

Canvas 授業

Canvas とは？？

HTML5 より導入された API。JavaScript を駆使して、Web 上に図形や絵の描画、画像映像の処理など、様々なことが出来る。いわば「グラフィック表現」が Web で可能になる。



- IE9 から対応、他のモダンブラウザは大抵対応（スマホも）
(<https://caniuse.com/#search=canvas>)
- ピクセルベースで描画表現を行う。
- Canvas 上でアニメーションさせる場合、は自力でスクリプトを駆使する必要がある。

先生呼び出しサービス

<http://www.ca-ll.in/room/gsdev09th>

Canvas の使用例



<http://festival.lattexplus.com/>



<http://experience.mausoleodiaugusto.it/en/intro>



<https://tkmh.me/>

Canvas はロジックの組み立てが大事!(復習)

CSS

```
①  
.area {  
  ②  
  background-color: ③ red;  
}
```

- ① class="area" がついている HTML タグの (どこの)
- ② background-color (背景色) を (何を)
- ③ red (赤色) にしたい (どうしたい)

JQ

```
$( ".btn" ).on( "click", function() { ①  
  イベント名  
  $( ".text" ). fadeOut( 3000 )  
  メソッドとセレクタは「.」で繋げる  
}); btn がクリックされたら function(){} の中身を順番に実行!!
```

- ① class="btn" がついている HTML タグがクリックされたら (いつ・どのタイミングで)
- ② class="text" がついている HTML タグを (何を)
- ③ 3 秒かけてだんだん消したい (どうしたい)

■ CSS

命令したい場所を決めて (どこの)
命令したい内容 (何をどうしたい) を書く!

■ JavaScript (jQuery)

命令をするタイミングを指定し (いつ・イベント)
命令をする場所を決めて (どこを・セレクタ)
実際に命令する内容を書く (どうしたい・メソッド)

5 歳の息子に初めてのお使いを頼むのと同じ!

例)

お昼になったら (イベント)
近所のスーパーに行って (セレクタ)
大根を買ってきて! (メソッド)

Canvas を扱ってみよう！

- 1、HTML 上に Canvas 要素を書く。
- 2、JavaScript で描画のための準備をする。
- 3、描画処理を書く。

1-1、Canvas 描画の準備をする

canvas 上に描画を色々したい場合、「**canvas タグ (場所)**」に「**描画 (命令)**」をする、となりますので、Canvas タグを何度も呼び出すことになります。

何度呼び出されてもいいように、変数に canvas タグを入れておきましょう！



```
<canvas id="xxx" width="1024" height="760"></canvas>
```

```
//id      →Canvas 要素に命名するユニーク ID
```

```
//width   →canvas 要素の幅
```

```
//height  →canvas 要素の高さ
```

※ CSS で幅と高さを指定すると、おかしいことになるので注意！！

(※ <http://jsdo.it/castero/vo2w>)

HTML

1-1、Canvas 描画の準備をする

- canvas タグを書いただけでは、描画は実行されない！
 - getContext メソッドを実行して初めて描画が出来る！
- (変数 ctx に canvas タグを自由に操作できる機能がたくさん入ったことになります)



