

Javascript 05



G's ACADEMY
FUKUOKA



本日の内容

講義 + 作業 : 2.5 h程度
演習 : 1.5 h程度

アジェンダ

■canvasの基礎

■図形の描画

- ・ 矩形
- ・ 三角形

■ペイントツール作成

- ・ ドラッグ&ドロップで線を描く
- ・ 離したら描かないようにする
- ・ その他

■課題発表→チュータリング(演習)タイム

授業のルール

- 授業中は常にエディタを起動！
- 隣の人と相談するときは周りの迷惑にならない大きさで.
- 周りで困ってそうな人がいたらおしえてあげましょう！
- まずは**打ち間違い**を疑おう！

{ } ' " ; など

- 書いたら保存しよう！

command + s

ctrl + s

canvasの基礎

canvasの概要

■ canvasとは

- ・ 2Dのグラフィックスを描画する機能を持つ！
- ・ 3Dは未サポート（three.jsなど外部ライブラリ使用で実現可能）

■ できること

- ・ <canvas>でグラフやゲームのグラフィックが表現できる！
- ・ 例：<http://www.html5-memo.com/canvas/site/>

canvasの概要

■canvasの記述

```
<canvas id="***" width="***px" height="***px">
```

「id」 「width」 「height」 を指定する.

■注意点

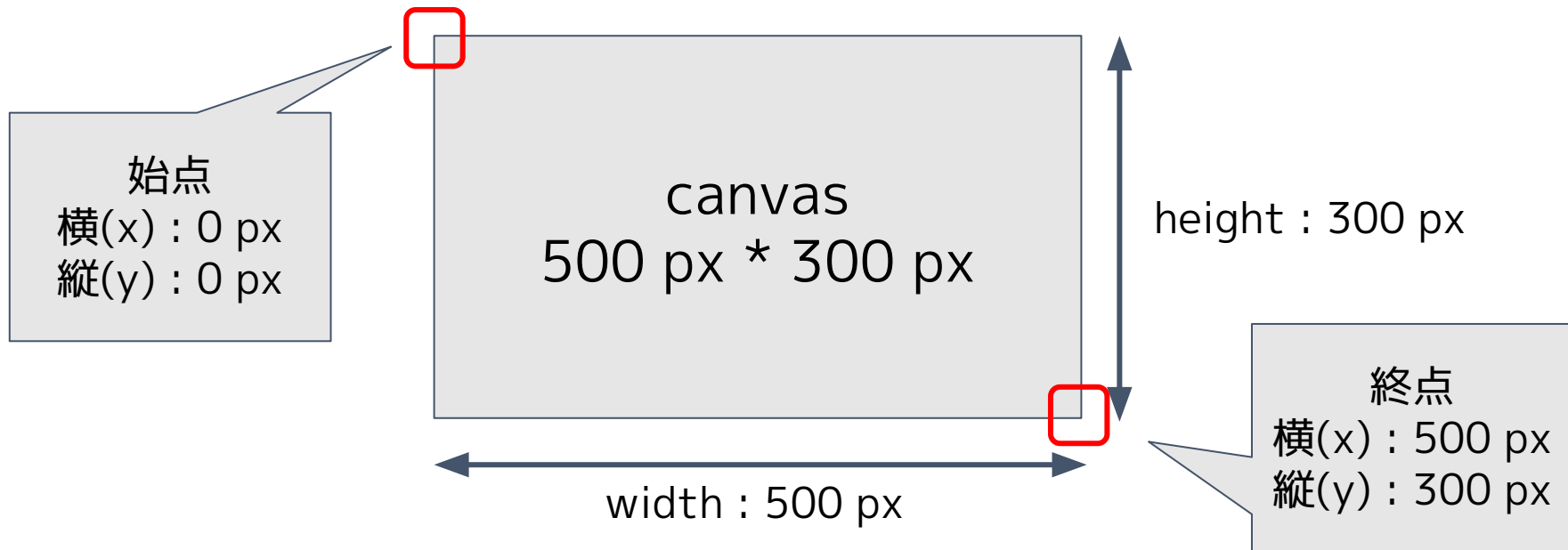
- ・ cssで指定(style=***)しても反映されない！
- ・ 必ず「width」 「height」 で指定！！

※ 「width」 属性とcssのwidthは異なる点に注意！

canvasの概要

■座標の指定

canvas内では座標で位置を指定します（単位はpx）



canvasの概要

■canvas実装の準備 (html)

```
<canvas id="***" width="***px" height="***px">
```

■javascript

```
var can = document.getElementById('drawarea');
```

```
var context = can.getContext('2D');    //現在は2Dのみ
```

上記「context」に対して描画の処理を記述する！

canvasの概要

■図形や線を描画する流れ

- ①図形の色や線の太さなどの条件を指定
- ②`context.beginPath();`
- ③図形や線を設定する処理
- ④実際に描画を行う処理

図形の描画

矩形の描画（線のみ）

canvas_rect.html

■図形や線を描画する流れ

```
var can = $('#canvas')[0];           //id指定
var context = can.getContext('2d');
context.beginPath();                 ///パスの開始
context.rect(0, 0, 60, 60);           //矩形を設定する処理
context.stroke();                     //線を描く処理
```

