

## Runstant Lite を用いて HTML, JavaScript プログラミングを学ぶ

Web アプリの作成により、次のことを学んでもらいます。

1. 無料ソフトウェアの活用
2. 無料のソフトウェア開発ツール Runstant Lite
3. Web ブラウザのゲームライブラリ phina.js
4. Web アプリとは

### ◆無料ソフトウェアの活用

**無料ソフトウェア**とは、インターネットなどで、無料で提供されているソフトウェアのこと。「フリーソフト」とも言う。パソコンの学習や活用に大いに役立つ。

※ 無料ソフトウェアも、製作者に**著作権**がある。作者が自分であるかのように嘘をついたり、無料ソフトウェアを自分の名前で販売したりなどの違反をしてはいけない。

※ 無料ソフトウェアを使うとき、もし、うまく動かなくなったとする。そのとき、**製作者に苦情を言うのはマナー違反**です。製作者は「いろんな人に使ってほしい」と善意を持っていますので尊重すること

### ◆Web ブラウザで動く無料ソフトウェア

自分のパソコンにインストールしなくても、インターネットと **Web ブラウザ**さえあればすぐに使える無料ソフトウェアも多数ある。たとえば gmail など。

この資料では、2つの無料ソフトウェアを体験する。

- ・ **pinta.js**: これは、簡単に **Web ブラウザで動くゲーム**（2次元のグラフィックス）が制作できるソフトウェア。

**phina.js** の情報は、**次の Web ページで公開**。

<http://phiary.me/phinajs-release/>

- ・ **Runstant Lite**: ソフトウェアの制作のために使う。

**Runstant Lite** の情報は、**次の Web ページで公開**。

<http://lite.runstant.com/about>

## ここまでのまとめ

- ・ 無料ソフトウェアは便利である。製作者に苦情を言うのはマナー違反である。
- ・ この資料では無料ソフトウェア **phina.js**, **Runstant Lite** を使う。

## ◆ 無料のソフトウェア開発ツール **Runstant Lite**

コンピュータはソフトウェアで動く。

ソフトウェアを、コンピュータを使って作る時、**ソフトウェアを制作するためのソフトウェア**（ソフトウェア開発ツール）を使う。ソフトウェア開発ツールにはいろいろな種類がある（Visual Studio など）。ここでは、Runstant Lite を使ってみる。

### 演習 1. Runstant Lite について、次を行いなさい。

① Web ブラウザで、次を開く。

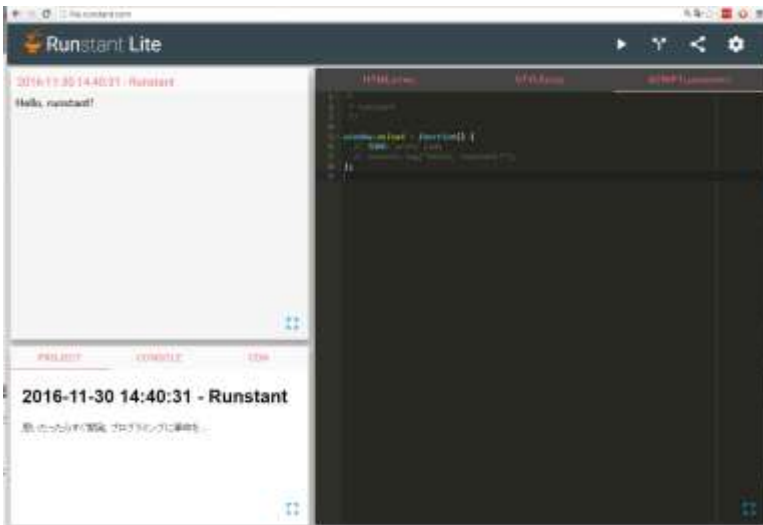
Edge、Chrome、Firefox を使ってください。IE は避けてください。