//JavaScript で書くと

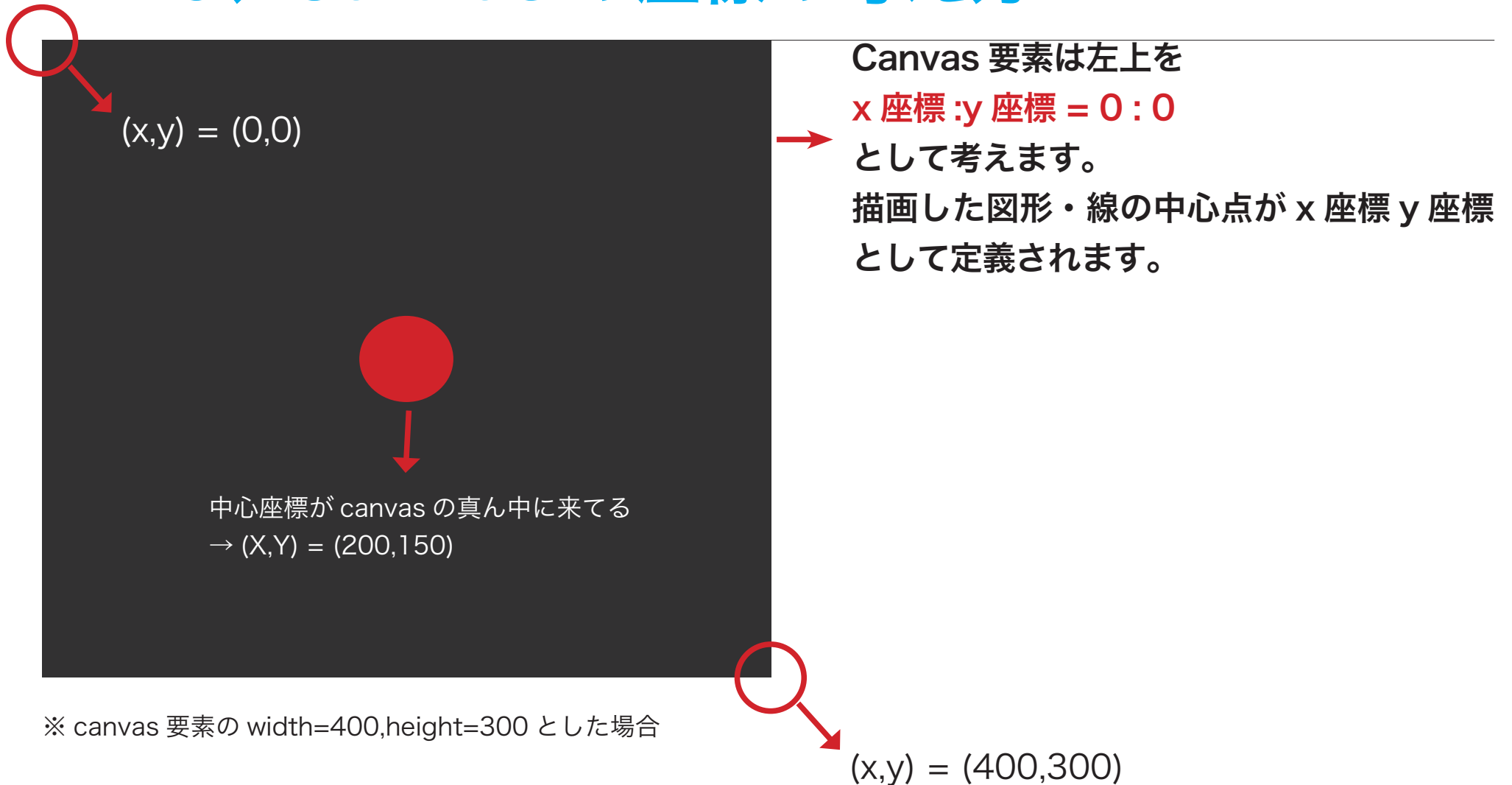
```
const can = document.getElementById("Canvas タグの ID");  
const ctx = can.getContext("2d");
```

//jQuery で書くと

```
const can = $("Canvas タグの ID")[0];  
const ctx = can.getContext("2d");
```

jQuery

2-0、Canvas の座標の考え方



2-1、Canvas で図形を描く時の手順



0、Canvas 要素をセレクト指定

1、**「どんな見た目の図形を描くのか」**を決めて
(・色は？線の太さは？ etc……..)

2、**「どこにどんな形の図形を描くのか」**を決めて
(・座標位置は？丸？四角？・ etc……..)

3、実際の描画を実行！

2-2、どんな見た目の図形を描くのか??

ctx.fillStyle 塗りつぶし色の指定
ctx.strokeStyle 線の色指定
ctx.lineWidth 線の太さの指定



■書き方例

```
ctx.fillStyle = "#f90";  
ctx.strokeStyle = "#090";  
ctx.lineWidth = 9;
```

<http://www.html5.jp/canvas/ref.html>

「2d コンテキストのプロパティ」を参照

3-1、どこにどんな形の図形を描くのか？（円）

■図形を描画 - 円 - （1,2 つめの引数は中心点の座標のこと）

`ctx.arc(X 座標, Y 座標, 半径, 描画開始角度, 描画終了角度, 円弧の向き);`

`ctx.stroke();` // 線の描画

`ctx.fill();` // 塗りつぶしの描画



■書き方例

```
ctx.arc(100,100,30,0,Math.PI*2,true);
```

```
//Math.PI*2 は 360°,Math.PI は 180°
```

```
// 一番最後は true だと半時計 false だと時計周り
```

