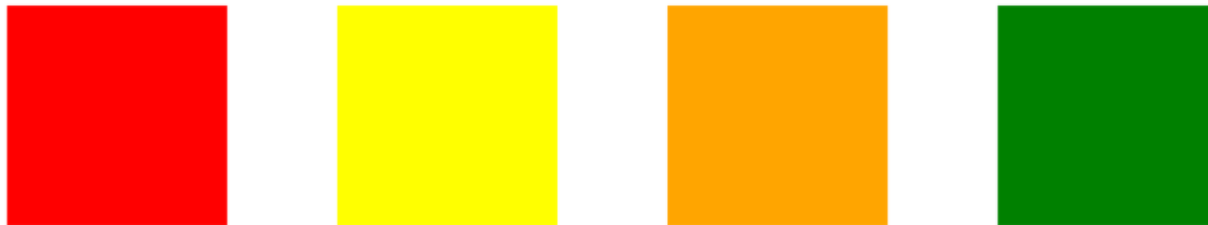
```
function handleLoad(event) {  
  createjs.Sound.setVolume(0.5);  
  createjs.Sound.play(event.id);  
}
```

Sound files

Register the sounds and then add event handler when the sound is loaded

Set the volume of the sound and play it

Drag and Drop Game



Drag the color
boxes onto the
matching grid
above

Game Code Listing

```
<head>
  <script src="https://code.createjs.com/createjs-2015.05.21.min.js"></script>
</head>
<body>
  <canvas id="c1" width="1000" height="500"></canvas>
</body>
<script>
```

```
var stage;
var slots = [];
var shapes = [];
var score = 0;
```

Define the array to
store the slots and
shapes sprites

```
var canvas = document.querySelector("#c1");
var stage = new createjs.Stage(canvas);

createjs.Ticker.setFPS(60);
createjs.Ticker.addEventListener("tick", function() {
  stage.update();
});
```

Game Code Listing

```
colors = ["red","orange","yellow","green"];

for(var i=0; i<colors.length; i++) {

    var slot = new createjs.Shape();
    slot.graphics.beginFill("white");
    slot.graphics.beginStroke(colors[i]);
    slot.graphics.drawRect(0,0,100,100);
    slot.x = (i * 150) + 100;
    slot.y = 100;
    slot.regX = 50;
    slot.regY = 50;

    slots.push(slot);
    stage.addChild(slot);
}
```

Create the color slots and push into the array

Game Code Listing

```
for(var i=0; i<colors.length; i++) {  
  
    var shape = new createjs.Shape();  
    shape.graphics.beginFill(colors[i]);  
    shape.graphics.drawRect(0,0,100,100);  
    shape.key = i;  
    shapes.push(shape);  
  
}  
  
// Generate a group of unique random numbers  
var positions = [];  
while(positions.length < 4) {  
  
    var r = Math.floor(Math.random()*colors.length);  
  
    if(positions.indexOf(r) == -1) {  
        positions.push(r);  
    }  
  
}
```

Create the color shapes and put the index in the key property for matching

Create a set of random number and put into the positions array

Game Code Listing

```
for(var i=0; i<shapes.length; i++) {  
  
    shapes[i].x = positions[i] * 150 + 100;  
    shapes[i].y = 250;  
    shapes[i].addEventListener("mousedown", drag);  
    shapes[i].regX = 50;  
    shapes[i].regY = 50;  
    shapes[i].homeX = shapes[i].x;  
    shapes[i].homeY = shapes[i].y;  
    stage.addChild(shapes[i]);  
  
}
```

Assign the position
of each shape
based on the
random number

Register
mousedown event
handler

Record the x and y
position in the homeX
and homeY for going
back

Game Code Listing

```
function drag(e) {  
  
    var shape = e.target;  
    var slot = slots[shape.key];  
    stage.addEventListener("stagemousemove", function(e) {  
        shape.x = e.stageX;  
        shape.y = e.stageY;  
    });  
    stage.addEventListener("stagemouseup", function(e) {  
  
        stage.removeAllEventListeners();  
        var point = slot.globalToLocal(stage.mouseX, stage.mouseY);  
        if(slot.hitTest(point.x, point.y)) {  
  
            createjs.Tween.get(shape).to({x:slot.x, y:slot.y}, 200, createjs.Ease.quadOut);  
            score++;  
            if(score == 4) {  
                alert("You win");  
            }  
        } else {  
            createjs.Tween.get(shape).to({x:shape.homeX, y:shape.homeY}, 200);  
        }  
    });  
}
```

Move the
shape

Hit test code