**<http://lite.runstant.com/about>**

② 始めたいので「**GETTING STARTED**」をクリック。



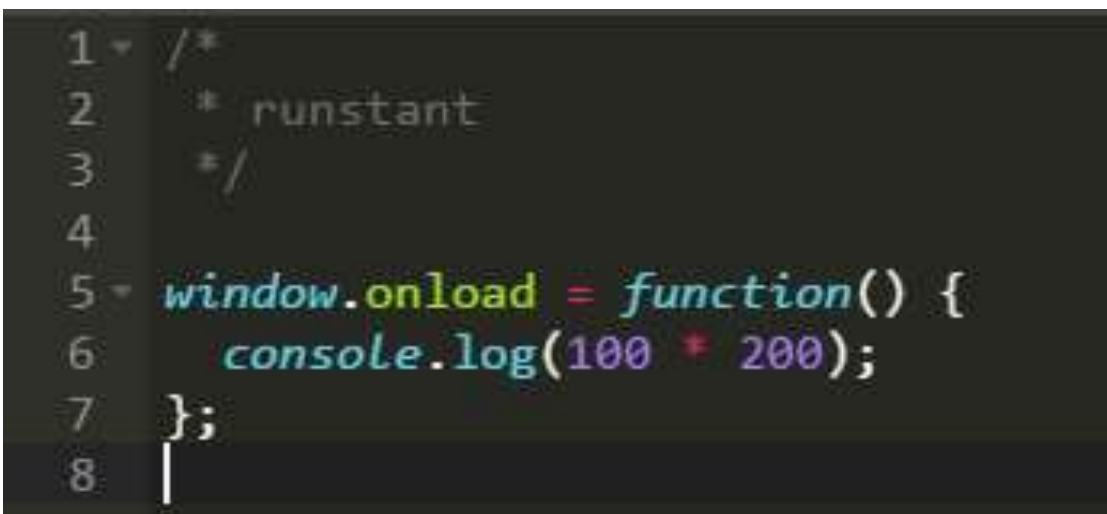
③ 次のように表示されるので確認する。



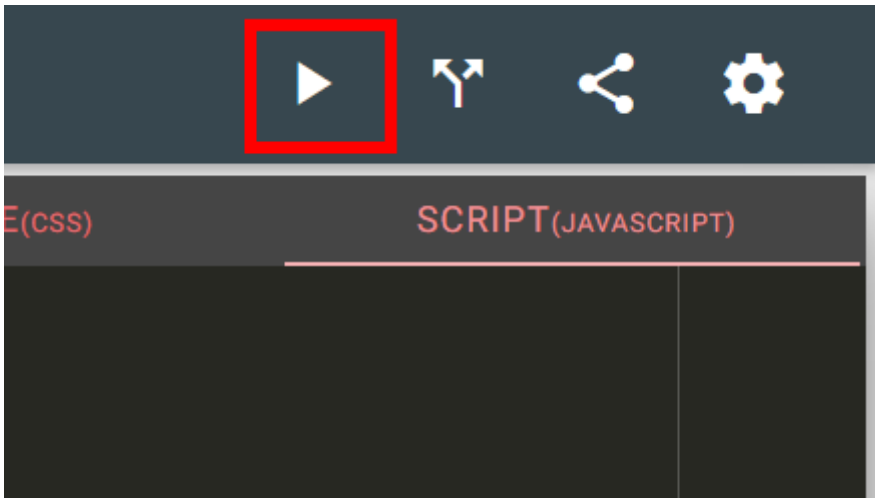
④ 右側の黒い画面で、ソフトウェアを作る。

右側の黒い画面で、次のように書き換える。

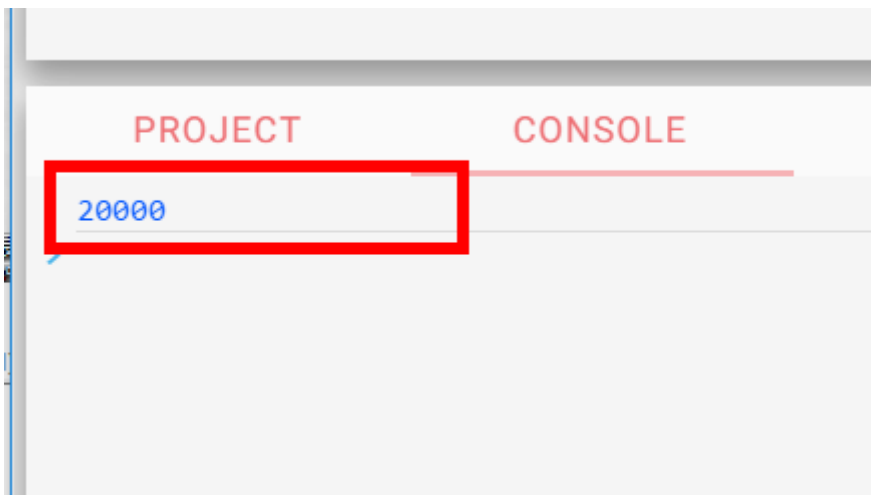
```
/*  
 * runstant  
 */  
  
window.onload = function() {  
    console.log(100 * 200);  
};
```