3-2、どこにどんな形の図形を描くのか？（四角）

■図形を描画 - 長方形などの四角形 - （1,2 つめの引数は中心点の座標）

※共通して引数は以下の通り

（図形の左上点の X 座標 , 図形の左上点の Y 座標 , 図形の幅 , 図形の高さ）



■書き方例

```
ctx.fillRect(100, 100, 30, 30 ); // 塗りつぶし
```

```
ctx.strokeRect(100, 100, 30, 30 ); // 枠線のみ
```

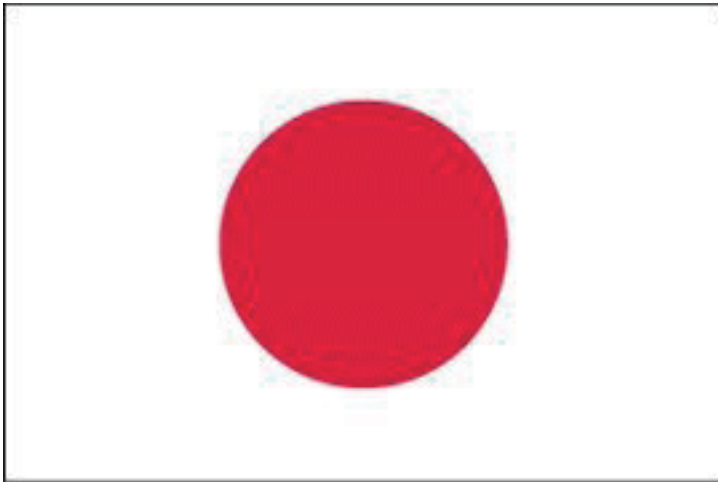
```
ctx.clearRect(100, 100, 50, 50 ); // 消す
```

※線付き四角形を作るときは `fillRect()` と `StrokeRect()` を両方使う。

ミニワーク

canvas 上に日本の国旗を書いてみる !!!

- ・ canvas エリアの色は白以外にしてみる。
- ・ サイズ感はさておき、日本の国旗っぽいのを canvas 上に作って、canvas の扱いに慣れる！



3-3、どこにどんな形の図形を描くのか？（線）

■線を描画

手順がすごく大事。



■書き方例（Illustrator できるとイメージしやすい）

```
ctx.beginPath();  
ctx.moveTo(10,10); // 開始点 X 座標と Y 座標  
ctx.lineTo(100,100); // 終了点 X 座標と Y 座標  
ctx.stroke(); // 線を描画  
ctx.closePath(); // 図形を閉じようとしています。
```

4-1、図形をアニメーションさせる

■ setInterval() や requestAnimationFrame() を使用



```
<script>
  var timer = setInterval(function(){
    ctx.fillStyle="#fff";
    ctx.clearRect(0,0,300,300);
    ctx.fillStyle="#f00";
    ctx.fillRect(30+count,30+count,30,30);
    count++;
    if(count>200){
      clearInterval(timer);
    }
  },100);
</script>
```