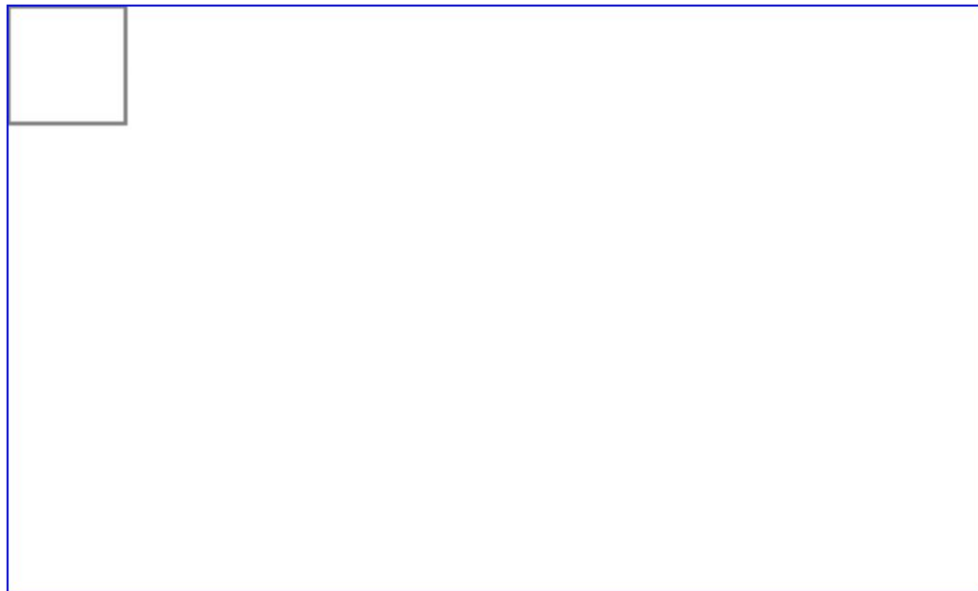
矩形の定義はrect(始点x, 始点y, 終点x, 終点y)

線を引く処理がstroke()

動作確認

canvas_rect.html

■ 矩形



■直線を組み合わせて三角形を描画する流れ

...

```
context.beginPath();
```

```
context.moveTo(20, 200); //始点を設定
```

```
context.lineTo(100, 20); //始点から頂点1までラインを設定
```

```
context.lineTo(200, 200); //頂点1から頂点2までラインを設定
```

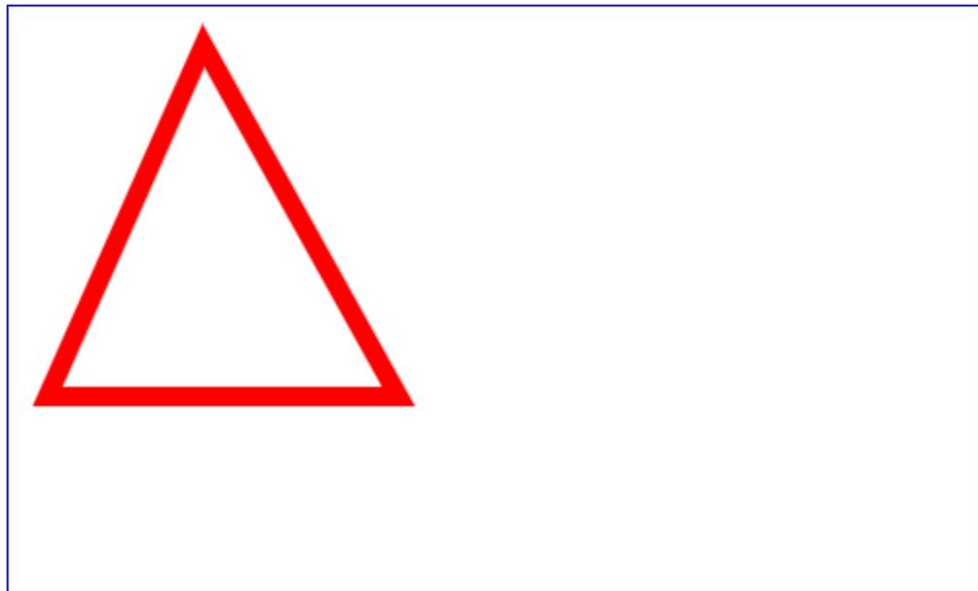
```
context.closePath(); //自動的に図形を閉じる
```

```
context.stroke(); //描画
```

動作確認

canvas_tri.html

■ 三角形



練習

■ 2つの例について、実際に記述してみよう！

- ・ 動作確認のように表示されることを確認しよう！
- ・ 数値を変えて、どのように動きが変化するか試してみよう！
→ `closePath();` をコメントアウトしてみよう.
- ・ 文字を表示するサンプル (`canvas_font.html`) も試そう！