⑤ 再生ボタンをクリックする.



⑥ 100 かける 200 の計算結果として, **20000** が表示されるので確認する



うまく表示されない場合には, ④に戻って確認する.

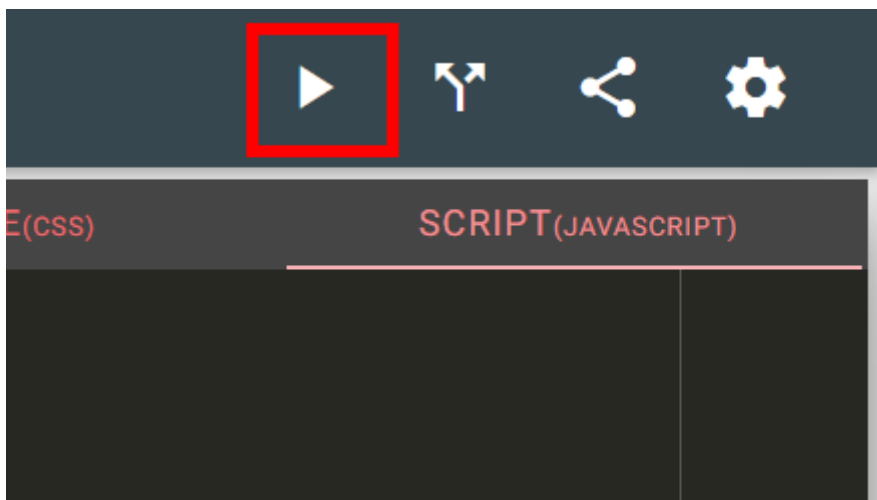
⑦ 右側の黒い画面で, 「100 \* 200」のところを書き換える.

右側の黒い画面で, 次のように書き換える.

```
/*  
 * runstant  
 */  
  
window.onload = function() {  
  console.log(1 / 2 + 3 / 4);  
};
```

```
1 - /*
2   * runstant
3   */
4
5 - window.onload = function() {
6   console.log(1 / 2 + 3 / 4);
7 };
8
```

⑧ 再生ボタンをクリックする.



⑨ 「 $1/2 + 3/4$ 」の計算結果として、**1.25**が表示されるので確認する



うまく表示されない場合には、⑦に戻って確認する.

## ここまでのまとめ

- ・ソフトウェア開発ツールを使って、**ソフトウェアを作り、実行**できる
- ・ソフトウェアの中には、**計算させたい式**などを書き下すことができる。

## ◆Web ブラウザのゲームライブラリ **phina.js**

ゲームのソフトウェアを作るとき、

画像、ボタン、ゲージ、サウンド、キーボード

などを扱うことになる。これらを簡単に扱えるようにするためのソフトウェアをゲームライブラリと言ったりする。以前の授業でもゲームライブラリを使った (Panda3d など)。この資料では、Web ブラウザで動くゲームに便利なゲームライブラリ **phina.js** を使う。