演習・描画アプリを作る

授業内課題の考え方

◎お絵かきアプリ作成で考え方を学ぶ（以下、最低要件）

1. 「canvas 要素内で mousedown
+ mousemove 中は線を描く」

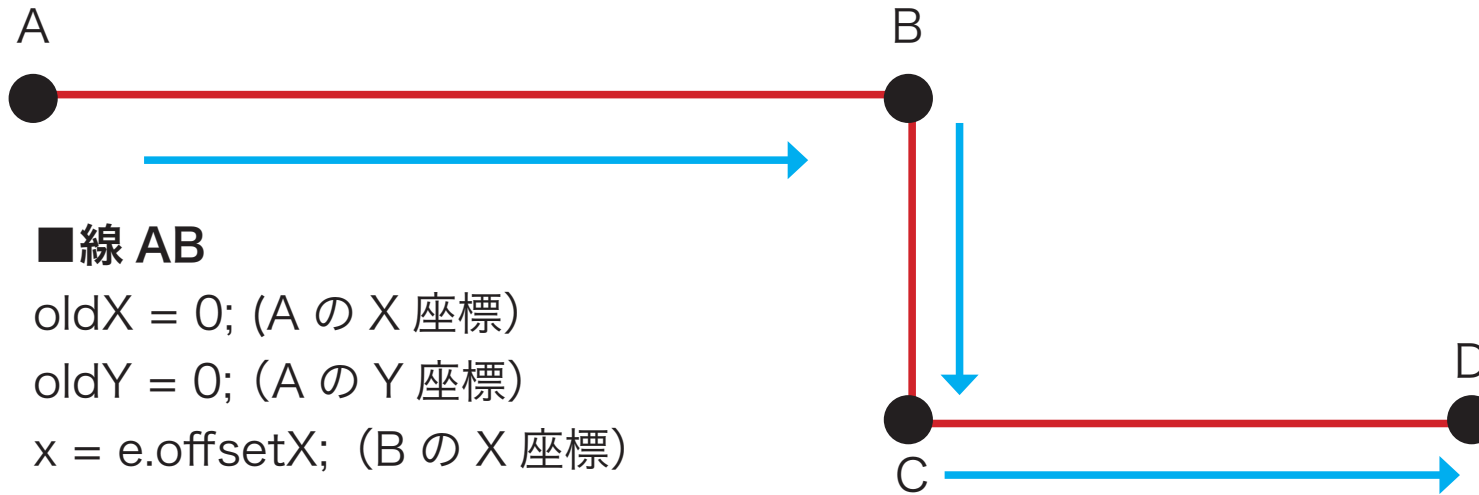


2. 「mouseup でマウス放したら
線を描かないようにする」

3. 「canvas 要素から mouse が外にでたら、
線を描かないようにする」

4. 線の色を変える

線を描画する動き



■線 AB

`oldX = 0;` (A の X 座標)

`oldY = 0;` (A の Y 座標)

`x = e.offsetX;` (B の X 座標)

`y = e.offsetY;` (B の Y 座標)

→ **x と y が、線 BC の出発点になる！**

`oldX = x;`

`oldY = y;`

線を描画する動き

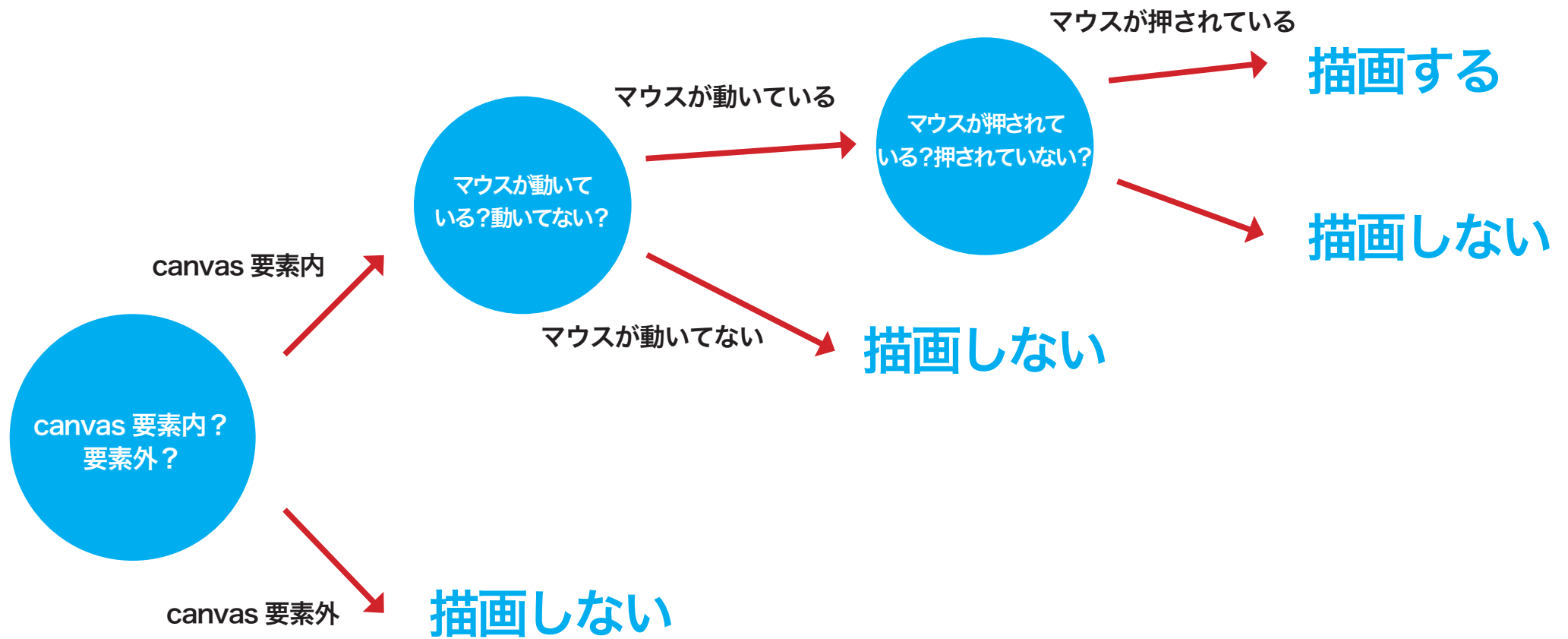
```
let x = e.offsetX;  
let y = e.offsetY;
```