ペイントツール作成

ブラウザでペイントツールを作成しよう！

canvas_paint.html

■イメージ

- ・ クリックして**ドラッグすると線が引ける**。
- ・ 線の太さや色を変更できる。
- ・ 鉛筆モードと消しゴムモードがある。
- ・ 描いた内容をクリアできる。

■仕様とイメージ

- ①canvas要素内で**mousedown**+**mousemove**中は線を描く.
 - 切り替え用の変数を用意し, 「**true**」 「**false**」 で管理
- ②マウス離したら線を描かないようにする.
 - 「**mouseup**」 で管理
- ③canvas要素からmouseが外に出たら, 線を描かないようにする.
- ④線の色を変える.
- ⑤線の太さを変える.
- ⑥消しゴムボタンクリックで消しゴムモード

①mousedown+mousemove中に線を引く

canvas_paint.html

■クリックしたときのイベント

```
$(can).on('mousedown', function (e) {  
    console.log(e);  
    oldX = e.offsetX; //x座標  
    oldY = e.offsetY - txy; //y座標  
    draw = true; //切替用スイッチ  
});
```

現在の座標を取得できる！

①mousedown+mousemove中に線を引く

canvas_paint.html

■ドラッグしているときのイベント

```
$(can).on('mousedown', function (e) {
```

```
    if (draw == true) {
```

```
        ...
```

```
        context.beginPath();
```

```
        context.moveTo(oldX, oldY);
```

```
        context.lineTo(e.offsetX, e.offsetY - txy); //線を結ぶ先を設定
```

```
        context.stroke();
```

```
        oldX = e.offsetX;
```

```
        oldY = e.offsetY - txy;
```

```
    }
```

```
}
```

今の座標と一つ前の座標がキー

//スタートする座標（一つ前）を設定

//線を引く

//一つ前の座標を更新する

②離したら線を描かなくする処理

canvas_paint.html

■離れたときのイベント

```
$(can).on('mouseup', function (e) {  
    draw = false;           //変数をfalseにして描画を中止  
});
```

■ 2つの処理を実装しよう！

- ・ クリックして動かしたときに線を引く.
- ・ ボタンを離したら線を引かないようにする.

③canvasから外れたら線を描かなくする処理

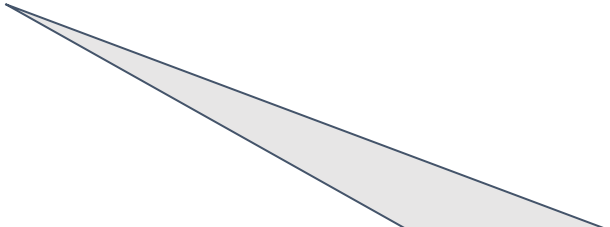
canvas_paint.html

■外れたときのイベント

```
$(can).on('mouse????', function (e) {
```

```
    draw = false;           //変数をfalseにして描画を中止
```

```
});
```



処理自体は離れたときと同じ！

④線の色を変える処理

canvas_paint.html

■色ボタンのイベント

```
<input id="color" type="color">
$('#color').on('?????', function (e) {
    console.log('色変更！');
    color = ????????;
});
```

イベント名と値のとり方を考えよう！
(色を入れる変数「color」は上で定義済)

⑤線の太さを変える処理

canvas_paint.html

■太さボタンのイベント

```
<input id="bold" type="range">
```

```
$('#bold').on('?????', function (e) {
```

```
    console.log('太さ変更！');
```

```
    bold_line = ????????;
```

```
});
```

色変更と同じパターン！！

⑥消しゴムの処理

canvas_paint.html

■消しゴム切り替え

```
$('#erase').on('click', function (e) {  
    console.log('消しゴム！');  
    color = '#fff';  
});
```

色を白くすればOK??

■残りの処理も調べながら実装しよう！

- ・ canvasから外れたら描かないようにする.
- ・ 色変更
- ・ 太さ変更
- ・ 消しゴム

※その他

- ・ 透明度の管理
- ・ 戻るボタン（高難度クエスト）

課題

【課題】ペイントツール作成

■最低限ここまで作ろう！！

- ①canvas要素内で**mousedown**+**mousemove**中は線を描く。
→切り替え用の変数を用意し、**「true」「false」**で管理
- ②マウス離したら線を描かないようにする。
→**「mouseup」**で管理
- ③canvas要素からmouseが外に出たら、線を描かないようにする。
- ④線の色を変える。
- ⑤線の太さを変える。
- ⑥消しゴムボタンクリックで消しゴムモード

【課題】 ペイントツール作成

■ プラスアルファの例

- ・ 戻るボタン
- ・ 画像アップロード
- ・ 画像編集&ダウンロード
- ・ 3D表示（three.jsというライブラリがあります）
- ・ インベダーゲーム

提出は次週木曜日「23:59:59」まで！！

今回でjavascriptラスト！
写経も卒業！

チュータリングタイム

17:00までは一人でもくもく
後半は近くのメンバーで教え合おう！