### 演習. phina.js について、次を行いなさい。

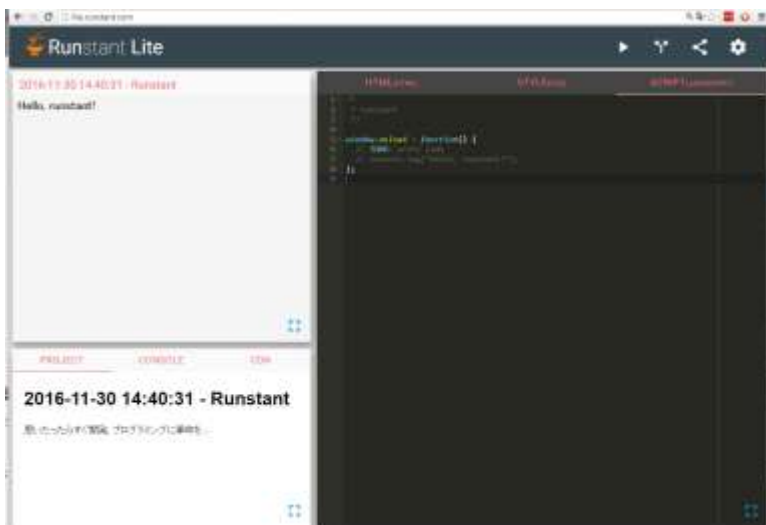
- ① Web ブラウザで、次を開く。(さっきの演習と同じ)。

**<http://lite.runstant.com/about>**

- ② 始めたいので「**GETTING STARTED**」をクリック。(演習 1 と同じ)。



③ 次のように表示されるので**確認**する。(さきほどの演習と同じ)。



④ 「HTML(HTML)」をクリック



⑤ 次のように表示されるので**確認**する。



⑥ 右側の黒い画面の「<style>\${style}</style>」の直後に、次の1行を追加。

他はそのまま。 ※コピー&ペースト

これは、 **phina.js** の機能を取り込むためのもの。

```
<!doctype html>

<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, user-scalable=no" />
    <meta name="apple-mobile-web-app-capable" content="yes" />

    <title>${title}</title>
    <meta name="description" content="${description}" />

    <style>${style}</style>
    <script src='http://cdn.rawgit.com/phi-jp/phina.js/v0.2.0/build/phina.js'></script>
    <script>${script}</script>
  </head>
  <body>
    <h1>Hello, runstant!</h1>
  </body>
</html>
```



```
HTML(HTML)                                     STYLE(css)
1  <!doctype html>
2
3  <html>
4  <head>
5    <meta charset="utf-8" />
6    <meta name="viewport" content="width=device-width, user-scalable=no" />
7    <meta name="apple-mobile-web-app-capable" content="yes" />
8
9    <title>${title}</title>
10   <meta name="description" content="${description}" />
11
12   <style>${style}</style>
13   <script src='http://cdn.rawgit.com/phi-jp/phina.js/v0.2.0/build/phina.js'></script>
14   <script>${script}</script>
15  </head>
16  <body>
17    <h1>Hello, runstant!</h1>
18  </body>
19 </html>
20
```