線 AB の終了点を代入

```
ctx.beginPath();  
ctx.moveTo(oldX,oldY);  
ctx.lineTo(x,y);  
ctx.stroke();
```

```
oldX = x;  
oldY = y;
```

線 AB の終了点が
線 BC の開始点になる



■判定の仕方

- canvas 要素内か、canvas 要素外か? . . . マウスの座標位置で考える
- mouse が動いている? 動いていない? . . . 動いている→mousemove、動いていない→該当イベントなし
- マウスが押されてる? 押されてない? イベントで判定できない

状態のフラグ管理を覚えよう！

今回の演習でいうと、「マウスが押されている or 押されていない」を判定するための変数を作り、if 文を使って、押されている時にだけ描画が実行されるように JavaScript を記述してみます。

```
// マウスが押されている時は true (1)、押されていないときは false (0) と定義する

//false の時はマウスが押されていない (mouseup)
let flag = false;

//true の時はマウスが押されている (mousedown)
flag = true;

//if 文を使って判定
if(flag==true){
    // マウス描画するプログラムをここに記述
}
```

色（プロパティ）の変更

input 系タグの値を変更→ JavaScript から取得したい時は、change イベントが便利！
値が変更された段階で、input 系タグの value 値を取得出来ます。



■関連して覚えておくといつか使えるかもしれないイベント

- change
- blur
- focus

課題発表

次回までの課題内容

お絵かきアプリのさらなるカスタマイズ！！！（以下最低ラインとして制作）

0. 先ほどまでの内容は完全にクリアをする
1. 線の太さを変える機能をつける
2. 消しゴム機能をつけてみる。（方法は色々ある！）
3. 保存ボタンを作成 → ボタンを押すと LocalStorage に Canvas の内容が保存される
（可能なら表示 or 呼び出しボタンで、保存されているデータを Canvas 上に表示する）