⑦ 「SCRIPT(javascript)」をクリック



⑧ 右側の黒い画面で、次のように書く。

※長いので、コピー&ペーストしてください。

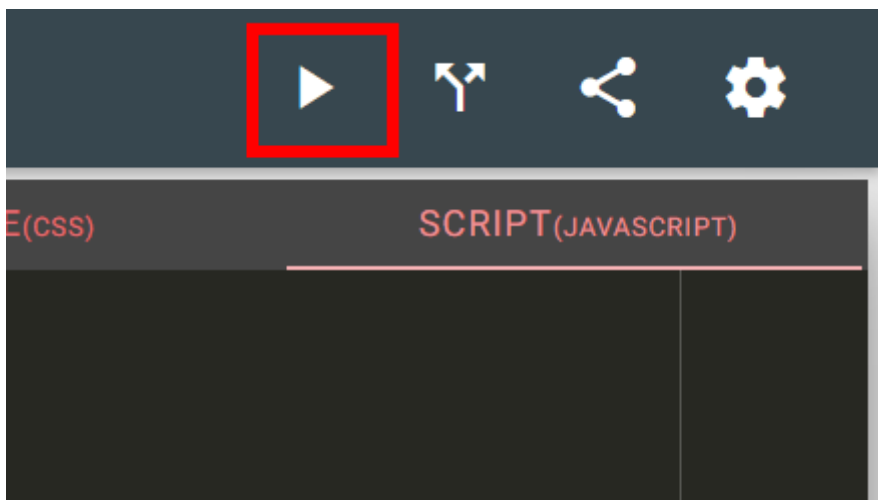
```
phina.globalize();

phina.define('MainScene', {
  superClass: 'CanvasScene',
  init: function() {
    this.superInit();
    var obj = RectangleShape().addChildTo(this);
    obj.setPosition(100, 200);
    this.obj = obj;
  },
});

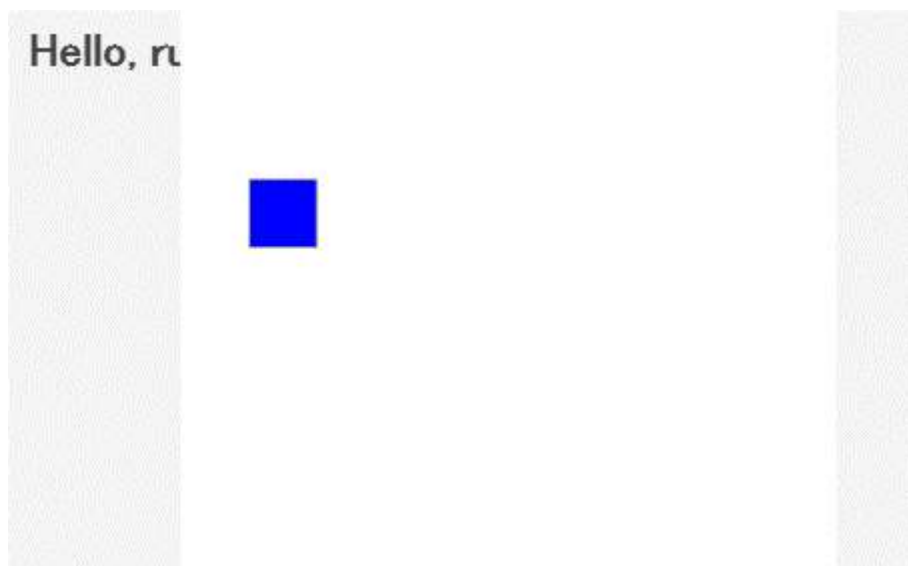
phina.main(function() {
  var app = GameApp({
    startLabel: 'main',
  });
  app.run();
});
```



⑨ 再生ボタンをクリックする.



⑩ 画面に青い四角が表示される.



うまく表示されない場合には, ⑧ に戻って確認する.

**後で使うので, Webブラウザは閉じずに残しておくこと.**

ここまでのまとめ

- ・ゲームエンジンには, 青い四角を描くなど, ゲームを作るのに必要な機能がそろっている.  
(画像, ボタン, ゲージ, サウンド, キーボードなどの機能がある).

## ◆ Webアプリとは

いま体験してもらった「青の四角を描く」ための Web アプリは、次のような仕組みになっている。



Web ブラウザ  
(画面表示と情報の受付)

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, user-scalable=no" />
    <meta name="apple-mobile-web-app-capable" content="yes" />
  </head>
  <body>
    <div style="text-align:center">
      <div style="display:inline-block; width:100px; height:100px; background-color:blue; margin:auto;></div>
      <br>
      <div style="display:inline-block; width:100px; height:100px; background-color:yellow; margin:auto;></div>
    </div>
    <div style="margin-top:10px;>
      <div style="display:inline-block; width:100px; height:100px; background-color:blue; margin:auto;></div>
    </div>
  </body>
</html>
```

HTML  
(画面に何を表示するかの指示など)

```
phiho.gibblio();
phiho.define('MainView', {
  superClass: 'GizmoView',
  init: function() {
    this.superClass();
    var obj = this.viewPage().addChild({id:1,
      obj.setPosition(100, 100)});
    this.obj = obj;
  },
});
phiho.main(function() {
  var app = newApp({
    startLabel: 'main',
  });
  www.run();
});
```

SCRIPT  
(計算など)

Web アプリは、上の図で書いたように、(1) Web ブラウザ、(2) HTML、(3) SCRIPT (スクリプト) が共同して動くという仕組みになっている。

先ほどのプログラムは、いろいろな **Web アプリを作ることができるための土台**がすでに完成しているのです。

今度の演習では、SCRIPT (スクリプト) の部分を書き換えて、**ボールの動きのシミュレーション**を行うようにします。

### ① 「SCRIPT(javascript)」をクリック



② 右側の黒い画面で、今度は、次のように書く。

**※長いので、コピー&ペーストしてください。**

```
phina.globalize();

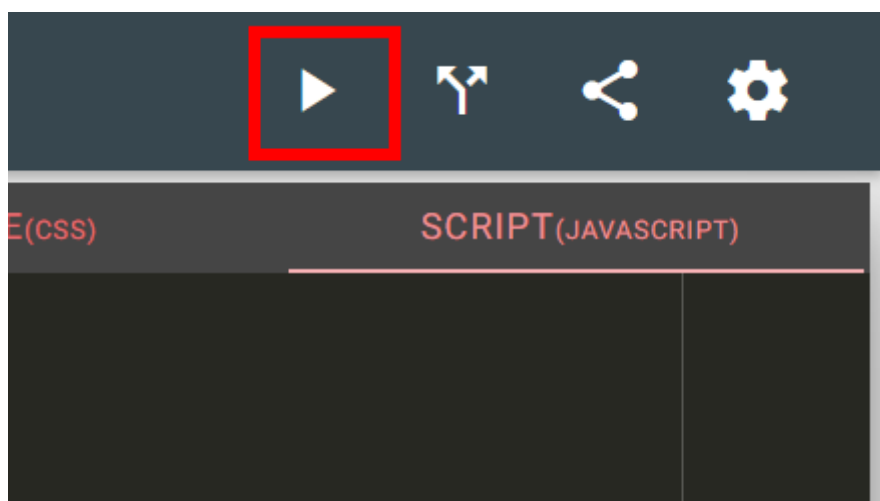
phina.define('MainScene', {
  superClass: 'CanvasScene',
  init: function() {
    this.superInit();
    var x = CircleShape().addChildTo(this);
    x.setPosition(80, this.gridY.center());
    x.physical.force(8, 0);
    x.physical.gravity.set(0, 9.8 / 10);

    var floor = RectangleShape({width: this.gridX.width}).addChildTo(this);
    floor.setPosition(this.gridX.center(), this.gridY.width);
    var w1 = RectangleShape({height: this.gridY.width}).addChildTo(this);
    w1.setPosition(0, this.gridY.center());
    var w2 = RectangleShape({height: this.gridY.width}).addChildTo(this);
    w2.setPosition(this.gridX.width, this.gridY.center());
    this.ball = x;
    this.floor = floor;
    this.w1 = w1;
    this.w2 = w2;
  },
  // 毎フレーム処理
  update: function() {
    var vx = this.ball.physical.velocity.x;
    var vy = this.ball.physical.velocity.y;
    if (this.ball.hitTestElement(this.floor)) {
      this.ball.physical.force(vx, -vy);
    }
    if (this.ball.hitTestElement(this.w1)) {
      this.ball.physical.force(-vx, vy);
    }
    if (this.ball.hitTestElement(this.w2)) {
      this.ball.physical.force(-vx, vy);
    }
  },
});