※ toDataURL

<http://www.html5.jp/canvas/ref/HTMLCanvasElement/sample/toDataURL.html>

※ drawImage

<http://www.html5.jp/canvas/ref/method/drawImage.html>

余裕のある方はどんどん挑戦だ！



Canvas で色々なことに挑戦してみよう！

- ・ お絵描きアプリを超高機能にする
- ・ Canvas でシューティングゲームを作る
- ・ 花火を打ち上げてみる
- ・ 時計を作る



Developer Tools のブレイクポイント機能

Developer Tools のブレイクポイント機能を使ってデバッグ上手に！

指定した関数が動かない！？そんなときはありませんか？？

そんなときは、Developer Tools のブレイクポイントを使うと、かなり便利に効率的にデバッグを行うことができます。

今回のうちに、ブレイクポイント機能の使い方をマスターしましょう！

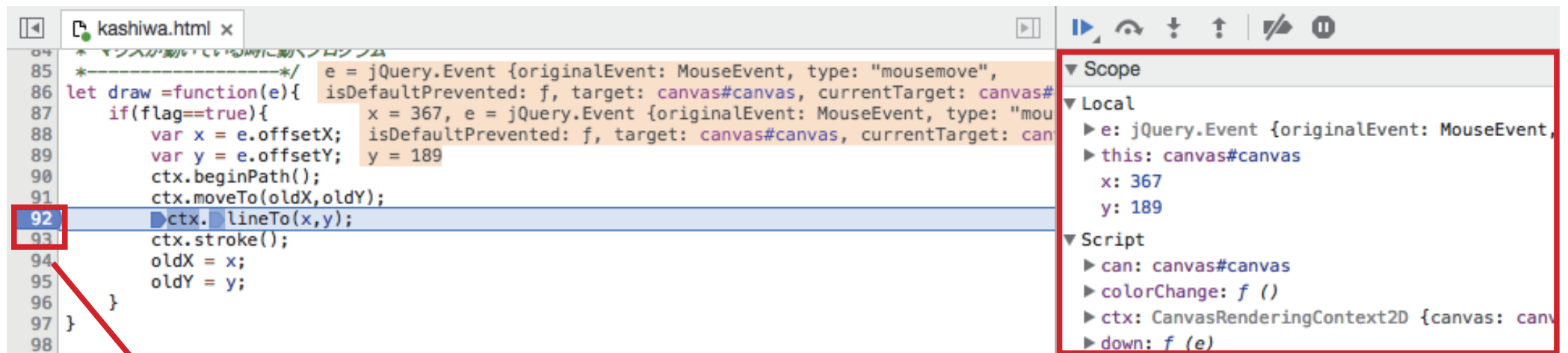
Developer Tools のブレイクポイント機能

Sources を選択



プログラムの動きを止めたい行にマーカーを打ちます。その行が読み込まれた時点で、一度動きを止めることができます。

Developer Tools のブレイクポイント機能



プログラムの動きを止めたい行にマーカーを打ちます。その行が読み込まれた時点で、一度動きを止めることができます。

動きを止めた時点で、各変数のスコープ範囲や、具体的な変数の中身などを確認することができます。
(scope 欄)

e ってなんだ？？

e は、イベントオブジェクトに相当します。

関数の function 直後の () の中で、イベントオブジェクトと呼ばれるものを取得することができます。

クリックイベントの場合であれば、**クリックイベントが起きた地点の各種情報（座標位置とか）を取得することができるようになります！**

（e は汎用的に使用されているだけで、必ず e を書かなければならないというわけではありません）

お手上げよ～という方へ。。。



前回の課題が出来なかった方！今回の内容意味不明という方！
チュータリングタイムで解決しましょう！



17 時までは 1 人で頑張ってみて、17 時からのチュータリングタイムを
活用！今日のうちに課題の最低限の内容をクリアしてしまいましょう！

逆襲の写経

```
/*-----
 * お絵かきのおまじない2行
 *-----*/
const can = $("#canvas")[0];
const ctx = can.getContext("2d");

/*-----
 * データを保存するための場所「変数」の作成
 *-----*/
let flag = false;
let oldX,oldY;

/*-----
 * お絵かきのお工具箱から好きなものを選んで選ぶ
 *-----*/
ctx.lineWidth = "10";
ctx.strokeStyle = "#990";
ctx.lineCap = "round";

/*-----
 * マウスが押されているときに動くプログラム
 *-----*/
let down = function(e){
    flag = true;
    oldX = e.offsetX;
    oldY = e.offsetY;
}

/*-----
 * マウスアップした時に動くプログラム
 *-----*/
let up_out = function(e){
    flag = false;
}
```

```
/*-----
 * マウスアップした時に動くプログラム
 *-----*/
let up_out = function(e){
    flag = false;
}

/*-----
 * マウスが動いている時に動くプログラム
 *-----*/
let draw =function(e){
    if(flag==true){
        let x = e.offsetX;
        let y = e.offsetY;
        ctx.beginPath();
        ctx.moveTo(oldX,oldY);
        ctx.lineTo(x,y);
        ctx.stroke();
        oldX = x;
        oldY = y;
    }
}

// 動かしてみよう！
$(can).on("mousedown",down);
$(can).on("mousemove",draw);
$(can).on("mouseup mouseout",up_out);
```



ありがとうございました