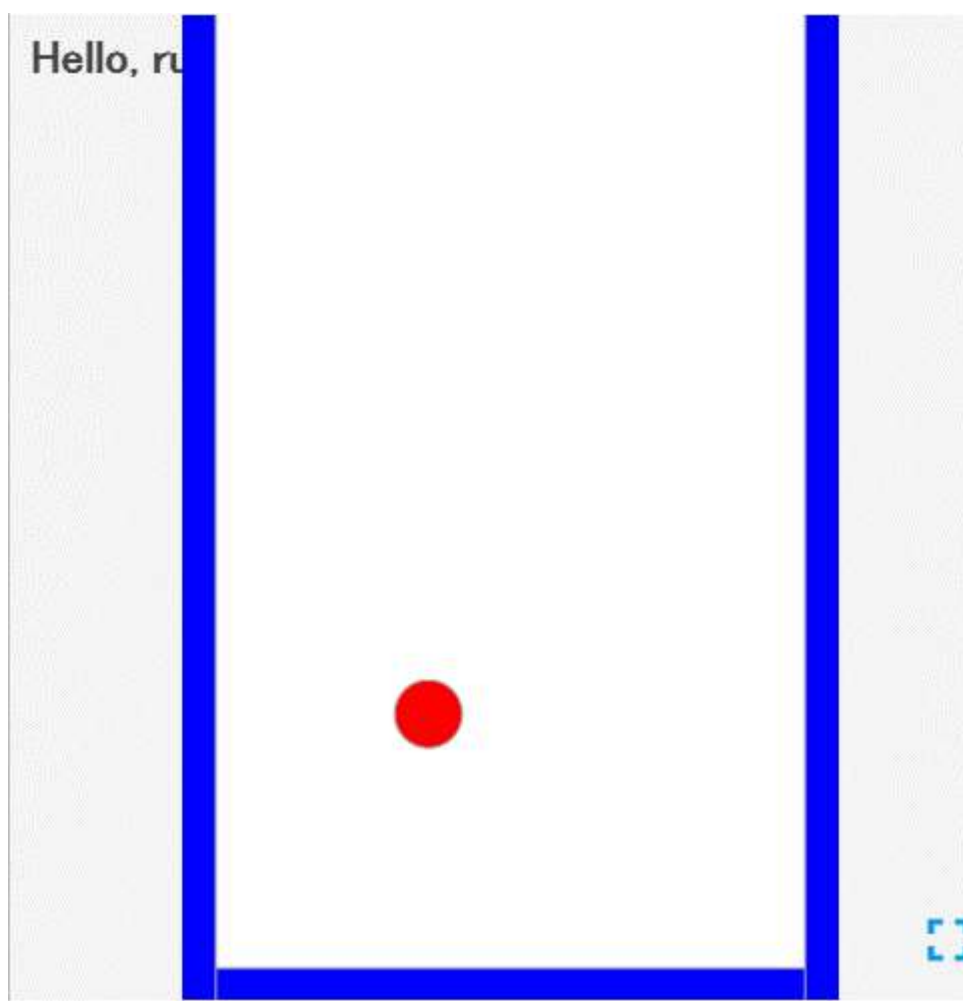
phina.main(function() {
  var app = GameApp({
    startLabel: 'main',
  });
  app.run();
});
```

```
1 phina.globalize();
2
3 phina.define('MainScene', {
4   superClass: 'CanvasScene',
5   init: function() {
6     this.superInit();
7     var x = CircleShape().addChildTo(this);
8     x.setPosition(80, this.gridY.center());
9     x.physical.force(8, 0);
10    x.physical.gravity.set(0, 9.8 / 10);
11
12    var floor = RectangleShape({width: this.gridX.width}).addChildTo(this);
13    floor.setPosition(this.gridX.center(), this.gridY.width);
14    var w1 = RectangleShape({height: this.gridY.width}).addChildTo(this);
15    w1.setPosition(0, this.gridY.center());
16    var w2 = RectangleShape({height: this.gridY.width}).addChildTo(this);
17    w2.setPosition(this.gridX.width, this.gridY.center());
18    this.ball = x;
19    this.floor = floor;
20    this.w1 = w1;
21    this.w2 = w2;
22  },
23  // 毎フレーム処理
24  update: function() {
25    var vx = this.ball.physical.velocity.x;
26    var vy = this.ball.physical.velocity.y;
27    if (this.ball.hitTestElement(this.floor)) {
28      this.ball.physical.force(vx, -vy);
29    }
30    if (this.ball.hitTestElement(this.w1)) {
31      this.ball.physical.force(-vx, vy);
32    }
33    if (this.ball.hitTestElement(this.w2)) {
34      this.ball.physical.force(-vx, vy);
35    }
36  },
37 });
38
39 phina.main(function() {
40   var app = GameApp({
41     startLabel: 'main',
42   });
43   app.run();
44 });
```

③ 再生ボタンをクリックする.



④ ボールの動きのシミュレーションが, アニメーションで表示される.



うまく表示されない場合には, ③ に戻って確認する.

## まとめ

言語： JavaScript

枠組み： Web ブラウザ + HTML + スクリプト

行ったこと： 四角形の描画, ボールの動きのシミュレーション